Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems

Cooperative Sensing by a Group of Unmanned Aerial Vehicles in Environments with Obstacles

Doctoral Thesis

Ing. Vít Krátký

Prague, January 2024

Ph.D. programme: Computer Science

Supervisor: RNDr. Petr Štěpán, Ph.D.
Supervisor-Specialist: doc. Ing. Martin Saska, Dr. rer. nat.

Acknowledgments

The whole period of my Ph.D. studies was a continuous learning process from good and bad examples of behavior, attitude to various aspects of life and research, and knowledge shared through personal interactions and academic resources. Hence, the list of people contributing to this thesis in a certain way is too long to detail them individually. However, a few people contributed significantly to improving my knowledge and skills and keeping me motivated throughout the work on this Ph.D. thesis, thus deserving special acknowledgment.

First, I would like to express my gratitude to my supervisor specialist, Martin Saska, for inspiring me to start Ph.D. studies, which I would never have thought about myself. Further, I would like to acknowledge both my supervisors, Petr Štěpán and Martin Saska, for introducing me to the world of academic research, providing valuable advice and feedback on my work, and for the opportunity to engage in international projects with a highly motivated and extremely knowledgeable team. I am grateful to all former and current members of the Multi-robot Systems Group at CTU in Prague for inspiring discussions, continuous development of the MRS UAV System that enables me to transfer my research into a real-world environment and meaningful applications, and for creating a motivational, friendly atmosphere allowing me to significantly extend my working hours beyond the standard while still enjoying the work.

Special thanks go to Tomáš Báča for always being there to answer questions that stumped everyone else, Matěj Petrlík for showing me that the level of responsibility for your work and the level of prioritizing helping others over your own work may not have a limit, Robert Pěnička for sharing his experience, providing invaluable advice, and also personalizing the hope that doing Ph.D. studies does not necessitate sacrificing a family life, and finally to Pavel Petráček for all the true collaboration on joined projects, seamless balancing of our workload, sharing the same enthusiasm for solving problems we have faced throughout our studies, and also for sharing the frustration during our process of learning that the others do not always share our working attitude and requirements on the quality of the work. My appreciation extends to other colleagues, including but not limited to Daniel Heřt, Jan Bednář, Jiří Horyna, Vojtěch Vonásek, Matouš Vrba, Petr Štibinger, Viktor Walter, and Vojtěch Spurný, whose impact on my research activities and results was also significant. Besides, I acknowledge our collaborators from the CTU-CRAS-NORLAB team at DARPA SubT, as well as the colleagues working with us in the DRONUMENT and AERIAL-CORE project.

Further, I extend my heartfelt thanks to prof. Pavel Dungl, Jan Masin, and my parents for the opportunity to live an active life, thus being able to alleviate the stress through physical activities. Finally, I would like to thank my girlfriend Martina for her support, and especially for asking annoying questions about finishing my studies often enough to force me to submit the thesis at the right time.

During my Ph.D. studies, my work had been supported by the government of the Czech Republic through a Ph.D. scholarship. Further financial support was provided by the Czech Technical University through grants SGS17/187/OHK3/3T/13, SGS20/174/OHK3/3T/13, and SGS23/177/OHK3/3T/13. The Ministry of Culture of the Czech Republic supported the work by the grant no. DG18P02OVV069. The Ministry of Education of the Czech Republic supported my research by OP VVV funded project no. CZ.02.1.01/0.0/0.0/16019/0000765. Further support to this work was provided by the Czech Science Foundation under research projects no. 23-06162M, 20-10280S, and 20-29531S. The Defense Advanced Research Project Agency supported the presented work through funding of our participation in DARPA SubT Challenge. Last but not least, this work was partially funded by the European Union under the project Robotics and advanced Industrial production (no. CZ.02.01.01/00/22_008/0004590), and Horizon 2020 research and innovation programme under grant agreement no. 871479.

Copyright

This thesis is a compilation of several articles published and submitted during my Ph.D. studies. The included publications are presented under the copyrights of IEEE, and Wiley Periodicals LLC for posting the works for internal institutional uses. The works are protected by the copyrights of respective publishers and cannot be further reprinted without the publishers' permission.

Abstract

This thesis concerns the autonomous navigation and coordination of multi-rotor Unmanned Aerial Vehicle (UAV) teams in cooperative sensing tasks. Multi-rotor UAVs have become a very popular tool in a variety of tasks requiring gathering sensory measurements from diverse locations. The range of possible applications has been further increased by progress in the development of localization and control algorithms that enable deployment of UAVs in environments with high obstacle density. Cooperative sensing offers advantages over single-robot systems by distributing sensory equipment and tasks to be solved among multiple robots, thus overcoming the limited payload of UAVs and enabling more effective data collection including obtaining simultaneous measurements from distant locations. This thesis focuses on cooperative sensing methods that necessitate tight cooperation between robots and cannot be performed by a single robot. These methods bring new challenges in coordination and cooperative motion planning of UAVs in constrained environments required to perform the missions effectively. The first part of the thesis details the design and development of a software stack for autonomous navigation of multi-rotor UAVs in indoor, GNSS-denied environments motivated mainly by search and rescue operations. The proposed UAV system was verified in a DARPA SubT challenge robotic competition, where the designed technology was among the top-performing UAV-based systems. In the second part of the thesis, the focus is on the development of a novel, worldwide unique technology for cooperative sensing, tailored for advanced documentation techniques. This innovative approach utilizes a team of tightly cooperating multi-rotor UAVs, equipped with a varying sensory and lighting tools, for realization of documentation techniques applied in the field of restoration and historical science. This technology, leveraging the developed autonomous indoor navigation system, was used to document eighteen valuable historical structures, such as churches, historical halls, and chapels, including monuments listed among UNESCO World Heritage sites. The last part of the thesis tackles the problem of time-optimal formation reshaping in three-dimensional environments with guarantees of inter-agent collision avoidance. The efficient solution to this problem provides the basis for expanding the previously proposed cooperative sensing methods to large-scale scenarios involving multiple cooperating formations of UAVs in a shared operational space. Besides the development and design of novel methodologies and frameworks to tackle diverse research problems, particular emphasis is put on the evaluation and experimental verification of all of the proposed algorithms and technologies towards their applicability onboard light-weight UAVs in real-world applications. To this end, all algorithms and technologies presented in this thesis have been verified in real-world scenarios and often also deployed in targeted applications, helping to perform required tasks with increased efficiency compared to conventional methods.

Keywords Unmanned Aerial Vehicles, Remote Sensing, Cooperative Motion Planning, Autonomous GNSS-denied Navigation, Formation Reshaping, Automated Inspection, Documentation of Historical Buildings

Abstrakt

Tato práce se zabývá autonomní navigací a koordinací týmu vícerotorových bezpilotních helikoptér (UAV) v úlohách kooperativního snímání. Vícerotorové UAV se staly velmi oblíbeným nástrojem v řadě úloh vyžadujících shromažďování senzorických měření ze zadaných míst. Rozsah možných aplikací byl dále rozšířen pokrokem ve vývoji lokalizačních a řídicích algoritmů, které umožňují nasazení UAV v prostředích s vysokou hustotou překážek. Oproti systémům s jedním robotem, kooperativní snímání umožňuje rozdělit senzorické vybavení a požadované úkoly mezi více agentů, čímž překonává nevýhodu omezeného užitečného zatížení UAV a umožňuje zvýšit efektivitu sběru dat včetně provádění simultánních měření ve vícero vzdálených místech. Tato práce se zaměřuje především na metody kooperativního snímání, které vyžadují úzkou spolupráci mezi jednotlivými roboty a z principu je není možné realizovat pomocí jediného robota. Tyto metody přináší nové výzvy v podobě koordinace robotů a přístupů ke kooperativnímu plánování pohybu v prostředí s překážkami, jejichž řešení je základem efektivní realizace požadovaných úloh. První část práce podrobně popisuje návrh a vývoj metod pro autonomní navigaci vícerotorových UAV ve vnitřních prostředích motivovaný především pátracími a záchrannými operacemi. Navržený systém byl ověřen v rámci robotické soutěže DARPA SubT Challenge, kde vyvinutá technologie patřila k nejlepším systémům na bázi UAV. Druhá část práce se zabývá návrhem celosvětově unikátní technologie pro realizaci pokročilých technik kooperativního snímání. Navržený inovativní přístup využívá týmu úzce spolupracujících vícerotorových UAV, vybavených různými zdroji osvětlení a senzory pro sběr dat, pro realizaci dokumentačních technik aplikovaných v oblasti restaurování a památkové ochrany. Tato technologie, využívající systém pro autonomní navigaci UAV v interiérech budov, byla použita k dokumentaci osmnácti hodnotných historických staveb, například kostelů, historických hal a kaplí, včetně památek zapsaných na seznamu světového dědictví UNESCO. Poslední část práce se zabývá problémem časově optimální změny tvaru formace v trojrozměrných prostředích s garancí vyhýbání se kolizím mezi jednotlivými roboty. Efektivní řešení tohoto problému poskytuje základ pro rozšíření navrhovaných kooperativních metod snímání na rozsáhlé scénáře zahrnující více spolupracujících formací UAV ve sdíleném operačním prostoru. Kromě vývoje a návrhu nových algoritmů a technologií práce klade zvláštní důraz na hodnocení a experimentální ověření všech navržených postupů a systémů, a jejich využitelnosti na palubě malých kompaktních UAV v rámci reálných aplikací. Všechny algoritmy a technologie prezentované v této práci byly ověřeny v reálných scénářích a často rovněž nasazeny v cílových aplikacích, kde pomohly plnit požadované úkoly s vyšší efektivitou než které je možné dosáhnout s konvenčními technologiemi.

Klíčová slova Bezpilotní letecké prostředky, dálkové snímání, kooperativní plánování pohybu, autonomní navigace v interiérech, změna tvaru formace, automatizovaná inspekce, dokumentace historických budov

Abbreviations

CAT-ORA Collision-Aware Time-Optimal formation Reshaping Algorithm

CPP Coverage Path Planning

DARPA Defense Advanced Research Projects Agency

FoV Field of View

GNSS Global Navigation Satellite System

LiDAR Light Detection And Ranging

LBAP Linear Bottleneck Assignment Problem

LSAP Linear Sum Assignment Problem

MPC Model Predictive Control

NeBula Network Belief-aware perceptual autonomy

RTI Reflectance Transformation Imaging

S&R Search & Rescue

SLAM Simultaneous Localization And Mapping

SubT Challenge DARPA SubTerranean Challenge

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

UWB Ultra Wide Band

Contents

Chapter 1

Introduction

The rapid advancements in artificial intelligence and computational robotics, as well as in sensory and computational technologies, have enabled robots to move beyond well-established automated production lines and research laboratories to the general outdoor and indoor environments closer to everyday life. The miniaturization of various sensors and computational devices have also facilitated the increasing popularity of multi-rotor Unmanned Aerial Vehicles (UAVs), which have recently found utilization in various fields ranging from monitoring and inspection tasks [4c], [23], [24], to aerial cinematography [25], [26], Search & Rescue (S&R) scenarios [14a], [27], [28], and protection of critical structures against aerial intruders [29], [30] (Figure 1.1). However, the transition from well-controlled environments, often tailor-made for robotic operations, to the general dynamically-evolving real world imposes special demands on the perception, localization, and navigation capabilities of mobile robots, as well as their reliability.

Just as humans form social relationships to build communities and enhance their collective abilities and competitive edge, mobile robots similarly benefit from being organized into teams. Establishing robot teams augments their overall capabilities, reliability, fault tolerance, and operational efficiency, far surpassing the limitations of single-robot operations (e.g., limited payload in manipulation tasks, limited operational time in data collection missions, and lack of redundancy in safety-critical operations). The advantages of multi-robot teams have allowed UAVs to extend to various applications in professional domains, such as distributed remote sensing [31], cooperative monitoring of environmental disaster areas [24], [32], precision agriculture [33], and inspection tasks [34], [35]. A unique family of tasks is composed by those that go inherently beyond the scope of a single robot due to temporal and spatial constraints and thus the use of a team of UAVs is essential (e.g., cooperative manipulation [36], [37] and cooperative tracking or sensing in dynamic environments [5c], [24], [32]). These tasks deserve special attention as they require much tighter cooperation that goes beyond a simple allocation of goals and sharing data among team members, which is a typical level of cooperation in current multi-robot systems. These complex tasks necessitate a high degree of cooperation with precise coordination and maintenance of low inter-robot distances, rendering manual control and teleoperation practically unfeasible. Therefore, while autonomy in single-robot systems is a beneficial feature, it becomes an essential component for successful operation in the context of closely cooperating multi-robot teams.

The goal of this thesis is to develop algorithms for the realization of cooperative sensing tasks by a team of UAVs in cluttered real-world environments with possible extension to multiple cooperating UAV-teams sharing operational space. Further in this work, we deliberately omit cooperative sensing tasks that necessitate simultaneous visits to distant locations (e.g., data collection missions), focusing instead on tasks that demand close collaboration between particular UAVs. These tasks include, e.g., cooperative documentation techniques where part of the robotic team provides lighting to actively influence the quality of captured data by other members of the team. The specifics of these applications lie particularly in the demand for the synergy between robots and in unavoidable dynamics of the environment formed by the presence of multiple UAVs in a shared operational space. The need to coordinate the actions of individual UAVs has given rise to the development of cooperative motion planning algorithms and formation control theory, which are comprehensively reviewed in [38]. However, while many algorithmic solutions to this problem have already been developed, most of them deal only with a general formation, lacking in-depth inter-robot cooperation and not assigning specific roles to individual agents [39]–[42]. This thesis addresses the full range of challenges associated with these scenarios, from ensuring the reliable navigation of a single UAV in complex environments [1c], [2c], [6c], [8a] to developing algorithms for the coordination of UAV teams during cooperative sensing tasks [3c]–[5c], [18a], and even taking initial steps toward the deployment of multiple UAV teams with interchangeable agents [7c]. The thesis is structured around seven core publications [1c]–[7c], which are divided into three thematic sections. These publications collectively pursue a unified objective: to facilitate a safe deployment of UAV teams in complex sensing tasks that necessitate close cooperation among the UAVs. The achieved results are further supported by additional authored publications [8a-22a].

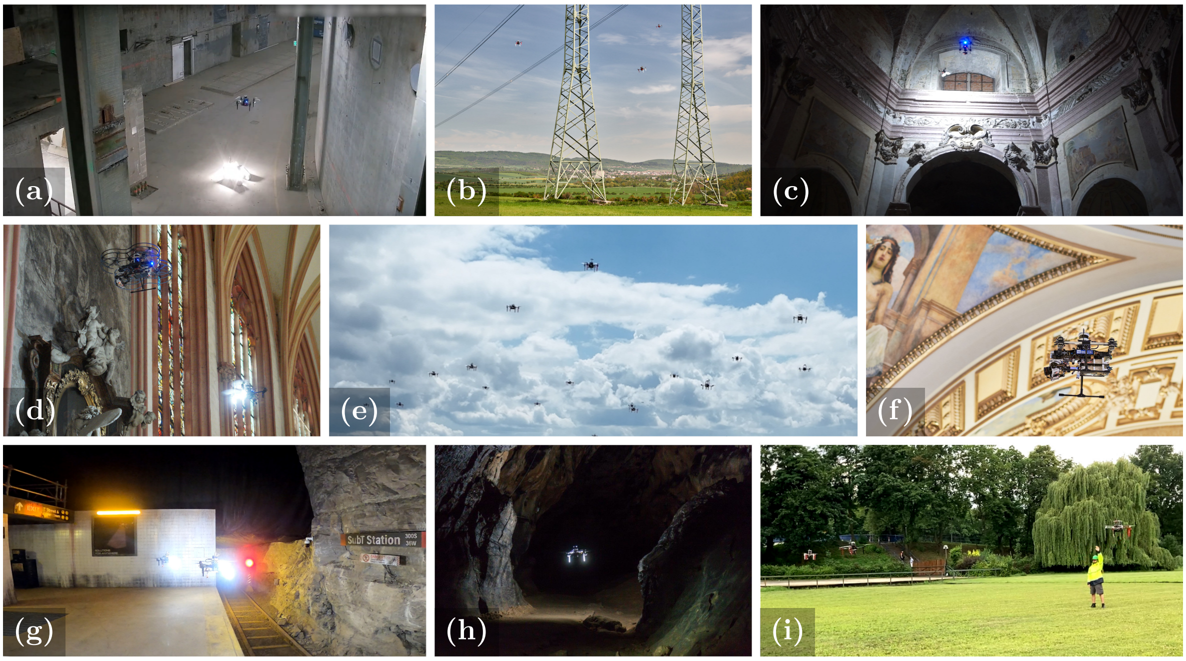


Figure 1.1: Example outcomes of authors' contributions presented in this thesis — (a) deployment of the UAV system for navigation in Global Navigation Satellite System (GNSS)-denied environments in an unfinished nuclear power plant [1c], (b) in cavern systems [2c], and (g) in multi-robot cooperative exploration scenario during DARPA SubT Challenge finals [6c], (c) realization of the Reflectance Transformation Imaging (RTI) documentation technique by a team of UAVs using the method proposed in [3c], (d), (f) deployment of the proposed framework [4c] for documentation of historical buildings in St. Maurice Church in Olomouc and the National Museum in Prague, (e) deployment of the proposed algorithm for collision-aware time-optimal formation reshaping [7c], (b), (i) deployment of cooperative motion planning and formation control algorithms [5c], [18a] in power-line infrastructure inspection and monitoring of a human worker.

The first set of core publications[1c], [2c], [6c] deals with single- and multi-robot autonomous navigation and exploration of GNSS-denied environments. This research was primarily inspired by the DARPA SubTerranean Challenge (SubT Challenge) whose goal was to detect and precisely localize objects of interest (e.g., survivors, their belongings, or a gas leak) in complex, previously unknown environments with human access forbidden. The challenge was held in three diverse environments simulating likely emergency scenarios where robots can assist rescue squads in initial reconnaissance of post-disaster sites, such as man-made tunnels, urban structures, and natural caves. During the competition, the deployed robots were exposed to adverse time- and space-varying real-world conditions that negatively affected both the traversability of the environment and perception of the robots (e.g., dust, humidity, mud, water, dynamic obstacles). The apriori unknown conditions and given limited deployment time within the competition forced participating teams to focus on the reliability of the deployed systems and their robustness with respect to all anticipated challenges.

While the SubT Challenge did not necessitate close cooperation among UAVs, the development of reliable autonomous navigation for a single UAV in indoor spaces lays a crucial foundation for all systems requiring autonomous operation in cluttered indoor environments, including realization of cooperative sensing tasks. In the relevant publications [1c], [2c], [6c], we introduced a novel UAV-based system designed for S&R scenarios featuring computationally efficient planning, exploration, and high-level mission planning algorithms that allowed for running the entire pipeline onboard lightweight UAV. The frequent deployment and testing of individual algorithms and the entire UAV-system in demanding and often completely unknown GNSS-denied environments helped to identify numerous challenges imposed by specific real-world conditions and unique characteristics of environments, such as high-humidity and dark cavern systems, abandoned buildings with transparent windows and extreme dustiness, and narrow man-made tunnels with repetitive visual and geometric features. Addressing these challenges in the development of the algorithms significantly increased the reliability of the system and the range of the environments and applications where the system can be effectively utilized.

The second collection of publications focuses on algorithms developed for advanced cooperative sensing tasks, particularly for the inspection and documentation of historical buildings. Documentation techniques applied in restoration and historical sciences often require capturing images with specific lighting angles to emphasize fine details, reveal three-dimensional structure, or enhance the overall quality of the captured materials. UAVs proved to be invaluable in such contexts, especially when dealing with large historical monuments characterized by areas that are challenging for humans to access, such as the most upper parts of altars, and fresco and murals located in cupolas of cathedrals and churches. The documentation tasks usually take place in environments that are either completely or partially known. However, the invaluable nature of these historical sites, with their priceless artifacts, requires a high level of operational safety. Therefore, we leveraged the expertise and algorithms developed for autonomous navigation in unknown indoor environments to design a UAV-based framework sufficiently reliable for use within historical structures. The specifics of the documentation techniques required to be performed in such environments imply the necessity of utilizing multiple tightly cooperating UAVs deploying algorithms that achieve safe and efficient operation of the UAVs in historical structures. In the relevant core publications [3c]–[5c], we present the entire framework for documenting historical buildings using UAVs along with particular methods for cooperative motion planning and coordination of UAV teams for realizing advanced documentation techniques, such as Reflectance Transformation Imaging (RTI) [43] and raking light. This framework has gained recognition, being incorporated into the methodology officially approved by the Ministry of Culture of the Czech Republic for the use of UAVs in documenting historical monuments.

The final set of publications presents a singular study [7c] on collision-aware formation reshaping, a crucial aspect of various practical applications involving large groups of mobile robots. Tasks, such as monitoring wildfires [44], cooperative object transport [36], drone light shows [45], [46], and maneuvering formations through cluttered environments [47], [48], all require adaptive changes of formation shape. Given the constrained operational duration of mobile robots, particularly multi-rotor UAVs, time efficiency is a critical concern when deploying robotic systems in real-world settings. Therefore, when the formation shape adaptation becomes part of a robotic mission, it is crucial to minimize the time required to complete this process. In the presented publication [7c], we introduce an algorithm designed to solve the problem of time-optimal formation reshaping, while ensuring collision avoidance between agents. We consider an efficient solution to this problem as one of foundational steps toward the efficient, long-term deployment of multiple collaborative robotic teams in shared operational spaces. Moreover, the problem of altering formations is not unique to the field of robotics. Thus, the significant results yielded by our algorithm have implications extending beyond the robotics community, impacting broader domains that require dynamic structural adjustments.

The remainder of this thesis is organized as follows. First, Chapter 2 summarizes the state of the art across relevant fields together with contributions of the thesis. Following this, Chapter 3 introduces works published in the domain of autonomous navigation within GNSS-denied environments, while Chapter 4 presents the core publications pertaining to cooperative sensing. Chapter 5 introduces the final core publication, focusing on the time-optimal formation reshaping. Next, Chapter 6 provides a detailed discussion, summary of the results achieved, and outlines potential future work. Finally, the thesis concludes with Chapter 7, which offers closing remarks.

Chapter 2

Contributions and Related Work

In this chapter, we highlight the contributions of this thesis, focusing primarily on the core publications [1c]–[7c]. These contributions are contextualized within the framework of related works in respective fields. The chapter is structured into three distinct sections, each aligning with the categorization of the core publications as outlined in the previous chapter.

2.1 Autonomous navigation and exploration of GNSS-denied environments

The tasks of autonomous navigation, exploration, and target search in GNSS-denied environments are often related to searching and inspecting post-disaster areas that are dangerous to enter due to potential structural instability, fire risks, or gas leaks. Automated vehicles equipped with various sensors are ideal for initial surveys of the areas of interest, being less vulnerable to potential risks imposed by the scenarios. Given the collected data, rescue personnel can assess the potential dangers or localize survivors to limit the time that human rescuers must spend at a potentially dangerous site. However, the technology currently used by joint emergency systems is mostly limited to teleoperated vehicles with a restricted operational range and stressful workload placed on human operators, further limiting the scalability of such an approach.

The problem of autonomous navigation of UAVs in GNSS-denied environments requires solutions to many challenges tackled in literature as self-standing research problems, starting with control algorithms [49], [50], through state estimation [51], localization and mapping [9a], path planning and trajectory generation [52]–[56], up to goal selection [57], communication [58], and high-level mission planning. Although literature offers high-performance solutions to these individual challenges [9a], [49]–[57], they often lack focus on real-world application performance metrics, such as reliability, robustness with respect to uncertainties in outputs of individual algorithms, resilience, and also the interconnections with other modules [59]. Therefore, the pure combination of the individual high-performance algorithms does not guarantee the highest performance in navigation and exploration tasks. Moreover, it often does not lead to a feasible solution due to computational demands and incompatibilities of individual modules.

The extreme complexity of the autonomous navigation task and requirements on the reliability in target scenarios allows for considering the design of autonomous systems for navigation and exploration of GNSS-denied environments as a distinct research topic, rather than an engineering system integration work. UAV-based systems capable of autonomous operation and exploration of unknown environments were presented in recent years, but often deployed and tested only in well-controlled environments with UAV-friendly conditions [60]–[66]. Significant advancements in this field are related to the teams that have participated in the SubT Challenge [67], [68] robotic competition organized by the Defense Advanced Research Projects Agency (DARPA). Due to the requirements on the payload, the extent of the environments, and narrow passages, the system's track of the SubT Challenge encourages teams to focus on the deployment of Unmanned Ground Vehicle (UGV)-based systems. However, the most successful teams in DARPA SubT Challenge competition took advantage of deploying diverse robots, including UAVs, to improve the overall mobility and agility of their robotic teams.

The winner of the challenge, team CERBERUS, designed a system based on the synergy of legged and aerial robots [69], [70]. The system deploys unified software stacks for tackling exploration path planning and local motion planning [71] and an onboard complementary multi-modal localization and mapping approach [72]. The unified software across the platforms allowed the team to narrow the focus of development and extensive testing, which were identified as crucial for achieving reliable operation in varying environments. The heterogeneity of environments was approached by deploying a team of robots with diverse mobility, payload and endurance, also by the second-best team in SubT Challenge, team CSIRO Data61 [73], that deployed tracked, wheeled, legged and aerial platforms utilizing a common sensor suite that consists of a spinning 3D Light Detection And Ranging (LiDAR), and RGBD cameras. All robots in the team processed the obtained data using CSIRO's Wildcat real-time multi-agent Simultaneous Localization And Mapping (SLAM)[1] and ran a frontier-based exploration approach utilizing direct point cloud visibility algorithm [74]. Team CSIRO Data61 also took advantage of using marsupial robotic systems to transfer the UAVs to areas that were suitable and advantageous for their deployment from the UGVs.

Team Marble presents their autonomous system deployed during preliminary circuits of the SubT Challenge in [75] followed by a description of the system designed for SubT Challenge finals together with achieved results in [76]. During the tunnel and urban circuit, the aerial part of the team relied on global map-based path planning with local reactive control based on direct high-resolution depth information to improve the resilience to dynamic environment, localization uncertainty, and imperfect trajectory tracking [75]. However, they decided to cease the development and deployment of aerial robots towards SubT Challenge finals. Team Explorer, on the other hand, employed both wheeled and legged UGVs along with collision-tolerant aerial platforms that could operate as selfstanding platforms, but could also be launched from wheeled UGVs [77], [78]. A detailed description of the UAV-based approach that focus on efficient connection of exploration using a range sensor with surface coverage by a cameras' Field of View (FoV) is provided in [79]. The authors combine behavior-tree for high-level decisions, RRT-connect-based [80] global planning maximizing the coverage of surfaces while exploring an environment, and fast local planning with adaptable speed based on distance to the closest obstacle. The proposed system demonstrated high resilience and reliability under adverse conditions, achieving the best performance in terms of the area explored in DARPA SubT Challenge Finals. Team CoSTAR deployed the developed Network Belief-aware perceptual autonomy (NeBula) framework, focusing on achieving resilience and intelligent decision-making through uncertainty-awareness [81]. The NeBula was deployed on legged, wheeled, and aerial platforms. The UAVs further utilize a multi-level rapid frontier exploration approach to effectively leverage the limited flight time of multi-rotor UAVs [82].

Our solution to UAV-based autonomous multi-robot exploration of unknown subterranean environments has been introduced in publications [1c], [2c], [6c]. The solution is based on the extensively tested multi-rotor UAV control and estimation system presented in [83] and

[1] https://www.csiro.au/en/research/technology-space/robotics/wildcat-slam

long-term multi-rotor platform development [10a], [15a]. The design of the proposed framework balances the flight time of the UAV platforms and the perception and computational capabilities key for reliable operation in complex environments. Considering these criteria, the designed system outperforms all UAV-based systems referred to in previous paragraphs, as its flight time is approximately double that of the maximum flight-time among other UAV-based systems participating in SubT Challenge, while having qualitatively comparable capabilities in performing required tasks in S&R scenarios. This result was achieved also through developing a computationally efficient software stack composed of tens of individual modules incorporated in such a way that restricts the computationally demanding data transfer and processing, which is essential for running such complex system onboard lightweight UAV. The list of developed individual modules comprises, but is not limited to, risk-aware, computationally efficient planning and exploration approaches avoiding demanding preprocessing of environment representation coupled with lightweight mapping and detection pipelines [1c], [6c], a graph-based multi-robot homing strategy for increasing the communication range of multi-robot systems [2c], a multi-robot cooperative exploration strategy utilizing SphereMap [84] (an effective representation of the environment for risk-aware planning), and sharing lightweight topological maps to facilitate efficient coordination of the robots [6c] under constraints on a bandwidth and range of communication.

Throughout development, the proposed system was deployed in the systems track of the Urban Circuit of SubT Challenge, where it explored the largest area among all aerial systems deployed from the starting gate. Further, the system was utilized both in the virtual track of the Cave Circuit and within the actual caves in Moravian Karst. Finally, the proposed system was deployed as part of the CTU-CRAS-NORLAB team in both the systems and virtual track of the SubT Challenge finals (see Figure 2.1) with minimal changes in software configuration for particular domains. As a key component of the virtual robotic team, the proposed UAV system achieved second place in the virtual competition, and was the second-best performing aerial system in the systems track of the DARPA SubT Challenge when considering the explored area and scored artifacts, and the best performing aerial system considering the operation time.

Although our solution has explored the narrowest areas among all deployed aerial platforms in the final round of SubT Challenge, the dimensions of the platform did not allow it to enter all areas of the environment safely. However, further reduction of dimensions would lead to unavoidable reduction in the payload, which would negatively affect the perception capabilities or endurance of the aerial platform. We have further addressed this problem through a cooperative exploration approach where a larger UAV with the full sensor stack navigates a smaller UAV with limited payload to constrained areas using relative localization based on detections in onboard 3D LiDAR data [19a].

2.2 Cooperative sensing by robotic teams

Thanks to their agility and the ability for reaching and hovering at difficult-to-access locations in constrained environments, multi-rotor UAVs have been commonly used as carriers of various sensory equipment since their introduction on the market. The UAV-based remote sensing applications, well-summarized in [85], [86], range from environmental surveys in outdoor open space [33], and the inspection of bridges [87], [88], power lines [34], and pressure vessels [89], to safety-critical documentation of historical buildings [90]. A common application in single-UAV remote sensing is the inspection and documentation of various structures,

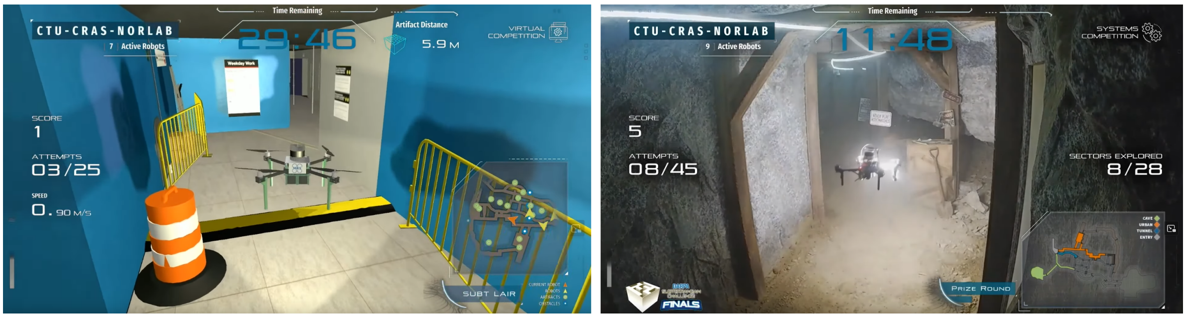


Figure 2.1: The UAV-based system for GNSS-denied navigation and exploration deployed as part of the CTU-CRAS-NORLAB team within both virtual and systems track of the DARPA SubT Challenge. The snapshots were taken from the official DARPA videos.

aiming to gather data for accurate 3D reconstruction or to capture detailed information in specific areas of interest. Much of the research in 3D scene reconstruction focuses on the Coverage Path Planning (CPP), where the objective is to generate paths efficiently connecting positions for scanning or capturing objects of interest to ensure adequate surface coverage for its later reconstruction, such as with photogrammetry methods [91]. Various approaches to solve the CPP have been applied for inspection of airplanes [92], radio and TV towers [93], power lines [94], castles [95] and bridges [96].

While a wide range of publications focus on the inspection of various structures [34], [87]–[90], [92]–[96], only a few address the entire problem of realizing inspection tasks in cluttered real-world environments. Some of these works present the entire process of realization of inspection relying on a GNSS. For instance, the authors of [23] propose an algorithm for large-scale aerial 3D scanning tested in various scenarios, including scanning of a barn. Similarly, in [97], a GNSS-based localization system is used to follow preplanned trajectories in inspection tasks performed by a fixed formation of UAVs. However, from a real-world point of view, the problem of an inspection task lies in flying relatively close to the inspected objects, often located in built-up areas where GNSS reliability for pose estimation is compromised. Therefore, several studies present automatic inspection systems that either rely entirely on non-GNSS localization systems or combine them with GNSS. An example solution relying solely on non-GNSS is presented in [35], which applies Ultra Wide Band (UWB) modules in uncooperative inspection of wind turbines by a team of UAVs. A fusion of multiple localization sources, including GNSS, is exploited in [87], where GNSS-based localization combined with pose estimation via the tracking of a prism mounted on the UAV by a ground total station is applied in the autonomous inspection of a bridge.

Considering indoor environments, UAV autonomy for inspection becomes rare, even though some applications strictly require such autonomy due to the characteristics of the environment not allowing for manual control of the UAV. In [98], the authors demonstrate a complete UAV system with vision-based localization applied to inspect a boiler inside a thermal power plant. In [99], an autonomous UAV is deployed to inspect cylindrical structures like penstocks and tunnels. Similar challenges are addressed in [100] and [101], where UAVs are used for inspecting empty gas vessels and chimneys, respectively. Beyond academic research, several commercial solutions have emerged for UAV-based realization of inspection tasks. These solutions, such as DJI[2] and Flyability Elios-3[3], typically offer outstanding assistance

[2] https://www.dji.com/cz/products/enterprise
[3] https://www.flyability.com/elios-3

features for manually controlled flights, but their autonomy remains limited. Systems tackling full autonomy within this application were recently presented by the Skydio[4] and DONECLE[5] company. The level of autonomy and reliability of these solutions is impressive, but inter-agent cooperation has yet to be introduced into these systems. Unfortunately, detailed information about the internal workings of these commercial products is often scarce, and their closed systems do not facilitate a fair comparison with other existing research and solutions in the field.

Although a single UAV can provide a large number of measurements during a single flight covering a vast area, the relatively low cost of UAVs enables to further speed up the process of data acquisition by employing a fleet of UAVs [35], [102]–[104]. This approach not only increases the speed of data gathering, but also enhances the fault tolerance of UAV-based remote sensing systems through redundancy. Furthermore, the distribution of sensing locations among particular UAVs allows to collect the sample measurements simultaneously at distinct locations. This capability is particularly beneficial for mapping dynamic events or localizing dynamic sources. Multi-UAV remote sensing has been applied in various fields, including the localization of radiation sources [31], monitoring of wildfires [24], and observation of oil spills [32]. However, in these applications, the level of cooperation is typically limited to simultaneous navigation to distant positions and more integrated forms of collaboration are not exploited. Such deployments emphasize efficiency and fault tolerance, but do not fully explore the potential of tight cooperative strategies among UAVs.

A specific application of remote sensing is cultural heritage documentation and preservation, where the use of UAVs is quite frequent. However, experts in the field of restoration and historical sciences primarily rely on commercial solutions, with their autonomy limited to following direct paths between predefined GPS positions [105], [106]. To the best of our knowledge, only a few works [90], [107], [108] address the UAV-based inspection of interiors of historical buildings. In [90], [107], the authors present an assistive system for the manual control of the UAV during inspection tasks, together with experimental deployment of the system in outdoor and indoor areas of historical buildings. The specialized platform for assisting in cultural heritage monitoring called *HeritageBot* is introduced in [108], but no evidence on the deployment of this platform in historical monuments is presented. Our work on documentation of historical buildings [3c], [4c], [8a], [11a], [16a] goes beyond these works in multiple aspects. First, we propose an autonomous system for the documentation of historical buildings that does not put additional requirements on the environment in which it is deployed [4c], [11a], [8a]. Further, in our recent publications [3c], [5c], [16a], we pushed cooperative sensing in indoor environments beyond the state of the art by presenting algorithms for performing documentation techniques that require the UAVs to fly in compact formations with low mutual distances. In these scenarios, the cooperation lies in an active influence of the perceived scene by part of a team in order to increase the quality of gathered data by the leading UAV (see Figure 2.2). To the best of our knowledge, such tight cooperation of UAVs for a single goal has been applied only in tasks of cooperative manipulation and transportation [36], [37] before.

[4] https://www.skydio.com/

[5] https://www.donecle.com/

Figure 2.2: Methods of cooperative sensing, including tight cooperation between individual agents, are applicable both in image capturing with distributed lighting (a) and simultaneous capturing of a dynamic scene from distinct directions (b).

2.3 Formation shape adaptation

The formation shape adaptation problem can be viewed as a particular case of cooperative motion planning, where the set of unassigned goals to be visited is provided to the group of robots instead of specific goals for each particular robot. Such a task essentially implies a necessity to combine the solution of the robot-to-goal assignment with the generation of collision-free trajectories for multiple robots. The formation reshaping task further shows two specifics with respect to the general assignment problem. Firstly, each robot must be matched with a single goal. This confines the problem to what is known as single-robot tasks, single-task robots, and instantaneous assignment (ST-SR-IA) task assignment problems [109]. Secondly, the existence of robots as physical entities in a shared environment requires considering mutual collision avoidance during the assignment process. This consideration implies that, in general, the individual cost of assigning two pairs of nodes in matching is not independent of the rest of the assigned pairs. This prevents direct use of general task assignment algorithms [110] for the solution of the formation reshaping problem.

The approaches applicable for solving the formation reshaping problem in robotics vary across several factors: the level of decentralization, environmental complexity and dimensions, optimization criteria, and the methodologies used. Centralized approaches [111]–[116] prevalent in the current literature mostly leverage the complete information to generate optimal solutions, but often operate under the assumption of a collision-free environment. In [111], the authors introduce a centralized method based on the concurrent solution of assignment of goals and planning of trajectories (CAPT algorithm), which is further extended to a decentralized approach. The CAPT combines the solution of the Linear Sum Assignment Problem (LSAP) [117], minimizing the sum of squared traveled distances with the implementation of constant-velocity and minimum-snap trajectories, which was proved to yield collision-free trajectories under the assumption on a minimum initial distance between agents. The problem is extended to a variable goal formation (variable scale and translation) in [112], where the authors show that the problem of goal assignment with variable goal formation can be transformed to LSAP via cost substitution. However, the approach is limited to 2D, and the mutual collisions are prevented by adapting the scale of the final formation.

Unlike the centralized solutions, the distributed approaches often suffer from incomplete information, leading to suboptimal solutions. For instance, a method involving local task swapping, as presented in [118], may not achieve optimal solutions due to its reliance on local information. Similarly, learning-based approaches [119] may offer limited guarantees on solution quality due to the inherent uncertainties in learning processes. Several studies have addressed the robot-to-goal assignment problem in the context of distributed control of multi-rotor UAV formations, both in obstacle-free areas [120] and complex environments [47], [48]. These works apply distributed task assignment algorithms [121], [122] to assign the robots to local goals during the alignment to the target formation. Although [47], [48], [120] are proposed primarily for multi-rotor UAVs, they utilize the sum of squared distances as the minimization criterion for the assignment problem. Such choice provides certain guarantees on the mutual distance of trajectories if solved optimally [111]. However, it does not reflect the requirements on the efficient operation of UAVs since minimizing squared traveled distances for in-flight multi-rotor UAVs is neither optimal from the point of view of flight duration nor energy consumption.

The time criterion, crucial for operational efficiency, is considered only in a few works dealing with formation reshaping. In [116], the authors prioritize minimizing the total time in motion and base the solution of an assignment problem on durations of time-optimal trajectories. The algorithm relies on collision resolution via a combination of time delays and altitude adaptation, which limits its application to 3D environments. The algorithm presented in [114] directly approaches the minimization of the makespan by defining the problem as a Linear Bottleneck Assignment Problem (LBAP). The proposed solution considers a variable goal formation, but is limited to 2D and does not account for inter-agent collisions. Another study addressing the minimization of the makespan is found in [115], which also formulates the assignment as LBAP, but considers constant-velocity trajectories only. The authors provide proof of collision avoidance guarantees; however, these are only valid under limitations to 2D with initial and final configurations constrained to the grid. These examples highlight a gap in the current research on formation shape adaptation, particularly in terms of efficiently combining time optimization and collision avoidance in an algorithm suitable for realistic environments and real-time onboard computation.

In the core publication [7c], we have introduced an optimal algorithm for solution of the time-optimal formation reshaping problem in a 3D environment, while considering inter-agent collision avoidance. The computational complexity of the algorithm allows its deployment on onboard autonomous vehicles in real-world missions, while significantly decreasing the makespan of the formation reshaping process. This result holds significant implications for enhancing existing methodologies and inspiring future research in the area of formation reshaping, particularly in the context of autonomous multi-robot systems in diverse real-world settings and limited operational time. The obvious target application for this algorithm is a spatial reorganization of a group of robots in various tasks, such as area monitoring [44], drone light shows [45], [46], precision agriculture [123], or active drug delivery systems [124], [125]. However, it also finds relevance in other fields where achieving specific spatial configurations or reorganizations is crucial for task performance, such as collision-free motion planning of discrete chemical droplets in analytical chemistry and biomedical diagnostics [126]–[128]. The broad applicability of this algorithm underscores its potential to impact various areas of robotics and automation, offering a versatile tool for complex spatial organization challenges.

Chapter 3

Autonomous Navigation in GNSS-denied Environments

In the first core publication [1c], we presented a complete framework for autonomous navigation of UAVs in GNSS-denied environments, that creates a basis for moving the UAVs from open, outdoor spaces into complex, cluttered environments. As an integral part of this framework, we developed a grid-based path planning approach and a frontier-based multi-robot exploration strategy specifically designed for scenarios with limited or completely unavailable communication. The proposed path-planning approach directly utilizes the output from the mapping module, eliminating the need for any time-consuming pre-processing of the map of the environment. This enables frequent global planning using the most recent representation of the environment, and thus timely reaction on an occurrence of dynamic obstacles. The iterative path post-processing increases the UAV-obstacles margin of the initial path to enhance the robustness of the system to imprecision of mapping, localization, trajectory tracking, and position control, all of which are inevitable in real-world environments.

The designed frontier-based exploration approach addresses the challenge of a time constrained multi-robot exploration in 3D environments, which can be considered as unbounded relative to the expected flight time of UAVs. The frontiers are selected based on assigned priorities influenced by a deviation from a given preferred absolute or relative direction of exploration, which varies across the robots. Such an approach enables achieving a cooperative exploration behavior in scenarios where communication is completely forbidden or impossible [2c]. The proposed framework integrating the aforementioned path-planning and exploration strategies, along with all remaining modules from low-level control and state estimation to high level planning, was deployed during the Urban Circuit of SubT Challenge where it achieved one of the best performances among UAV-based systems. As part of the CTU-CRAS-NORLAB team, the framework contributed to the team's success, securing first place among self-funded teams and third place overall in the Urban Circuit.

The contribution of the author of this thesis on the manuscript is 55%, which includes writing the manuscript, designing the computationally efficient planning and exploration approaches and high-level mission control, system integration, and experimental evaluation.

[1c] **V. Krátký**, P. Petráček, T. Báča, and M. Saska, "An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments," *Journal of Field Robotics*, vol. 38, no. 8, pp. 1036–1058, 2021

The algorithms and methods initially proposed and integrated for autonomous UAV operations were further refined with a specific emphasis on the cooperative multi-robot exploration of cave environments. This enhanced system was deployed in the Moravian Karst, a significant and complex cave system, and also participated in the Cave Circuit of the virtual track of SubT Challenge, which was held in a simulated environment inspired by natural cave conditions. In our subsequent core publication [2c], we detailed the advancements in our algorithms that improved inter-robot cooperation through introducing multi-robot homing strategy that significantly increases the operation range in the case that homing of all robots to an initial location is not required. We also described the results achieved during the Cave Circuit of SubT Challenge, along with a series of experiments conducted in real-world caves. The research showcased in this publication not only highlights the capabilities of UAVs in subterranean exploration, but also underscores the importance of cooperative multi-robot systems in effectively navigating and mapping complex, unstructured environments.

The author of this thesis contributed 20% to the manuscript [2c], including the implementation of planning and trajectory generation approach, system integration, writing path planning section of the manuscript, and participation in extensive experimental evaluation.

[2c] P. Petráček, **V. Krátký**, M. Petrlík, T. Báča, R. Kratochvíl, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021

The last core publication related to the topic of autonomous navigation in GNSS-denied environments summarizes the system developed over two years in preparation for the final round of the SubT Challenge [6c]. The manuscript contains both the description of applied methodology and the presentation of results achieved during this final stage of the SubT Challenge as part of CTU-CRAS-NORLAB team. In addition to other contributions, this manuscript introduces a multi-robot frontier-based exploration strategy utilizing lightweight topological volumetric maps that facilitate an efficient coordination of multiple robots in vast and complex environments, even under restricted communication. Furthermore, it details a complete pipeline for the detection, localization, and reporting of the found objects of interest. This includes a method for aggregating detection hypotheses, both within a single robot and across the entire robotic team. As an essential part of the fully autonomous multi-robot system deployed in the virtual track of the competition, the manuscript also describes an approach for an autonomous arbiter responsible for selecting the most confident hypotheses for reporting. This process can help to further filter data for human operators, thereby reducing their workload in real-world scenarios. While many components of the system were introduced in earlier publications, the system as presented in [6c] represents a significant advancement over its predecessors. It demonstrates superior functionality, adaptability to a variety of challenging environments, and overall performance in target search scenarios. Among teams composed from prestigious universities, research institutes, and companies specializing in S&R robotics, team CTU-CRAS-NORLAB achieved sixth place in the systems track and second place in the virtual track of the competition.

The contribution of the author of this thesis on this work is 15%, consisting of significant contributions in the system design, including design and implementation of high-level mission control, and autonomous arbiter for artifact reporting. Further, the author's main contributions reside in transferring the system to the DARPA virtual environment, system integration, simulation and real-world testing, and participation in writing the manuscript.

[6c] M. Petrlík, P. Petráček, **V. Krátký**, T. Musil, Y. Stasinchuk, M. Vrba, T. Báča, D. Heřt, M. Pecka, T. Svoboda, and M. Saska, "UAVs beneath the surface: Cooperative autonomy for subterranean search and rescue in DARPA SubT," *Field Robotics*, vol. 3, no. 1, pp. 1–68, 2023

An Autonomous Unmanned Aerial Vehicle System for Fast Exploration of Large Complex Indoor Environments

Vít Krátký[*]
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
vit.kratky@fel.cvut.cz

Pavel Petráček
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
pavel.petracek@fel.cvut.cz

Tomáš Báča
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
tomas.baca@fel.cvut.cz

Martin Saska
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
martin.saska@fel.cvut.cz

Abstract

This paper introduces an autonomous system employing multi-rotor unmanned aerial vehicles for fast 3D exploration and inspection of vast, unknown, dynamic, and complex environments containing large open spaces as well as narrow passages. The system exploits the advantage of small-size aerial vehicles capable of carrying all necessary sensors and computational power while providing full autonomy and mobility in constrained unknown environments. Particular emphasis is put on the robustness of the algorithms with respect to challenging real-world conditions and the real-time performance of all algorithms that enable fast reactions to changes in environment and thus also provide effective use of limited flight time. The system presented here was employed as a part of a heterogeneous ground and aerial system in the modelled Search & Rescue scenario in an unfinished nuclear power plant during the Urban Circuit of the Subterranean Challenge (SubT Challenge) organized by the Defense Advanced Research Projects Agency (DARPA). The main goal of this simulated disastrous scenario is to autonomously explore and precisely localize specified objects in a completely unknown environment and to report their position before the end of the mission. The proposed system was part of the multi-robot team that finished in third place overall and in first place among the self-funded teams. The proposed Unmanned Aerial Vehicle (UAV) system outperformed all aerial systems participating in the SubT Challenge with respect to versatility, and it was also the self-deployable autonomous aerial system that explored the largest part of the environment.

1 INTRODUCTION

Robotic systems have been employed for exploration, inspection and Search & Rescue (S&R) scenarios in unknown environments for decades. From the very beginning of their deployment in these tasks, teleoperated robots were intended to substitute a human rescue squad in hazardous, life-threatening tasks such as inspecting post-earthquake sites, places with a risk of explosion, underwater tunnels and the exteriors of high-rise buildings. The diversity of possible environments

[*] http://mrs.felk.cvut.cz

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021 2

of exploration, inspection and S&R scenarios, leads to the use of various robotic platforms. Mostly ground vehicles, aerial vehicles and underwater vehicles, where each of these types has mostly clear pros and cons.

Although fully teleoperated robots are very useful in these scenarios, they suffer from their dependency on operator commands and on the communication interface. This limits the number of deployed robots as well as their operational range. The ongoing development of algorithms and sensors has enabled the application of robots with a higher level of autonomy and has overcome some of the problems of teleoperation, has sped up the start of the systems and has made the systems more resistant to human misguidance. However, it has significantly increased the requirements on robustness of the system, on its reliability and on autonomous decision making, which are all crucial in the scenarios mentioned here.

In this paper, we present an autonomous aerial system designed for fast exploration and inspection of vast unknown dynamic environments, motivated by the DARPA Subterranean Challenge Urban Circuit (SubT Challenge)[1]. The goal of this challenge is to explore a previously unseen environment, to find specified objects and survivors, together referred as to *artifacts*, and to report their position in the map within a precision of 5 meters. During each run of the competition, 20 objects of five different types were located in the unfinished nuclear power plant. This challenging task is restricted by a one-hour time limit. One human member of the team is allowed to communicate with the robots on the course and remotely teleoperate them, but all physical manipulation with the robots on the course is forbidden. Competitors were even not allowed to enter the competition course before the end of the entire Urban Circuit challenge that truthfully reflects the conditions in real S&R missions.

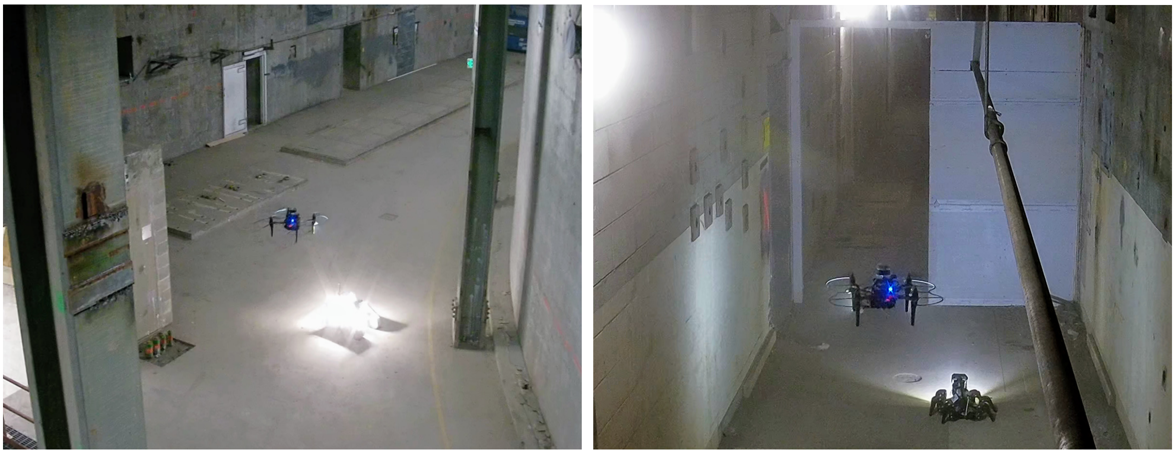
These rules simulate real-world conditions of S&R scenarios, and encourage the participating teams to develop and deploy highly reliable, robust, and autonomous systems. Various types of tracked, wheeled or legged ground vehicles are more favorable for long missions than Unmanned Aerial Vehicles (UAVs), since they can operate for several hours without battery replacement and are able to carry a heavy payload. However, UAVs have an indisputable advantage in overcoming obstacles, due to their remarkable agility and the speed with which they can move within a complex environment. Thanks to these features, UAVs have become an almost irreplaceable part of heterogeneous systems operating in complex environments. They can perform fast initial mapping, they can serve as quickly deployable communication translators, and they can explore parts of the environment that are completely unreachable by ground robots.

The proposed UAV system design aims to achieve maximum suitability for deployment in three-dimensional real-world unknown environments where no map exists and even the characteristics of the environment were unknown prior to deployment of the system. It therefore respects the requirements on resistance to challenging conditions, such as extreme dustiness and uncertain illumination, which are common for the intended scenarios. Particular emphasis is also put on minimizing the preparation time prior to the mission, since a fast and reliable start is for obvious reasons highly desirable for robotic systems applied in real-world challenges of S&R scenarios.

As part of the system, the novel approach to a grid-based path planning with path post-processing and a frontier-based exploration approach is presented. The proposed path planning approach with iterative post-processing directly uses the output of a mapping module and does not require any time-consuming pre-processing. This enables frequent updates of the map for global planning. The aim of the iterative post-processing is to increase the robustness of the system with respect to imprecision of localization, mapping, trajectory tracking, and position control, all of which are inevitable in the real-world. The designed frontier-based exploration approach aims to solve the problem of a time-constrained exploration task within apriori unknown 3D environment, which can be considered as unbounded with respect to the expected flight time of UAVs. The exploration approach is designed to be deployed within a multi-robot exploration task with limited or completely unavailable communication.

To evaluate the capabilities and properties of the system, an analysis of the real-time performance of particular subsystems is presented. The analysis includes a detailed inquiry into the contributions of the proposed approaches to verify and validate its influence on the robustness of a real-world system. The performance of the entire system is evaluated in numerous simulations in general 3D environments, including the virtual urban, and cave environments of the DARPA Subterranean Challenge. Moreover, the system was deployed within the real-world urban circuit track of the same competition, where it was a part of a multi-robot team deployed for a challenging scenario held in the unfinished Satsop Nuclear Power Plant located in Washington state, USA. In one of the real-world scenarios, the proposed system consisting of two UAVs explored the surroundings of the reactor hall with a total area of $1351\,\mathrm{m}^2$ in a total time of $453\,\mathrm{s}$. In the second scenario, we successfully tested the capability of our system to fly through narrow passages that are only $34\,\%$ wider than the footprint of the aerial platform. To show the applicability and to further showcase the seamless transition of the proposed methodology between various 3D environments, the system was also deployed during an exploration task in the Byci skala cave system of the Moravian Karst in the Czech Republic.

[1] https://subtchallenge.com/



(a) Alpha course

(b) Beta course

Figure 1: The proposed system operating in a simulated disastrous scenario during the Urban Circuit of the Subterranean Challenge. The figure shows the UAV platform as a part of a multi-robot team deployed within open spaces (a) and confined corridors (b) of the unfinished nuclear power plant in the Satsop Business Park, USA, WA.

1.1 Contributions

The absence of robustness, large-scale deployability, and utilization in demanding real-world environments, were identified by the DARPA as the main bottlenecks of current robotic technology developed for S&R scenarios. The DARPA organisers team, together with representatives of S&R-technology end-users, therefore designed three DARPA SubT Challenges with gradually escalating demands on robotic systems that push the current robotics state of the art towards the requirements for real-world deployment. The aim is to design a methodology that can be used without the need to tune the system parameters for particular experimental trials and without tuning the environment itself.

The participants in the SubT Challenge are not allowed to enter or even to see the environment of the competition trials before the robots are deployed. The participants know only the general characteristics of the workspace, similarly as in the case of real S&R missions. The real conditions of S&R scenarios are further emulated by the rules of the competition, which state that the one-hour mission starts according to a fixed schedule and teams can use only a very limited time for preparation before their trials. In the first SubT Challenge (tunnel systems of underground mines), the participating teams presented solutions going far beyond the current state of the art in terms of robustness and deployability in demanding environments, mainly for UAVs, since the real underground mine tunnels were dusty and full of unexpected obstacles [Miller et al., 2020, Ebadi et al., 2020, Rouček et al., 2019, Huang et al., 2019, Dang et al., 2019]. Although the tunnel circuit of the SubT Challenge was a very challenging task that pushed frontiers of field and S&R robotics, the environment was expected to be flat and constrained by tunnels that reduced navigation and motion planning problems into 2D space and often almost 1D space, as the obviously best strategy for Unmanned Aerial Systems (UAS) was to follow the longitudinal axis of a tunnel keeping maximum distance from its walls. For the second circuit of the SubT Challenge (indoor Urban Circuit), the aerial systems had to be completely redesigned in terms of both, HW and SW to allow full 3D motion in the completely unknown environment of a nuclear power plant.

Solving this new robotic problem of S&R coming from a plain and constrained environment into full 3D, where UAVs are more suited for use, together with the enforced large-scale deployability, is the main contribution of the SubT Challenge Urban Circuit. These are also the main contributions of this paper that presents a novel UAV system that achieved the best performance among all self-deployable aerial robots in the competition. Since the rules of the competition allow all types of inter-robot cooperation, several of the aerial platforms of other competitors were designed to be deployed from ground robots at some advantageous place on the course. In this paper, the term self-deployable is used to mark robots which do not need any assistance from other robots to enter the course, and so they can be deployed independently from any other system. To the best of our knowledge, the UAVs driven by this system travelled the longest distance in comparison with the self-deployable UAVs of other competitors, explored the largest area in the shortest time, and were able to traverse the narrowest passages relative to the size of the vehicles that were employed. In addition, a novel autonomous exploration and planning approach was designed for deployment in a demanding underground environment was designed to explore the largest possible amount of space in the shortest time, while simultaneously being able to fly

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021 4

through narrow passages. This is also a common requirement in most S&R missions, where no prior information on the positions of human survivors is available.

The main contributions of this paper are as follows:

- A complete fully-autonomous system for fast exploration of general apriori unknown 3D environments is presented. The performance of the system is objectively examined with respect to various experimental deployments in virtual, as well as real-world, S&R conditions in challenging urban and cave environments.
- A novel approach combining original grid-based path planning and path post-processing approaches is presented, increasing the robustness of the system with respect to imprecision of localization, mapping, trajectory tracking, and position control, all of which are inevitable in the real-world.
- To tackle the challenge of exploration of general 3D environments with limited time, a frontier-based strategy designed for use in multi-robot scenarios with limited or completely absent communication is presented.
- Extensive experimental analyses of aerial system in varying environments is presented, validating our approach during single and multi-UAV deployments, as well as in small and large-scale experiments in simulated and real-world urban and cave environments.

2 RELATED WORK

2.1 Localization and Mapping

In S&R scenarios, the disaster environments are mostly unknown (although floor plans of the object may be available to serve as a semantic reference for a mission operator). The mapping therefore has to be solved simultaneously with the localization of a mobile robot. The simultaneous localization and mapping (SLAM) is one of the most extensively studied and also one of the most complex problems in the field of robotics due to integration of two mutually reliant systems, both integrating various errors of sensory discrepancies and algorithmic approximations. The problem is further complicated for mobile robots, especially for dynamically unstable aerial vehicles, as these robots require a perpetual and real-time estimate of their state (the pose and its derivatives) in order to stabilize and control their motion. The SLAM algorithms have to run onboard these robots, using only onboard sensors and providing their outputs in real-time. In this section, we will, for obvious reasons of a S&R task, focus on related work on recent approaches to the SLAM problem suitable for real-time systems without access to Global Navigation Satellite Systems (GNSS).

Visual-inertial SLAM (VI-SLAM) [Cvišić et al., 2018, Zuo et al., 2020, Cadena et al., 2016, Mur-Artal and Tardós, 2017] and visual-inertial odometry (VIO) [Delmerico and Scaramuzza, 2018] approaches estimate ego-motion from the image streams of a lightweight mono- or stereo-camera, enabling the dimensions of robots to be minimized and allowing for deployment in unknown environments. However, visual methods require feasible lighting conditions [Alismail et al., 2017], which cannot be guaranteed during a disaster scenario. The mapping pipeline of visual approaches associates salient visual features, which are, however, unfeasible for other high-level systems, such as spatial planning and spatial exploration. This problem can be overcome by estimating the depth in a stereo-camera image stream, which is however, a computationally expensive process and limits the perception range to the resolution of the camera that is used. These two drawbacks make the sole use of visual systems for SLAM in S&R scenarios insufficient. Recent visual SLAM methods are analyzed and are well summarized in [Abouzahir et al., 2018] and in [Huang, 2019].

More robust methods with respect to lighting conditions use active laser sensors (LIDARs) as the primary source for ego-motion estimation. Laser-inertial SLAM (LI-SLAM) may employ various types of laser sensors, ranging from 2D methods [Petrlík et al., 2020] and 3D methods utilizing 2D LIDAR [Petráček et al., 2020] to approaches utilizing comparably heavyweight 3D sensors [Zhang and Singh, 2014, Ye et al., 2019, Geneva et al., 2018, Nieuwenhuisen et al., 2017]. The main advantages of LIDARs in SLAM are their high measurement range, the ability to measure the intensity of the reflected light beams and to estimate the reflectivity of the observed materials, as well as the high granularity of the data. The high granularity and the omnidirectional horizontal field of view allow more detailed volumetric maps to be constructed than when visual systems are used. These properties, however, come at the cost of a high computational load during processing and reasoning over the data. The high data throughput requires sufficient computational power onboard the robots, and efficient implementation of the entire SLAM pipeline. In section 3.2, we show that real-time performance can be achieved with efficient implementation of [Zhang and Singh, 2014] adapted for use on aerial vehicles.

2.2 Planning

Path planning is a widely studied problem in the field of robotics, and numerous approaches and their variants for 3D environments have already been developed [Yang et al., 2016, Mac et al., 2016, Zhao et al., 2018]. The approaches differ in the desired representation of environments, their time complexity as well as their optimality criterion, and their performance in varying environments. While many of the classical planning approaches aim to generate the shortest path or to maximize obstacle clearance, path planning for UAVs in cluttered unknown environments imposes special requirements. The requirements demand real-time performance and extended obstacle clearance to enable frequent replanning and to increase the robustness of the system, while maintaining the ability to travel through narrow passages and also to navigate within large open spaces.

Due to the high computational complexity of planning algorithms in a 3D environment, some of the algorithms applied for the planning of UAV motion in a cluttered environment decrease the computational complexity by reducing the number of dimensions of the planning space [Quan et al., 2020, Heng et al., 2014, Kothari and Postlethwaite, 2013]. In [Heng et al., 2014], the authors build a 3D occupancy map as a representation of an environment, but they use only 2D slices of this map for the planning. In [Kothari and Postlethwaite, 2013], the authors present a fast planning method based on RRT*, but their method is also limited to two-dimensional space. Reducing the planning space decreases the computational time and hence enables online planning, but it suppresses the advantage of the larger operational space of UAVs. The methods can therefore become inapplicable with increasing complexity of the environment.

Approaches using a 3D representation of environments are therefore favourable for UAV path planning. For outdoor applications, the authors in [Nieuwenhuisen and Behnke, 2015] presented a grid-based planning method with gradient-based motion optimization for planning in large outdoor areas. A similar problem is addressed in [Achtelik et al., 2014], where the authors introduced a path planning approach that respects the uncertainty of localization in featureless parts of the environment. However, the computational time, including the pre-processing phase of the environment representation, does not allow the use of these methods for online planning in unknown environments.

Methods focused on constrained indoor environments mostly aim to generate trajectories taking into account kinematic and dynamic constraints, leading to the introduction of planning approaches employing optimization methods [Ratliff et al., 2009, Richter et al., 2016, Usenko et al., 2017, Zhou et al., 2019, Mohta et al., 2018]. In contrast to traditional grid-based methods, this approach enables the application of kinematic and dynamic constraints on the resulting trajectories. As a result this enables the generation of high-speed trajectories. This advantage is fully exploited in [Mohta et al., 2018] where the authors present a trajectory generation approach for flying in an indoor environment at speeds up to $7\,\mathrm{m\,s^{-1}}$. However, the approach utilizes a bounded map of the environment for local planning and 2D environment representation for global planning. The trajectory-optimizing approaches to high-speed navigation in unknown environments proposed in [Gao et al., 2018, Zhou et al., 2019, Zhou et al., 2020] shows very impressive results in constrained indoor environments, as well as in small-scale real-world experiments in forests. The authors use the optimistic assumption that the unknown part of the environment is free of obstacles. Although this assumption is advantageous and enables navigation to a goal located in an unknown part of the environment, it decreases the safety of the system. Since the approach produces paths which can potentially collide with obstacles, any delay in the production of a replanned path or temporary inability of UAV to perform an avoidance manoeuvre will result in collision with an obstacle.

The most suitable approaches for the intended application of S&R are presented in [Cover et al., 2013, Droeschel et al., 2016, Lin et al., 2018, Perez-Grau et al., 2018, Zhou et al., 2020]. In [Perez-Grau et al., 2018], the authors apply an online planning approach with frequent replanning based on the lazy theta star algorithm. However, the approach does not implement any mechanism to increase obstacle clearance, which is a crucial property of planning algorithms for application in real-world scenarios. The approach in [Droeschel et al., 2016] respects the requirements on obstacle clearance. However, it uses a known map of the environment to generate an initial path with a grid-based planner, and it applies the local planner only for avoiding newly explored obstacles. In [Cover et al., 2013], the authors present a multi-layer planning approach that uses a sampling-based path-searching method to obtain a global plan and then uses a gradient-based optimization method to ensure trajectory smoothness and dynamic feasibility. A similar approach is presented in [Lin et al., 2018], where replanning is realized only locally in a receding horizon manner. The approach in [Zhou et al., 2020] addresses the previously mentioned problem of treating the unknown space as free and proposes the perception-aware replanning strategy which significantly increases the success rate of trajectory generation in comparison to previously published methods. Another way of approaching this problem is presented in [Tordesillas et al., 2019], where fast trajectories are generated in an unknown space while always having a backup trajectory exclusively within a known free space. Although the approaches mentioned here are well suited for particular applications, none of them complies with all of the requirements imposed by S&R missions in a completely unknown indoor environment.

The planning approach proposed for S&R applications in this work is exceptional in accounting for all the challenges imposed by real-world conditions, and also for its thorough testing and validation in complex real-world environments (see section 5). The method also does not impose any assumptions on the environment for the intended S&R scenarios and does not require computationally demanding pre-processing of the environment map, which further decrease the required planning time of local as well as global planning. These properties allow a rapid reaction to dynamic obstacles that may appear and also the exploitation of any newly-discovered unoccupied space in a continuously-built map for shortening the paths or for increasing the obstacle clearance. Although the proposed planning approach works with a grid-based representation of an environment, it does not require computation of the distance transform field for the planning. This further decreases the computational time required for planning, and thus enables more frequent replanning onboard UAVs. In addition, all parameters used in the proposed method directly express real-world constraints or requirements on obstacle clearance. The method is therefore not dependent on the tuning of any of its parameters. This makes it universally applicable to various environments without any change. A qualitative comparison of the proposed planning approach with the state-of-the-art planning approaches for UAVs is presented in Tab. 1.

Table 1: Comparison of other planning approaches for UAVs ([Petrlík et al., 2020], [Perez-Grau et al., 2018], [Droeschel et al., 2016], [Lin et al., 2018], [Cover et al., 2013], [Tordesillas et al., 2019], [Zhou et al., 2020]) with the planning approach proposed in this paper. Each feature was assessed on the basis of information obtained from presented papers,to the best of our knowledge and belief. Feature (4) stands for the requirement on pre-processing the representation of the environment for planning. Approaches marked with (×) use the output of the mapping modules directly. Feature (7) represents verification in an environment with no specific structures such as tunnels, a vast open space, or a forest. Feature (8) represents verification in an environment which has not been seen prior to deployment of the system, and it therefore does not enable any pre-tuning of the parameters. Feature (9) stands for verification in a dense environment with a large number of various obstacles. Feature (10) represents the ability of the algorithm to fly through narrow passages while maintaining bigger distance from obstacles in an open space.

Feature	ours	Petrlík	P.-G.	Droeschel	Lin	Cover	Tord.	Zhou
(1) Full 3D environment representation	✓	×	✓	✓	✓	✓	✓	✓
(2) Real-time performance	✓	✓	✓	✓	✓	✓	✓	✓
(3) Global planning on an evolving map	✓	✓	×	×	×	✓	✓	✓
(4) Map pre-processing not required	✓	×	×	✓	✓	×	×	✓
(5) Free from tuning coefficients	✓	×	×	×	×	×	×	×
(6) Real-world experiment	✓	✓	✓	✓	✓	✓	✓	✓
(7) Verified in general complex 3D env.	✓	×	×	✓	×	×	×	×
(8) Verified in completely unknown env.	✓	✓	×	×	×	×	×	×
(9) Verified in a cluttered environment	✓	✓	×	×	✓	×	✓	✓
(10) Narrow passages flight	✓	✓	×	×	✓	✓	✓	✓

2.3 Robotic Systems for Exploration and Inspection

Exploration-like applications of robot systems in real-world scenarios vary from an inspection of known structures such as chimneys [Nieuwenhuisen et al., 2017], bridges [Jung et al., 2019] or historical buildings [Saska et al., 2017, Krátký et al., 2020], through aerial monitoring of large areas [Yuan et al., 2015, Lindner et al., 2015, Rokhmana and Andaru, 2016, Casagli et al., 2017, Ma et al., 2018] up to S&R scenarios in completely unknown environments [Rouček et al., 2019, Petrlík et al., 2020, Miller et al., 2020]. These systems differ in the level of autonomy and also in their requirements for an external infrastructure.

In applications where the robots operate in an open space outdoor environment, many of these systems rely on teleoperation and on the use of GNSS. This type of setup is used in [Ma et al., 2018, Rokhmana and Andaru, 2016, Angermann et al., 2012], where UAVs are used to monitor agricultural areas affected by a natural disaster, to assess the state of a terrain damaged by an ammunition explosion, and to track the state of a recently-erupted volcano.

Teleoperated robot systems have also been deployed successfully in GNSS-denied environments. A system using Unmanned Ground Vehicles (UGVs), which is applied for inspecting contaminated areas of the Fukushima Daichi nuclear power plant after the natural disaster in 2011, is presented in [Nagatani et al., 2013]. In [Michael et al., 2012], the authors exploit the capabilities of cooperative UAV and UGV systems and present a similar system for mapping an earthquake-damaged building. With a similar motivation, the systems presented in [Kruijff et al., 2012] and in [Kruijff-Korbayova et al., 2015] were designed for inspecting post-disaster environments with the use of UAVs and UGVs in human-robot cooperation.

Several works introduce autonomous systems to overcome problems with teleoperation and communication with robots. The level of autonomy is increased in [Cantelli et al., 2013], where a formation of a UGV and a UAV is used in a surveying operation in humanitarian demining. The UGV that carries the UAV autonomously tracks the given path and transmits data to the operator, who decides about the necessity of autonomous takeoff of the UAV to provide additional information in obstructed areas. Fully autonomous behavior is achieved in [Qin et al., 2019], where the authors introduce a cooperative team of UGVs and UAVs capable of exploring a small office-like environment.

Algorithms for exploration applied to UAVs have also been intensively studied in recent years [Bircher et al., 2016, Dang et al., 2018, Selin et al., 2019]. However, most of these works aim at minimizing the distance travelled or exploration time required to explore the environment completely. This approach leads to a thorough exploration but does not maximize the explored space within a limited time, especially if the algorithm generates jerky trajectories, which are inefficient for UAVs. This problem is addressed in [Cieslewski et al., 2017], where navigation points are selected in the field of view of an onboard depth camera pointing in the direction of flight. This approach results in a smoother trajectory without sharp turns. This enables faster UAV motion and thus increases the explored space.

The deployment of UAV systems in cluttered indoor environments imposes additional requirements on the ability of UAVs to fly through narrow passages. This problem has been addressed in three different ways in recently published works. In [Falanga et al., 2019], the authors present a UAV with modifiable geometry, which increases the flight time by maintaining the most efficient shape whenever possible and diminishing its size in narrow passages. The second approach [Loianno et al., 2017] is based on aggressive manoeuvres. However, it is applicable only for window-like narrow passages and not for long narrow halls. The approach proposed in [Briod et al., 2014] introduces a collision-resilient UAV equipped with a mechanical structure that ensures resistance to collisions. However, all these approaches suffer from a short flight time and from strongly limited sensory equipment, which makes them unsuitable for exploring large and complex areas.

Although the works mentioned above are closely related to our system, they have tended to be tested in small office-like environments, without considering any external disturbances such as wind gusts, ventilators, smoke or dusty conditions, and variable illumination. The most closely related works from the perspective of real-world deployment and testing are [Petrlík et al., 2020] and [Fang et al., 2016]. In [Fang et al., 2016], an autonomous drone is applied to explore a vessel and to find fires. The approach is tested in a real-world scenario with narrow corridors and a hazy, smoky environment. The authors present several impressive experiments, but the navigation and planning approach is reduced to two-dimensional space. A state-of-the-art autonomous UAV system [Petrlík et al., 2020], designed for indoor aerial monitoring and inspection in S&R operations, is based on a lightweight aerial platform with a 2D mapping approach. Although the system shows high robustness and performance in the man-made tunnel environments of the SubT Challenge Tunnel circuit, its performance in general 3D environments is limited.

3 SYSTEM ARCHITECTURE

The architecture of the system presented here, including the interconnections among the components, is illustrated in Fig. 2. The underlying control pipeline is available open-source [2].

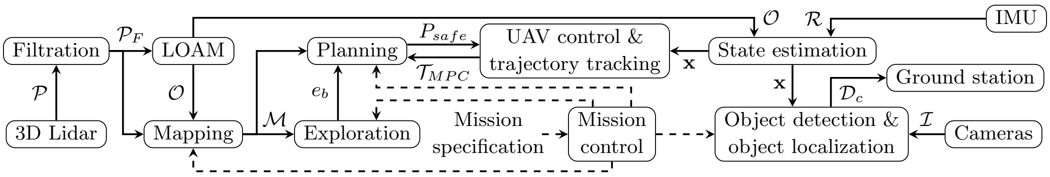


Figure 2: The architecture of the proposed system, where $\mathcal{P}$ stands for the point cloud obtained by 3D LIDAR, $\mathcal{P}_F$ stands for the filtered point cloud, $\mathcal{M}$ represents the map of the environment, e_b is the current best exploration node, P_{safe} stands for the planned path, $\mathcal{T}_{MPC}$ is the current trajectory generated by the MPC tracker, $\mathcal{O}$ is the odometry information, $\mathcal{R}$ are inertial measurements of the linear velocity, the linear acceleration, and the angular rate, $\mathbf{x}$ is the current pose estimate, $\mathcal{I}$ stands for images from cameras, and $\mathcal{D}_c$ represents confirmed hypotheses of object detections. Dashed arrows represent mission specifications and the commands produced by the finite state machine handling the mission control.

3.1 Environment Perception

To perceive and map the unknown dynamic environment for high-level subsystems, the UAV platform relies mainly on Ouster OS1-16 3D laser scanner (henceforth LIDAR). The LIDAR sensor measures distance in 16 vertical planes with horizontal resolution of up to 2048 samples per plane. It weighs 425 g and has a 120 m range, 33.2° vertical field of view, 10–20 Hz update rate, and 14–20 W power draw.

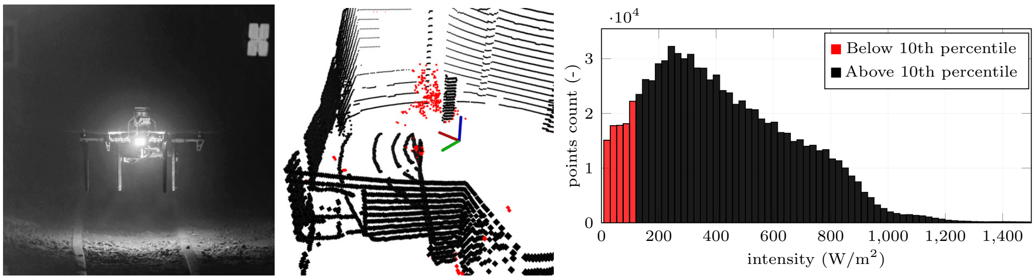
[2] https://github.com/ctu-mrs/mrs_uav_system

As has been verified experimentally, the targeted environments (mines, demolished structures, tunnels, abandoned buildings) can be characterized by a set of common features, namely by the presence of narrow passages and dynamic obstacles, by the presence of adverse and changing lighting conditions, and by the presence of whirling clouds of dust arising from wind gusts and from the aerodynamic influence of the UAV platform. To navigate an autonomous vehicle in such harsh conditions, the perception of the environment has to be extremely robust and adaptable, particularly for camera and LIDAR-based systems.

To improve the performance of all high-level systems in the proposed LIDAR-based perception, the raw data of the LIDAR are filtered with respect to the intensity of the measured points, as shown in Fig. 3. Assuming solely partial light energy reflection from a dust particle and significant energy dissipation of distant reflections, a local intensity filter is applied to a set of all points $\mathcal{P}$ such that a subset of points $\mathcal{P}_F \subset \mathcal{P}$ is preserved. The set is given as $\mathcal{P}_F = \mathcal{P}_D \cup \mathcal{P}_I$, where

$$\begin{aligned} \mathcal{P}_D &= \{p \,|\, ||p|| \geq \kappa,\, p \in \mathcal{P}\}\,, \\ \mathcal{P}_I &= \{p \,|\, \mathcal{I}(p) > \lambda_{10\%}(\mathcal{P} \setminus \mathcal{P}_D),\, p \in \mathcal{P} \setminus \mathcal{P}_D\}\,. \end{aligned} \tag{1}$$

$\mathcal{I}(p)$ is the intensity of the reflected light from point p, κ is a user-specified distance threshold (10 m was used for the robotic platform in experiments presented here), and $\lambda_{10\%}(\cdot)$ is the 10^{th} percentile of the intensity data. The 10 m threshold and the 10^{th} percentile are determined according to a set of empirical real-world tests with the OS1-16 3D laser scanner in dusty conditions. The idea behind the local filtering is to remove dust particles gusting through the surrounding air due to the aerodynamic influence of the fast-rotating propellers. As the aerodynamic influence of UAV is mostly local (hence the 10 m distance threshold), the filtration of data in the vicinity of the robot is sufficient for two reasons: the density of far dust clouds is low (assuming that there is no dust cloud generated by any other robot) and the probabilistic mapping filters out non-consistent spatial measurements in principle.



(a) Example of the targeted environment within the competition, including whirling clouds of dust and adverse lighting conditions

(b) Presence of a dust cloud (red) in a single-scan data item prior to intensity-based filtration (black & red) and after filtration (black)

(c) Histogram of the intensity of the reflected light beams

Figure 3: Data pre-processing of Ouster OS1-16 LIDAR based on light beam energy. 10 % of the points with the least reflected light intensity within distance κ are removed. The condition on distance κ preserves distant reflections with dissipated energy.

3.2 Localization and Mapping

For *localization* within an unknown environment, the LIDAR Odometry and Mapping in Real-time (LOAM) algorithm [Zhang and Singh, 2014] of SLAM is adapted for the use with OS1-16 LIDAR and tuned for the use in the position control feedback loop running onboard a fast-moving UAV. This method first estimates the odometry of the LIDAR from two subsequent data frames. Then, it matches and registers this initial estimate to update its inner representation of a map (a point-cloud of features). The method considers the intrinsic motion of a fast-rotating LIDAR, which is compensated by a constant angular velocity model of the LIDAR. By applying this model in the odometry estimation, which matches salient feature points that are on the edges and on the planar surfaces using a non-linear variant of the Iterative Closest Point (ICP) algorithm, the method has been shown to improve the robustness of the feature registration process onboard a fast-moving vehicle. Although the method does not provide loop closure during the map registration phase, it yields minimum drift, as evaluated on the Kitti dataset, where the method ranks second in accuracy among more than a hundred algorithms[3]. The qualitative and quantitative performance of the localization methodology during an internal experimental validation, as well as during the DARPA SubT competition is available in section 5. The analysis is further supported with attached multimedia materials. The above-mentioned tuning consists of setting parameters of the method (mainly the resolution of the feature map) to yield accurate results while still being able to run in real-time onboard a light-weight UAV.

[3] `http://cvlibs.net/datasets/kitti/eval_odometry.php` (visited on May 6, 2020)

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021

For online *mapping* of the environment, a single global probabilistic volumetric grid is continuously built given the estimated transformation between two consecutive LIDAR frames and the filtered LIDAR data. In the inner octree representation of the map, the voxels are updated according to the Bayes theorem, while higher-level systems access this representation in the form of a 3D occupancy octree. The inner map representation is granular enough to preserve sufficiently detailed information, while the depth of the output octree is pruned to a certain depth, representing the desired resolution of the map. The dynamic depth of the octree hence allows for a map representation with adaptive resolution. The occupied and unoccupied cells of this map then serve as a baseline for safe collision-less navigation within a previously unknown environment (see section 3.4).

The volumetric mapping fuses all the onboard sensors whose data can be converted to a 3D point cloud (e.g., time of flight cameras or depth cameras) to maximize the environmental perception capabilities of the robot. Such integration of data outside the field of view of the main LIDAR used for localization consequently enables efficient exploration of vertical shafts and small rooms, which could not be ensured using only a horizontal LIDAR with a small vertical field of view. This exact scenario is showcased in Fig. 4, where an upwards-oriented depth camera enables the exploration of a vertical shaft.

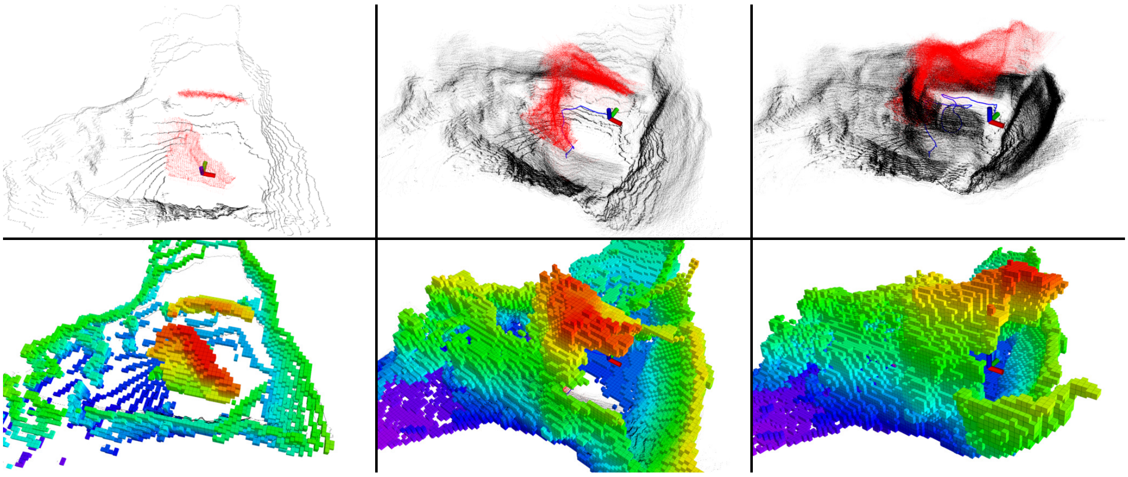


Figure 4: Modularity of the sensory setup employed in the volumetric mapping (bottom row). Example of two sensors (rigidly attached onboard a UAV) used in exploration of a vertical shaft — a horizontally-mounted OS1-16 LIDAR and an upwards-oriented RealSense D435 (the aggregated data are highlighted in black and red in the upper row). Presented data come from a real-world experiment performed in Byci skala cave system in Moravian Carst.

3.3 Automatic Control and Tracking

The concepts of UAV stabilization, control, and state estimation utilize our previous research on the stabilization of autonomous aerial vehicles [Báča et al., 2018, Petrlík et al., 2020, Báča et al., 2021]. The control pipeline uses a model predictive control (MPC) trajectory tracker [Báča et al., 2018, Báča et al., 2021] (MPC tracker), which takes the desired trajectory and produces a feasible reference obeying the model dynamics up to snap and jerk derivatives. This reference is supplied to the MPC controller [Petrlík et al., 2020, Báča et al., 2021] to determine the desired linear and angular acceleration for an acceleration controller. The acceleration controller then takes the desired acceleration and the current attitude and orientation from the state estimation pipeline to produce the desired angular rate and thrust reference for an embedded attitude rate controller. The state estimation pipeline fuses the inertial measurements from the onboard inertial measurement unit (IMU) with an estimate of the local or global state. In our case, this state estimate is the localization output of the approach described in section 3.2.

3.4 Planning

A navigation approach suited for the S&R missions has to combine real-time responsiveness in dynamic, diverse, and complex 3D environments with fast global planning to effectively and safely exploit the limited time of flight of a UAV. For this purpose, a grid-based planning algorithm with iterative path post-processing was designed and was integrated into the complex system presented here.

The inputs for the planning procedure are the map of the environment in the form of a 3D volumetric occupancy grid

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021 10

and the desired start and goal positions, which are updated in real time by the online mapping and high-level exploration modules. Given these inputs, an informed heuristic search is applied to find the initial path between the start and goal positions.

In contrast to traditional grid-based path-finding approaches, several changes were introduced to improve the real-time performance of the hereafter presented algorithm. Firstly, the point cloud formed by centers of occupied and unknown cells in the global 3D volumetric occupancy grid is converted to a KD-Tree structure forming the representation of obstacles. The KD-Tree representation of a point cloud allows for a fast search of the nearest occupied voxel. This information is used to conclude the feasibility of particular cells during the heuristic search. This step enables us to completely avoid the computationally expensive obstacle growing procedures at the beginning of each planning iteration and thus to use the single global map for path planning. However, this shifts part of the computational requirement of the pre-processing phase to the planning phase. In practice, this significantly reduces the total computation time. This is especially true for simpler planning instances that do not require search over a significant part of the environment.

The informed heuristic search applies Euclidean distance as heuristics and uses the set of all 26 adjacent voxels as the neighborhood of a single voxel. This set of cubical voxels sharing at least one common face, edge, or corner is further referred to as the ”26-neighborhood” in this paper. To continue to improve the real-time capabilities of the graph search algorithm, the number of expanded nodes is decreased by pruning the expansions of cells that cannot lead to a better solution. This procedure is based on the node-pruning approach within Jump Point Search (JPS) algorithm [Harabor and Grastien, 2011], but the rules are applied on the three-dimensional grid. The influence of node pruning on the computational requirements of the path searching algorithm is shown in Fig. 10.

In contrast to the vanilla versions of grid-based algorithms, our method further introduces a mechanism to improve the overall applicability and robustness in real-world autonomous systems working with incomplete and noisy data. The proposed mechanism is based on sub-path production and an estimate of the proper local movement direction if the global plans cannot be achieved (due to as-yet unavailable or sparse sensory data) or if a specified planning timeout is exceeded (1 s timeout is used in the presented experiments). In both cases, a sub-path is generated to an already expanded node with the minimal Euclidean distance to the goal. This approach prevents idle hovering caused by repeated attempts to find a path to unreachable exploration nodes, while it is sufficient to reach their vicinity.

Required preliminary knowledge of environment properties is another issue of state-of-the-art methods related to their deployment in real-world unstructured workspaces. Classical occupancy grid-based approaches require a parameter that represents the minimum allowed distance between the center of the robot and the nearest obstacle. The threshold usually incorporates the dimensions of the robot as well as an additional safety distance representing the required obstacle clearance. The distance from a particular node to the nearest obstacle is then compared to the threshold, and the result implies the tractability of the cells. Since the threshold has to be specified prior to the application of the algorithm and is usually constant for the entire path, it is difficult to define this threshold for an unknown environment. The threshold has to be set to a sufficiently low value to enable navigation through narrow corridors (e.g., doors or narrow halls), which might be essential for reaching parts of the environment. However, this distance between a UAV and obstacles is maintained throughout most parts of the generated path. The threshold should therefore be sufficiently high to avoid generating paths that are unnecessarily close to obstacles, and that therefore reduce the safety when navigating a UAV within the environment. We therefore propose and apply an iterative path enhancement process that increases the obstacle clearance in all parts of the generated path up to a minimum distance, while maintaining the ability to pass through narrow corridors. The minimum UAV-obstacle distance is based on the size of the UAV and the sensing and controller performance. The pseudocode of this iterative process is presented in Algorithm 1.

As the first step of the path enhancement process, a KD-Tree is generated from the point cloud representation of the environment in the surroundings of the initial path. Then, an enhanced new path P_{new} is found in the following way. The algorithm consecutively passes all nodes forming the initial path, and for each node that is closer to the nearest obstacle than distance d_s, it finds its neighboring node N_{best} with the maximum distance to the nearest obstacle. When node N_{best} is found, the algorithm checks whether it is in the neighborhood of the last node in P_{new}. If yes, N_{best} is inserted at the end of the list of nodes that forms P_{new}. Otherwise, nodes connecting N_{best} with the last node in the new path with maximum distance from the obstacles are inserted into P_{new} together with N_{best}. Thus the connectivity of the path after each iteration is preserved. When the last node of the initial path is reached, the algorithm proceeds to the next iteration in which the previously found P_{new} substitutes the initial path. The algorithm ends when the maximum number of iterations is reached, or when the path has not changed during the iteration.

The function *getBestNeighbor*(N), in Algorithm 1, returns a node from the set of adjacent 26 nodes of node N with the maximum Euclidean distance to the closest obstacle or the node itself if N is further from the obstacle than any of its

Algorithm 1 Pseudocode of the iterative path enhancement process enlarging the obstacle clearance of a given path.

```
procedure ENLARGE_OBSTACLE_CLEARANCE
    Input:
        $\mathbf{P}_{A*}$                                ▷ the original A* path
        $\mathcal{O}_{tree}$                             ▷ KD tree representation of environment
    Output:
        $\mathbf{P}_{safe}$                              ▷ path with enlarged obstacle clearance
    $n\_iteration \leftarrow 0$
    $path\_current \leftarrow \mathbf{P}_{A*}$
    while $n\_iteration < max\_iteration$ do
        $path\_next \leftarrow []$
        $path\_next.Add(path\_current[0])$
        for $node$ in $path\_current[1 : end]$ do
            if $node$ in $path\_next$ then               ▷ current node is already in plan
                continue
            if dist($node$, $\mathcal{O}_{tree}$) $\geq d_s$ then   ▷ current node is far from obstacles
                if isInNeighborhood($node$, $path\_next[end]$) then
                    $path\_next$.Add($node$)
                else
                    $cn \leftarrow$ getConnectingNodes($node, path\_next[end]$)
                    $path\_next$.Add($cn$)
                    $path\_next$.Add($node$)
                continue
            $N_{best} \leftarrow$ getBestNeighbor($node$)
            if $N_{best} = path\_next[end]$ or $N_{best} = node$ then
                continue
            else if not isInNeighborhood($node, path\_next$) then
                $con\_nodes \leftarrow$ getConnectingNodes($path\_next[end]$, $N_{best}$)
                $path\_next$.Add($con\_nodes$)
            $path\_next$.Add($N_{best}$)
        if $path\_current = path\_next$ then
            break
        $path\_current \leftarrow path\_next$
        $n\_iteration \leftarrow n\_iteration + 1$
    $\mathbf{P}_{safe} \leftarrow path\_current$
    return $\mathbf{P}_{safe}$
```

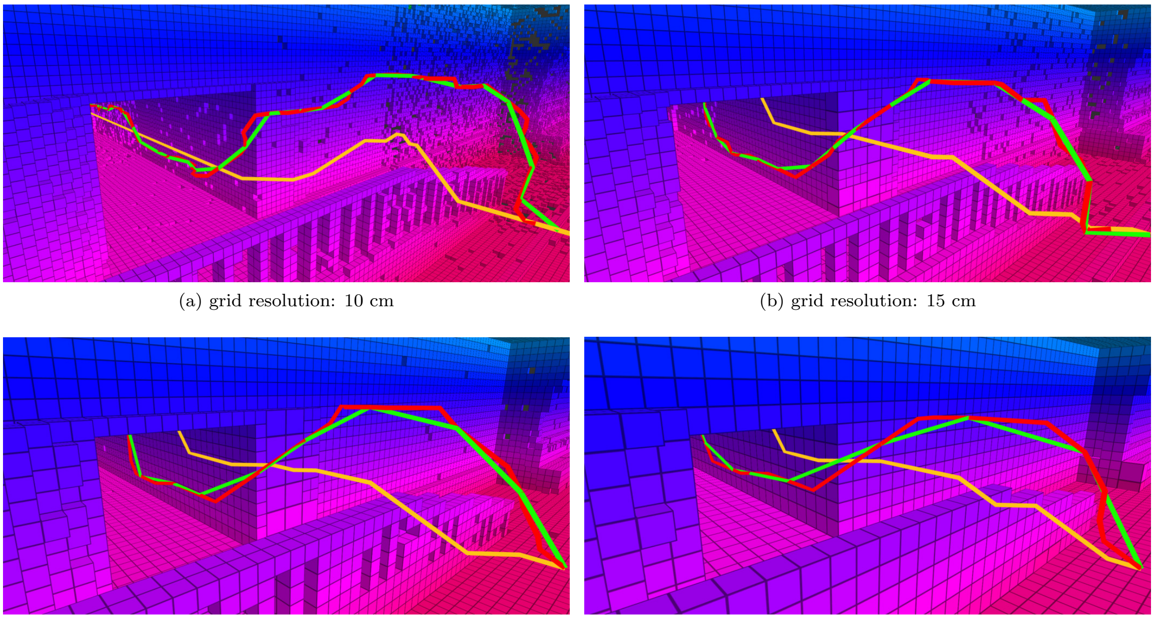
neighbors. If the distance to the closest obstacle is equal for two nodes, then the secondary key is defined as

$$R(N_n) = -\text{dist}(N_n, N_{prev}), \tag{2}$$

where N_{prev} is the last node on the current path and nodes with a greater value of $R(\cdot)$ are preferred. The secondary key helps to prevent path lengthening while not increasing the distance from obstacles. The function $getConnectingNodes(\cdot)$ returns the sequence of nodes that connects two nodes given as arguments with the greatest minimum distance from obstacles. The function considers only the sequences containing the lowest possible number of connecting nodes, which means a single node or pair of nodes in the applications of this paper. The proposed algorithm maintains the connected path at the end of each iteration and, in each iteration, every node can be shifted only to one of its neighboring nodes. Therefore the function $getConnectingNodes(\cdot)$ is always called on a pair of nodes whose predecessors were the neighbors in the previous iteration. Therefore, the predecessors (eventually one of the predecessors) compose the lower bound on the best feasible connecting nodes. Consequently, given the initial feasible path, the function $getConnectingNodes(\cdot)$ always produces a connection resulting in a feasible path with minimum distance from obstacles equal to or greater than the previous iteration.

Due to noise in the 3D LIDAR data, imprecision of the real-time localization and mapping module, and the discretization deficiency of the global map, the path generated by Algorithm 1 is often noisy and requires further post-processing. For this purpose, we apply a simple filtering approach where any two nodes in the path that are closer than a user-specified threshold d_f and are at a mutual distance along the generated path smaller than the size of a filtering window are removed from the path. Since no collision checking is applied during the filtering phase, the threshold d_f has to be set so that the filtering approach can only decrease the safety margin and cannot lead to a collision. The post-processed path is then segmented and is sent to the MPC-based trajectory tracking system. The tracking system then produces a smooth reference taking into account the kinematic and dynamic constraints of the aerial platform. An example of a path in particular processing phases is shown in Fig. 5.

The crucial part of path planning in unknown dynamic real-world environments is reactive replanning, which is particularly important for agile UAVs. Within the presented system we apply the replanning approach (see Fig. 6 for the scheme), which is triggered by two distinct events. First, replanning is triggered after a collision is detected between the planned trajectory and a newly discovered environment (dynamic obstacles). This is handled by a 2 Hz low-level loop that checks the feasibility of the trajectory generated by the MPC tracker with respect to the current map on the t_f seconds long horizon. If the trajectory is evaluated as infeasible, tracking of trajectory is stopped, and a new path



(a) grid resolution: 10 cm

(b) grid resolution: 15 cm

(c) grid resolution: 20 cm

(d) grid resolution: 30 cm

Figure 5: A visualization of the initial A* path (yellow), the enhanced path increasing the obstacle clearance (red), and the final filtered path (green) for maps with varying resolution.

starting at the current position is generated. Second, replanning is called periodically every t_p seconds (3 s are used in presented experiments) to exploit a newly-discovered non-collision space, allowing the path to be shortened or the obstacle clearance to be extended even if the previous plan does not collide with the obstacles. The periodic replanning is introduced in order to decrease the computational load of the onboard computer.

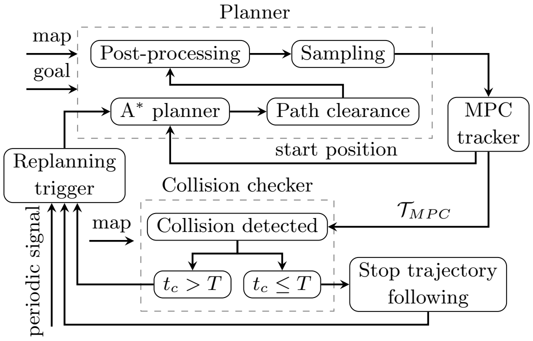


Figure 6: Illustration of the online replanning approach described in section 3.4. $\mathcal{T}_{MPC}$ is a trajectory produced by the MPC tracker, t_c is the length of the time horizon in which the first collision is detected, and T is the threshold for an immediate stop of the trajectory following, and for restart of replanning from the current position.

To ensure smooth transitional behavior of the UAV during online replanning, the start point P_s is taken from the currently-tracked MPC-based trajectory which consists of 40 transition points on a prediction horizon 8 s in length. The point which will be reached in t_m is selected. Then the whole planning procedure is performed, including the initial grid-based planning. Although starting this pipeline with Algorithm 1 is sufficient to get a feasible path, running the planning pipeline from its beginning fully exploits newly discovered free cells and therefore shortens the path. A new path is then composed of the remaining segments of the MPC trajectory up to P_s, followed by the filtered path generated by the planning algorithm.

The limited time for global planning mostly does not allow for planning paths longer than tens of meters. This becomes insufficient for planning the return to starting position, as the starting position can be much further from the final position reached during the exploration phase. To overcome this problem, a global-navigation homing tree is iteratively built from the poses visited by the UAV during the exploration phase. This homing tree is used to estimate the needed time to return, and to find a sequence of temporary planning goals successively leading the UAV to a starting position.

The proposed planning approach can hardly be compared with existing planning approaches, since it does not aim to be optimal in relation to the length of the path or in relation to maximization of the obstacle clearance. However, it fulfills all the requirements on a planning system that can be used in general real-world environments, such as real-time performance, event-triggered replanning, independence from environment characteristics, and the generation of robust paths with respect to uncertainties in localization, mapping, and control, all of which are inevitable in the real-world. It further reveals several additional advantages over recently published works. First, the only variable parameters that constrain the method are the requirements on obstacle clearance, on the smoothness of the path and the timeout for planning. These parameters explicitly depend on the dimensions of the UAV, the performance of the controller and the available computational power. This makes the method independent from the environment characteristics and it remains dependent only on the specifications of the vehicles. Second, the proposed planning method utilizes the output directly from the mapping pipeline, and does not require any computationally demanding pre-processing before the start of the planning algorithm. This enables fast global replanning, and therefore fast reactions to newly discovered parts of the environment and dynamic obstacles. The short replanning period also facilitates the use of adaptive map resolution. This enables the use of a high-resolution map for flights through narrow corridors while maintaining low computational demands in large open spaces, where lower resolution is sufficient. Last but not least, enlarging the obstacle clearance in open space areas of the environment significantly increases the robustness of the entire system.

3.5 Autonomous Exploration

Another crucial part of a fully autonomous mission in an unknown environment is the automatic exploration executed by following paths generated between a set of navigational goals (exploration nodes). In the proposed system, the basic set of exploration nodes E_b is identified in the volumetric map provided by the mapping subsystem as

$$E_b = \{c \,|\, N_{oc}(c) = 0, N_{un}(c) \geq T_{un}, c \in \mathcal{O}_{free}\}, \tag{3}$$

where $\mathcal{O}_{free}$ is the set of all free voxels in the voxel occupancy grid, $N_{oc}(\cdot)$ and $N_{un}(\cdot)$ are the number of occupied and unknown cells in the 26-neighborhood of a particular cell, and $T_{un} \in \{1, 2, \dots, 26\}$ is a user-specified threshold ($T_{un} = 16$ for the presented experiments). To reduce the number of exploration nodes, the nodes are clustered according to their mutual Euclidean distance. To avoid unnecessary computational demands and to achieve real-time performance, a reduced set of exploration nodes E_r is created by iterating over cells in E_b and applying the following rule

$$E_r = \begin{cases} E_r & \text{if } \operatorname{dist}(c, E_r) \leq T_{dist}, \\ E_r \cup c & \text{otherwise}, \end{cases} \tag{4}$$

where $\operatorname{dist}(c, E_r)$ stands for the Euclidean distance between the center of the cell c and the center of the closest cell in set E_r. This grouping method can be understood as a simplified version of the DBSCAN spatial clustering method [Ester et al., 1996]. The clustering procedure is followed by removing the set of already-visited exploration nodes E_v. This process is defined as

$$E_r = E_r \setminus \{c_g \,|\, \operatorname{dist}(c_g, E_v) \leq T_{dist}, c_g \in E_r\}. \tag{5}$$

Given the set of E_r, a simple cost function is applied to evaluate the exploration nodes. The cost function is defined as

$$\begin{aligned} E_{cost}(c) = {} & \alpha \operatorname{dist}(c, R) + \beta \operatorname{dist}_{xy}(c, R) + \gamma \operatorname{dist}_z(c, R) + \\ & + \delta \operatorname{adiff}_{xy}(c, R) + \epsilon \operatorname{adiff}_z(c, R), \end{aligned} \tag{6}$$

where R is the current pose of the UAV, $\operatorname{dist}(\cdot)$, $\operatorname{dist}_{xy}(\cdot)$ and $\operatorname{dist}_z(\cdot)$ are functions returning the Euclidean distance, the Euclidean distance omitting the difference in height, and the difference in height respectively. The $\operatorname{adiff}_z(\cdot)$ is a function returning the angle between the ground plane and the line segment defined by two points given as an argument of function. α, β, γ, δ, and ϵ are the weights of particular summands. The value of $\operatorname{adiff}_{xy}(\cdot)$ is given as

$$\operatorname{adiff}_{xy}(c, R) = \operatorname{mindiff}(\operatorname{atan2}(c_y - R_y, c_x - R_x) - \varphi), \tag{7}$$

where φ is the preferred direction of exploration. The value of φ can be set to either a constant value equal to the user-defined preferred direction of exploration or to the direction given by the last two used exploration goals.

Function $\operatorname{mindiff}(\cdot)$ is used for conversion of the difference in angles to interval $\langle 0, \pi)$. The function is defined as

$$\operatorname{mindiff}(\varrho) = \begin{cases} |\varrho| & \text{if} \quad |\varrho| \leq \pi, \\ 2\pi - |\rho| & \text{otherwise.} \end{cases} \tag{8}$$

The next exploration node c_{best} is then chosen with respect to the cost function defined in (6) as

$$c_{best} = \arg\min_{c \in E_r} E_{cost}(c). \tag{9}$$

The coefficients used in the cost function defined in (6), together with the choice of a preferred direction of flight φ, enable UAVs to have different behaviors even though they are running the same exploration approach. This option is advantageous especially when certain UAVs participating in the mission are not able to communicate directly with each other due to the environment constraints that are usually encountered in subterranean environments. During the exploration phase of the mission, the UAV switches to a new exploration node after reaching the previous exploration goal, or if the planner does not find a path to the exploration goal or has produced a sub-path of a zero length. The analysis of the computational requirements of the method for generation of exploration nodes is provided in Fig. 12. The computational time is independent to the size of the map as the search space for new exploration nodes is limited based on the assumption that new exploration nodes cannot be generated outside the perceptive field of view.

3.6 Artifact Detection

One of the main goals of UAVs in S&R scenarios is to find objects of interest (usually victims and their belongings), and to localize them in an unknown environment. Although the primary focus of this paper is on planning and exploration, artifact detection is a computationally expensive process that adds additional load on the available computational resources of the system. In the SubT Challenge, the objects of interest need to be detected and localized in relation to the entrance to the facility within an allowed error of 5 m. Precision in 3D coordinates enables a rescue team to be navigated to the location identified by an autonomous system. Objects of interest (called artifacts in SubT) may be identified in the visual spectrum (a cell phone, a backpack, a survivor in a protective vest, or a vent), or in the thermal spectrum (a survivor), or may also be identified non-visually (the WiFi signal of a cell phone, a carbon dioxide (CO_2) leak). To visually detect the artifacts, two RGB detection cameras with a combined 90° vertical and 180° horizontal field of view are carried onboard each UAV. In the object detection approach, the image is processed by a pre-trained YOLOv3 neural network (NN) object detector. The spatial detection hypotheses are then tracked in time, are localized in 3D space, and are eventually reported to other robots or back to the ground station. Object detection runs onboard the UAVs in real-time at 0.5 Hz, which is sufficient for flights up to $1\,\mathrm{m\,s^{-1}}$.

3.7 Mission Control

In order to keep the proposed system easy and fast to use, while preserving its ability to perform a range of tasks, we introduce a mission controller for high-level planning of the mission. It is implemented as a finite state machine with parametrized transitions providing configurable employment of various mission strategies. A simplified scheme of this state machine is presented in Fig. 7.

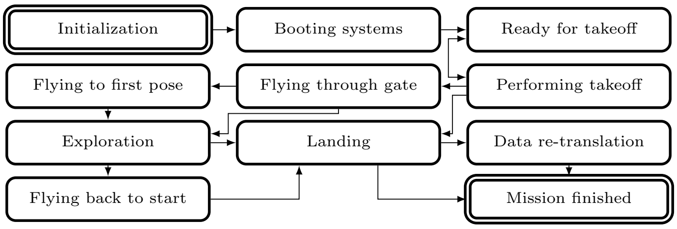


Figure 7: Illustration of a transition diagram among the most prominent blocks of the mission controller. The state machine can be adapted for various mission strategies, it supervises the operating conditions of all the important subsystems, and it may react to faulty scenarios by triggering a safety procedure such as controlled landing.

Use of the mission controller enables fully autonomous deployment of the system. It requires only two user actions. These are not necessary for correct behaviour of the system, but they widen the range of possible deployment scenarios. The first voluntary user action is to specify the desired landing position and to customize the requirements on returning to the start location, the first exploration position and other mission-related preferences. Setting these preferences allows us to configure the desired behavior of the UAV and enables the use of any available prior knowledge of the mission. This information may be gained from floor plans of a disaster environment (such plans are usually available for general S&R missions, but were not provided at the SubT Challenge) and from a visual analysis of the environment right at the S&R site. The second intervention is human-operator approval of the commencement of the mission. This serves as a safety feature to prevent an unexpected takeoff of the system. The human operator can also override the entire autonomous mission by pushing a safety button to trigger a safety procedure. Apart from the supervising the exploration mission, the mission controller manages the operation state of some subsystems that are not necessary for certain parts of the

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021

mission — the power state of onboard lights or restrictions on artifact detection in certain phases of the mission. This saves a certain amount of energy and therefore increases the available flight time.

4 System Performance Analysis

The real-time performance of all subsystems is essential for all robotic systems that are meant to be deployed in real-world environments. Several experiments supporting the statements about real-time performance of particular subsystems and several experiments analyzing the benefits of particular algorithms are presented in this section. For a more objective evaluation of the real-time performance of the whole system, the presented computational times were mostly measured on the onboard computer of the UAV during the experiments performed in real-world scenarios.

4.1 Localization and Mapping

As discussed in section 3.2, the employed SLAM approach does not perform loop closure detection, nor global optimization. This would typically be required in order to obtain global consistency, especially in complex subterranean environments. However, the results presented in section 5.1.1 and section 5.2 show the employed state estimation pipeline yields low drift. The advantages of global optimization in terms of global consistency and localization accuracy are undeniable, however the low-drift performance allows disregarding global optimization in order to reduce computational load. This highly correlates with the designed purpose of the proposed system suited for use on-board real-world UAVs, where constrained computational resources must be available to run all other real-time subsystems necessary for mid-air stabilization and navigation.

To show the resulting performance after the trade-off between consistency and accuracy, Fig. 8 shows the required computational time for procedures within the localization pipeline. The maximum overall computational times are close to 90 ms allowing it to run in real-time with only small time reserves for data coming at 10 Hz. Extending the pipeline with other computationally heavy features (e.g., a global optimization) would slack up the real-time performance of the system. This is the most crucial property on-board fast-moving aerial robots requiring real-time feedback in order to stabilize mid-air. Moreover, Fig. 8 also shows that the system is not dependent on the size of the map as the SLAM methodology optimizes only locally the incoming data. The quantitative analysis of the localization accuracy is provided further in Table 3.

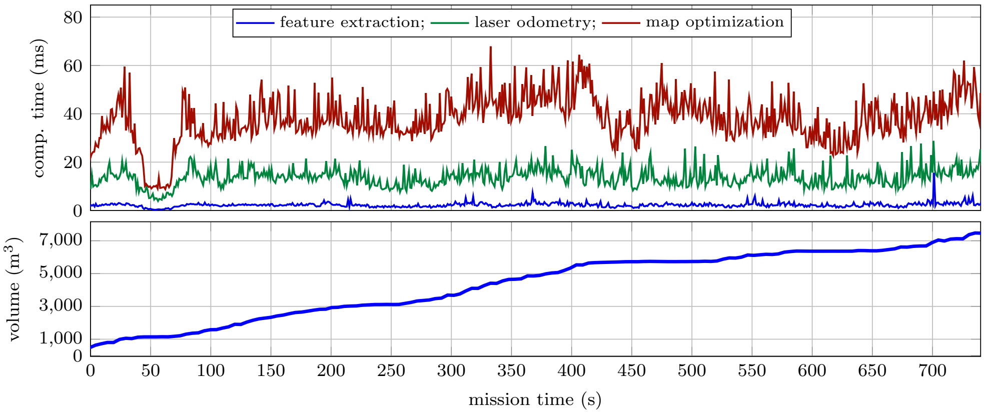


Figure 8: Computational time of particular procedures related to localization and mapping during 475 m long flight with an explored volume of 7463 m^3. The whole pipeline is triggered by incoming data with update rate 10 Hz. The average total runtime is 53.41 ms. The system was run on-board the UAV with computational unit Intel Core i7-10710U.

4.2 Planning

The real-time performance of the planning module is essential for performing frequent replanning of both local and global plans and thus essential for effective use of limited flight time of the UAV. The planning procedure can be divided into three steps — conversion of a 3D occupancy grid to KD-Tree, path planning, and path post-processing. The conversion of the 3D occupancy grid produced by a mapping module to KD-Tree structure is the only step required prior to the start of the planning phase. The detailed analysis of the dependence of the computational time required for this operation on the size of the environment is provided in Fig. 9. Data presented in this figure show that the computational requirements depend linearly on the size of the map. Although the computational time is tractable even after 15 minutes of flight time during which the aerial system flies 1200 m within a deep cave system, a large portion of the converted space would not be used by the planning algorithm. Therefore, only a local part of the environment is converted to the KD-Tree representation and consequently used in path planning. As shown in Fig. 9, this approach makes the computational time required for conversion independent on the size of space that has already been explored.

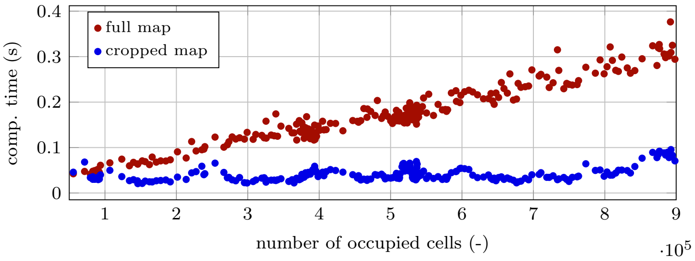


Figure 9: The dependence of the computational time required for conversion of the 3D occupancy grid to a KD-tree structure on the number of cells which are considered to be occupied in the original volumetric global map. The graph also shows the comparison between computational requirements for the conversion of the entire global map (a resolution of 20 cm is used in this experiment) and for the conversion of a local subspace surrounding only the given starting position (a cuboid of 120x120x40 m is used in this experiment). Data were gathered during a 15 minute long flight in the simulation environment, with the entire system running on Intel Core i7-10510U CPU.

To illustrate the computational gain or loss associated with particular modifications applied to a classical grid-based path searching algorithm, we present an analysis in Fig. 10. The analysis shows that the usage of precomputed distances from obstacles for all voxels in an occupancy grid speeds up the path searching process more than two times as much when compared to the proposed approach using KD-Tree for evaluation of the feasibility of particular voxels. However, this loss in computational efficiency is compensated for by the replacement of the computationally demanding obstacle growing algorithms with usage of the KD-Tree representation of the environment. For the method of building an occupancy map with inflated obstacles as applied in [Liu et al., 2017], the computational demand depends linearly on the size of set of points representing the occupied part of the environment. For an average number of occupied cells within the map during the autonomous exploration, the map conversion lasts approximately 70 ms. However, by considering the unknown part of the map as well, the computational time increases up to 10 s, which makes the method impractical for the real-time planning. On the other hand, the KD-Tree representation can be obtained without any approximations from a global voxel map within 100 ms even for maps containing $7.2 \cdot 10^7$ cells.

Replacement of the KD-Tree representing the entire environment by the KD-Tree representing only the subspace of the environment does not have such a significant influence on the path searching process as it does on the computational requirements in building the KD-Tree. This results is consistent with the theoretical asymptotic complexity of building a KD-Tree ($\mathcal{O}(n \log n)$) and searching for the nearest neighbor in the KD-Tree ($\mathcal{O}(\log n)$), where n is the number of nodes in the tree.

The significant improvement of the computational requirements were achieved by applying the pruning algorithm that effectively reduces the number of expanded nodes that cannot be part of the optimal path. This procedure speeds up the planning process by 49% on average. Detailed analysis presenting the average ratio between particular approaches for different complexity of planning problems is presented in Fig. 10. The presented results were generated on the set of planning problems solved during the deployment of the system in large-scale simulation scenarios.

The analysis of the computational requirements of the path post-processing method is presented in Fig. 11. The computational requirements of the iterative post-processing depends on the number of waypoints processed and also on the complexity of the environment, which further influences the length of the processed path in particular iterations and the total number of iterations. The data obtained during simulated flights shows that the whole post-processing, including the filtering step, does not exceed 64 ms for paths consisting of up to sixty way-points. For the setting used in most of the experiments ($t_p = 3$ s and grid resolution 0.2 m), this number of way-points enables post-processing of paths of a minimum length of 12 m that guarantees safety for allowed maximum velocities up to 4 m s^{-1}, which is above the reasonable maximum velocity for UAVs in an unknown cluttered environment.

The computational time required for the conversion of the 3D occupancy grid to KD-Tree, which is necessary to evaluate the distance of particular cells, is presented separately as the post-processing can make use of the KD-Tree previously converted for the purpose of global planning.

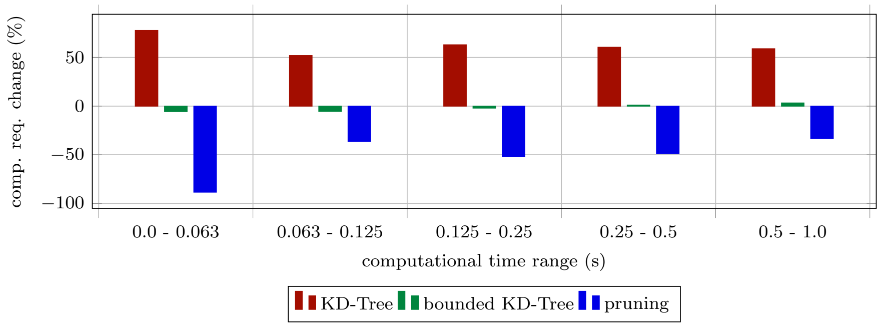
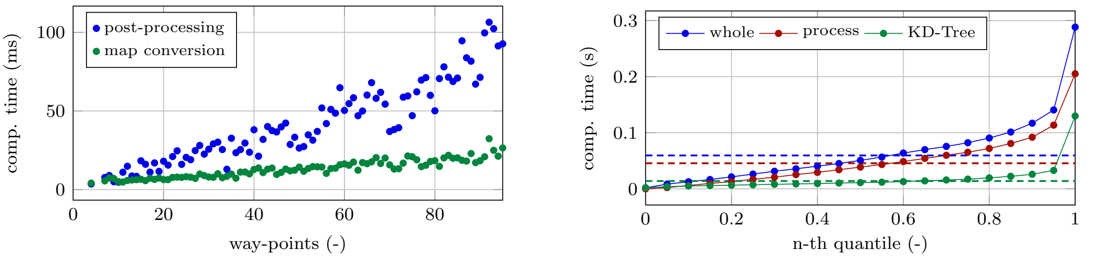


Figure 10: The analysis of influence of particular modifications of the path planning subsystem on computational requirements. The graph shows the average change of computational requirements achieved for the path planning method when applying node pruning (blue), by using KD-Tree representation of obstacles instead of a grid with precomputed distances from obstacles (red), and by using KD-Tree created only from subspace of the map instead of the whole map (green). The data are divided into several bins based on the computational complexity of the planning problem to show how important the particular modifications are for planning problems of different complexity. The computational time of the approach when not applying the particular feature is used as division criterion. The data were collected on a set of more than 300 planning instances corresponding to instances solved during autonomous missions.



(a) dependence of the computational time on length of the post-processed path

(b) quantiles of the computational requirements on the path post-processing gathered during ten simulation flights including paths of up to 100 way-points

Figure 11: The computational requirements of the proposed path post-processing. Figure (a) shows the dependence of the average computational requirements on the number of post-processed way-points. The green points stand for the computational requirements of the conversion of the 3D occupancy grid to a KD-tree structure. This operation is performed only on the surroundings of the initial path. Blue points show the computational requirements of the whole post-processing including the filtering. Figure (b) shows the distribution of the computational times required for path post-processing. The dashed lines in (b) stand for the average computational time. Data for both figures were gathered during flights in the simulator with total duration of 60 minutes which includes more than 1000 planning iterations. The simulation, including the whole system, was running on the computational unit Intel Core i7-10510U.

4.3 Exploration

The time performance of the exploration module was evaluated based on data collected during seven flights in real-world scenarios including urban and cave environments. The computational time required for getting the next navigation goal depends primarily on the portion of the unknown environment and thus on the size of the basic set of exploration nodes E_b. The distribution of the computational times needed for finding the next navigation goal is shown in Fig. 12.

5 Experimental evaluation

The presented system was evaluated in numerous simulations in the realistic Gazebo simulator, in experiments in simulated indoor and outdoor mockups, and finally in the real environment during the DARPA SubT Challenge Urban Circuit robotic competition. In this section, we analyze the results of these experiments and the performance of particular subsystems.

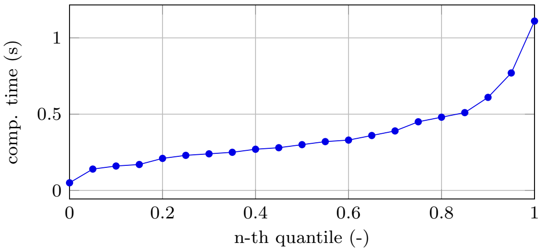


Figure 12: Distribution of the computational time needed for finding the next exploration node. The presented data were collected during seven flights in urban and cave environments with average duration of 10 minutes. The exploration module was run on an onboard computer with computational unit Intel Core i7-10710 along with the rest of the presented system.

5.1 Simulation Analysis

The influence of *path post-processing* (see section 3.4) on the output trajectory is examined with respect to obstacle clearance. For this purpose, multiple paths were generated in the virtual environment (see Fig. 16) containing dimension-constrained passages, steep corridors, and small obstacles. The quantitative results presented in Fig. 13 show significant growth in the obstacle clearance of the post-processed paths in comparison with the initial A* path. Although the filtering process may decrease the obstacle clearance in principle, the quantitative results do not show significant decrease in the average obstacle clearance after filtration has been applied to the post-processed path.

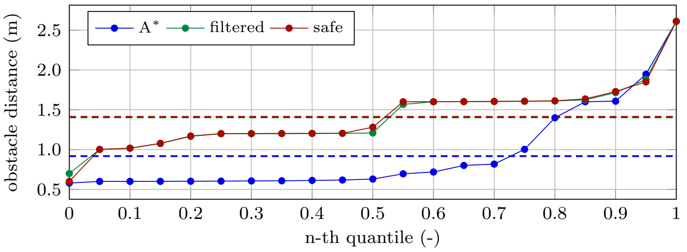


Figure 13: Influence of path processing (see section 3.4) on obstacle clearance. The data were obtained from the set of sampled paths generated in the constrained virtual environment (Fig. 16) with the desired obstacle clearance $d_s = 1.6\,\mathrm{m}$. The dashed lines show the average obstacle distance throughout all UAV positions on segmented paths.

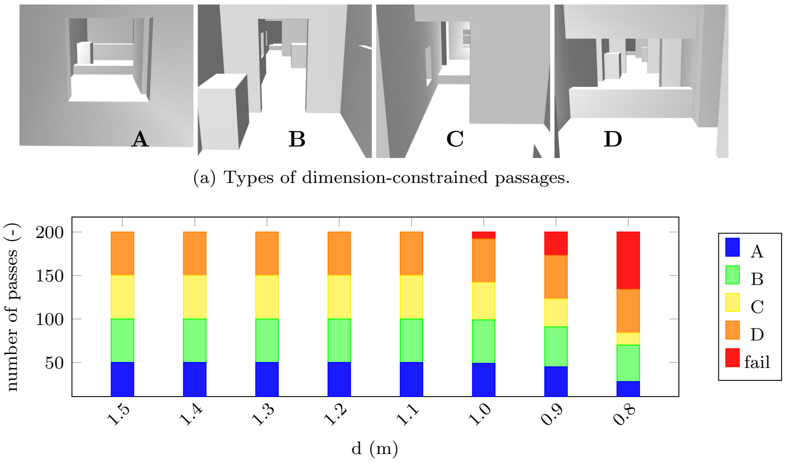
To examine the performance of the proposed system regarding *maneuverability within confined corridors*, an artificial scenario with various types of narrow passages was used for simulation analysis (see Fig. 14). This analysis showed the capability of the system to fly through passages that were extremely narrow in relation to the diameter of the UAV platform. The main cause of the unsuccessful attempts within the results was a physical collision emerging from inaccurate trajectory tracking due to external disturbances, constraints on the maximum allowed velocity and acceleration of the UAV.

To showcase the ability of the system to respond to the presence of *dynamic obstacles*, we present an experiment demonstrating how the path planning module behaves towards a moving obstacle. The experiment, shown in Fig. 15, displays the online path re-planning after a collision between the initial navigation plan and a newly-perceived obstacle has been detected.

5.1.1 Virtual Track of DARPA Subterranean Challenge

As a representative of *long-term complex validation*, we showcase the system performance on an autonomous UAV exploration of urban and cave indoor environments provided by DARPA for the virtual track of the competition. In the urban variation, a single UAV exploration using the proposed methodology is showcased. In the cave variation, a multi-UAV team is deployed to cooperatively explore the environment while the system runs independently unchanged on each robot apart from the parameters of exploration. A selected example of an urban world is shown in Fig. 16, where a two-floor structure combines narrow halls, stairs, steep corridors, a large open space, and obstacles such as a railing or lights hanging from the ceiling. The proposed system managed to autonomously explore the entire virtual object with a total space of $5819\,\mathrm{m}^3$ in 474 s, while traveling a distance of 207.75 m. The onboard-built map created by the autonomous UAV and its exploration trajectory within the environment are shown in Fig. 16.

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021



(a) Types of dimension-constrained passages.

(b) Dependence of the number of successful passes through narrow passages on the size of these passages *d*. Fail represents unsuccessful passes of a UAV with a physical diameter of 0.65 m through the constrained passages. Data were collected during flights through the constrained environment containing a sequence of dimension-constrained passages presented in (a). First fifty passes through each type of passage were included in presented results.

Figure 14: An analysis of the maneuverability of the presented system through tight passages with constrained dimensions. The analysis was performed using the realistic Gazebo simulator.

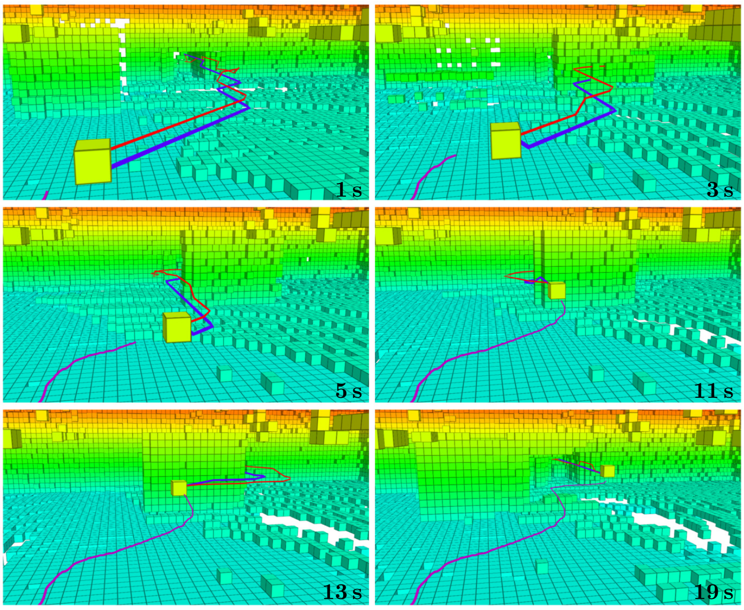
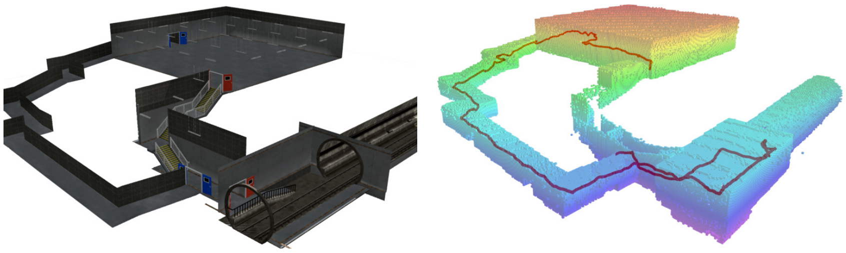


Figure 15: Navigation response to the detection of a dynamic obstacle. The experiment illustrates the reaction to the occurrence of a dynamic obstacle colliding with the current navigation path. As analyzed in Fig. 13, the figure shows the generated A* path (blue), the processed and filtered path (red), and the trajectory of the UAV (purple) up to the current UAV position.

A selected example of a cave world (taken from the set of virtual worlds of DARPA SubT Challenge) is showcased in Fig. 17. This environment consists of long, wide passages as well as narrow passages, vertical shafts, loops, and large open spaces. A heterogeneous team of UAVs [4] with different sensory setup and flight time was deployed within this testing world. In this experimental validation, no robot had communicated with other robots its exploration strategy nor reused any apriori information about the environment. This correlates with the requirements of real-world deployments assuming

[4] Specifications of robot highlighted in blue: 10 min flight time, OS1-16 LiDAR, upward and downward oriented depth camera. Specifications of robots highlighted in red and green: 20 min flight time, OS1-16 LiDAR.

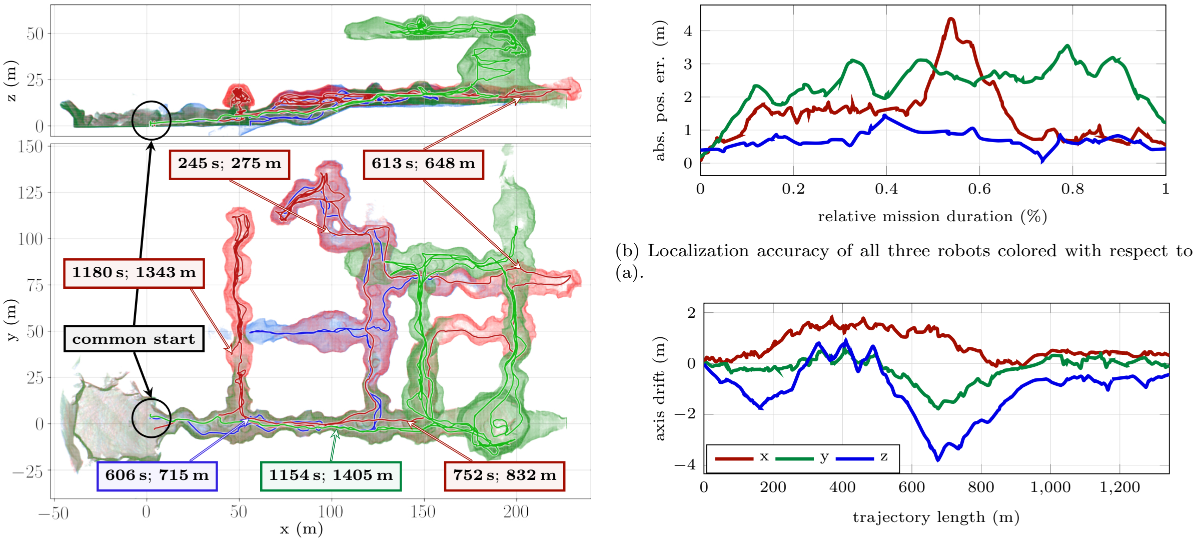
low reliability of any communication infrastructure, making the entire system applicable in fully communication-less applications. Within the cave exploration scenario in Fig. 17, all robots had explored a large portion of the environment over the course of their mission duration and returned back to communication range of the base station located next to the common start area. The robot highlighted in red in Fig. 17a had performed an approximately 650 m long trajectory loop starting at 180 s and ending by visiting a previously uncovered location at 750 s. As discussed in section 4.1, the pose and the map information is associated only locally without the need for global optimization, which is verified by the performance of the system in the trajectory loop of the robot highlighted in red in Fig. 17a.



(a) Virtual environment

(b) Onboard-built map

Figure 16: An urban environment used for simulation analysis of long-term exploration. The figure shows the simulated object (a), and the traveled trajectory of the UAV (red) within the onboard-built map (b).



(b) Localization accuracy of all three robots colored with respect to (a).

(a) Overview of the cave environment (the *Five Fingers* virtual cave world of the DARPA Subterranean Challenge) composed of three maps built onboard a team of aerial robots. The robot with a trajectory highlighted in red performed a round-way exploration and returned back to a previously visited location at approximately 752 s.

(c) Position drift in each axis of a robot highlighted in red in (a), where the robot spatially visited a previously uncovered location after 830 traveled meters. As our SLAM approach yields low-drift without a global optimization, the pose and the map information is associated to the old part of the environment using only the local optimization.

Figure 17: Autonomous exploration of a virtual cave environment by a team of three UAVs (distinctively colored). The figure shows the maps of the environment, the UAV trajectories, and the onboard-built maps of the cave built by a team of three aerial vehicles (a). The localization accuracy of the SLAM subsystem during this selected virtual mission is shown for all robots (b), while the position drift per each spatial axis of a single robot highlighted in red is also shown (c).

5.1.2 Comparison with a state-of-the-art approach

The importance of the proposed path post-processing applied for increasing the obstacle clearance is showcased on comparison with the state-of-the-art method FASTER [Tordesillas et al., 2019]. FASTER (Fast and safe trajectory planner) applies an optimistic assumption that the unknown part of the environment is free of obstacles to allow for the generation of trajectories lying partially in an unknown space. It also improves safety by maintaining a feasible and safe back-up trajectory in a known free space. FASTER was originally designed for use with a depth camera; however, the mapping module could be modified to enable the integration of FASTER into the presented system using the 3D LIDAR. The comparison is based on 50 flights of total length 2000 m through the environment presented in Fig. 14a with

© Wiley Periodicals LLC 2021. Accepted to Journal of Field Robotics. DOI: 10.1002/rob.22021

narrow passages of 100 cm width. The maximum velocity of both planners was limited, and the minimum UAV-obstacle distance was set to a similar value for both planners. The experimental results presented in Fig. 18 show that the herein proposed approach produces trajectories that are, on average, significantly further from the obstacles than the trajectories produced by FASTER. Moreover, the analysis showed that when working with the data provided by the applied mapping module, the FASTER algorithm is more computationally demanding than the approach proposed here. The average computational time consumed by our approach was 0.95 s, while one replanning step of the FASTER algorithm using the local map of size 8 m × 8 m × 4 m with resolution 0.1 m took 1.34 s on average. The complete statistics obtained from presented comparison are presented in Table 2.

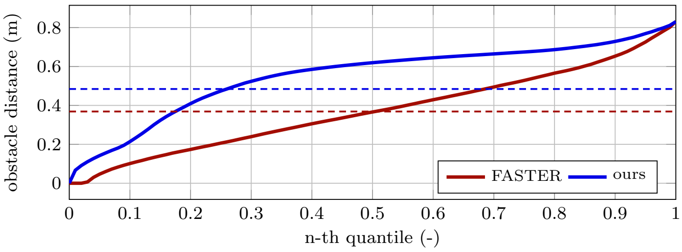


Figure 18: The comparison of the UAV-obstacle distance of the trajectories generated by FASTER algorithm [Tordesillas et al., 2019] and planning approach presented herein (see section 3.4). The data were obtained during the flights in the constrained virtual environment (see Fig. 14). The dashed lines show the average UAV-obstacle distance throughout all UAV positions.

We are aware that the main objective of the FASTER algorithm is not increasing the distance to obstacles but the generation of very fast but still safe trajectories. However, by making this comparison, we intend to demonstrate that even though the state-of-the-art approaches to trajectory generation show very impressive performance in simulation scenarios and also in artificial real world environments, they are not suited for all applications in an imperfect real world where UAV position estimation and distance to obstacles measurements are influenced by precision of real sensors. Particularly, this comparison shows the importance of increasing the distance to obstacles above the minimum threshold for safety and reliability of the whole system. All trajectories produced by the FASTER algorithm were collision-free with respect to the current map. Nevertheless, the natural imperfections of the mapping, trajectory tracking, and UAV control caused that the UAV following trajectories produced by FASTER resulted in 22 collisions over 50 flights, while the UAV following trajectories produced by the proposed approach encountered only three failures.

Table 2: Quantitative results of the comparison with FASTER algorithm.

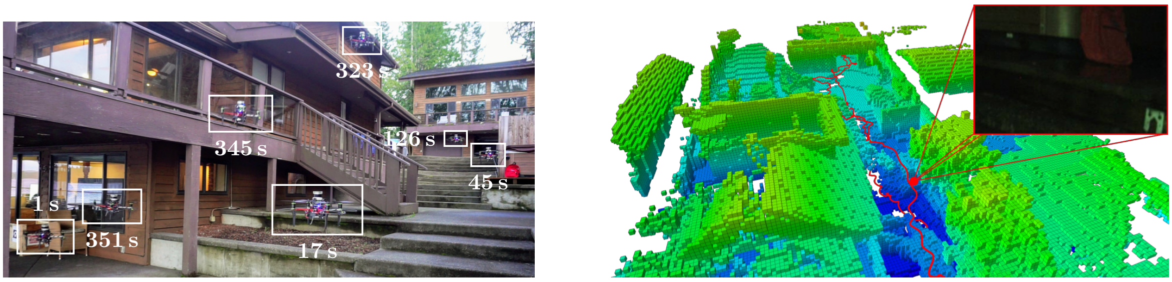
Planning approach	ours	FASTER
computational time - mean (s)	0.95	1.34
computational time - std. (s)	0.35	0.30
UAV-obstacle distance - mean (m)	0.55	0.37
UAV-obstacle distance - std. (m)	0.19	0.21
total number of collisions (-)	3	22
total number of getting stuck (-)	0	13

5.2 Experiments in Testing Facilities

To validate the integration of all the subsystems presented in section 3 during an entire autonomous mission, several experiments were performed within controlled real-world indoor and outdoor environments. During this validation phase, particular emphasis was put on experimental verification of all the subsystems, namely localization and mapping, safe navigation (dynamic replanning and generation of exploration points), localization of artifacts, and the ability to return to a starting location in order to report the obtained data (maps, artifacts detections) back to a ground station. This experimental phase is crucial, particularly in fully-autonomous systems and in time-constrained applications requiring a fast, reliable, and effective response to a hazardous scenarios, such as the targeted scenario of S&R.

A small-scale validation was performed in a constrained outdoor environment shown in Fig. 19. This experimental deployment shows the ability to perform the entire mission with limited time while detecting and localizing an object of interest and returning the object's position back to a ground station. A large-scale validation, shown in Fig. 20, presents three separate testing deployments in an apriori unknown cave environment. The first deployment (highlighted in blue in Fig. 20a) verifies performance over the course of the mission, similar to the experiment in Fig. 19, while additionally showing the seamless transition of the system between highly differing environments (from urban to cave). The second and third flights (highlighted in red and green in Fig. 20a, respectively) show the system performance during deep-cave exploration tasks. In all experiments, the UAVs did not re-use any previously-built maps nor any information about the previous flights. The onboard-built maps were globally registered to a reference map of the cave during an offline post-processing phase. The reference map of the cave was built by registering over 100 largely-overlapped scans taken by Leica BLK360 terrestrial 3D scanner. The characteristics of the three experimental flights are quantified in Table 3. The ground truth data for the quantitative analysis in Table 3 and Fig. 20 were obtained by sequential registration of a local map of each trajectory sample to the precise and highly granular 3D map of the environment obtained with the terrestrial 3D scanner.

Fig. 20b shows the absolute position error of the localization over the course of each particular mission. The data show that the employed SLAM approach yields low drift in environments with high numbers of salient features for extraction within all axes of the structured laser data. Due to the organization of the data in principle of a rotational laser scanner (dense horizontally, sparse vertically), this is not always ensured however. In instances with degraded data in the vertical axis, the information gain during a vertical ego-motion is low, causing the estimated robot pose drift. The occurrence of drift is showcased in Fig. 20c, where the pose drifts due to degradation of vertical information while the UAV flies through a steep narrow corridor. In this scenario, the information gain from the extracted features is low during a large portion of the elevation change, which is the source of position drift at around 300 m in Fig. 20c. However, the possible drift in position is not challenging from the point of mission navigation, which works locally over the map with integrated drift. As the experimental analysis showcases, the accuracy of the SLAM approach is less than the required accuracy for artifacts detection (below 5 m). This makes the approach feasible for short missions (typical of UAVs due to battery constraints), even though it can be improved by employing global optimization at the cost of larger computational load onboard a lightweight UAV.



(a) In-mission snapshots of the identical UAV in time within an experimental environment containing an a priori unknown artifact (red backpack).

(b) Onboard-built volumetric map of the environment with spatial detection of the unknown artifact from an onboard RGB camera and the YOLOv3 neural network (see section 3.6).

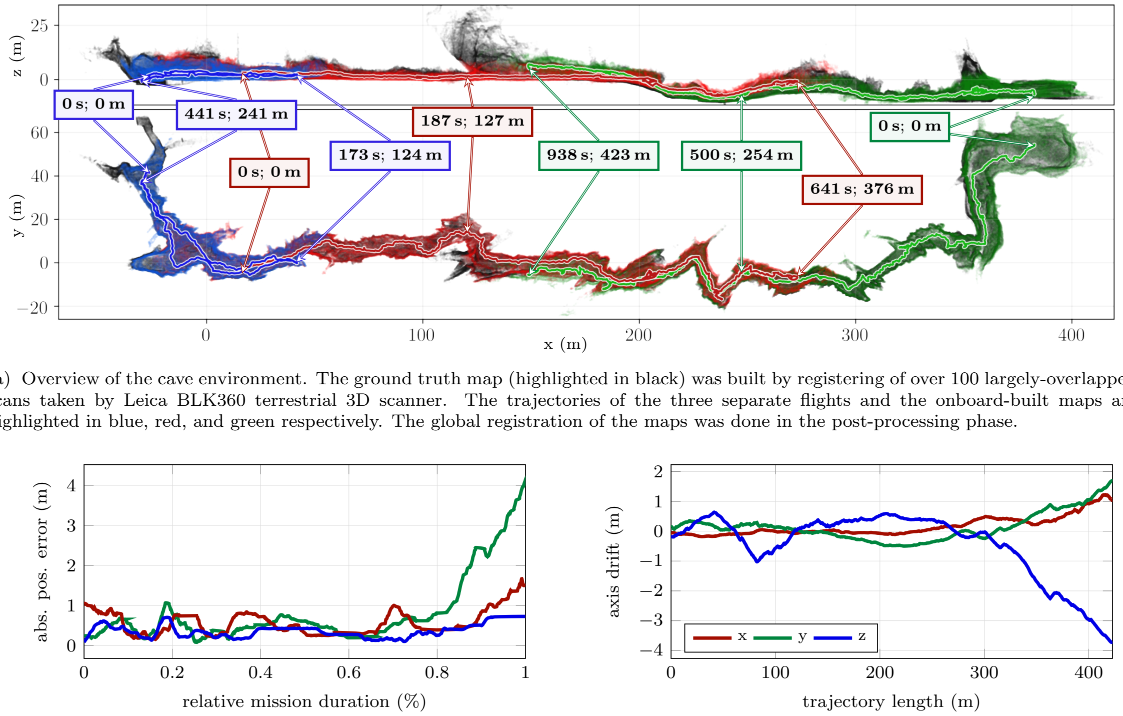
Figure 19: Experimental validation of the entire system. In the experimental outdoor scenario, the system autonomously explored in 361 s a free-space volume of approximately 3665 m^3 (via 3D LIDAR) after a 117.20 m long flight, while it successfully finished the mission by returning to the starting location and landing at the original takeoff spot.

Table 3: Quantitative results of the SLAM performance over the three separate flights shown in Fig. 20.

Trajectory Color in Fig. 20a	Trial 1 Blue	Trial 2 Red	Trial 3 Green
trajectory length (m)	241	376	423
flight time (s)	441	641	938
avg linear velocity (m s^{-1})	0.55	0.59	0.45
explored volume (m^3)	3055	7463	11 403
mean translation error (m)	0.33	0.61	0.93
median translation error (m)	0.29	0.54	0.55
translation RMS error (m)	0.36	0.69	1.36
max translation error (m)	0.73	1.66	4.38

5.3 Experiments in Nuclear Power Plant

The performance of the entire system was objectively examined during the SubT Challenge in an unfinished nuclear power plant in the Satsop Business Park, WA, USA. The competition presented a diverse operation space (confined halls, small rooms, large open spaces) with omnipresent challenges of water pools, layers of dust, hanging ropes, and altering lighting conditions. During four runs on two different courses (Alpha and Beta – see Fig. 1), our system was deployed



(a) Overview of the cave environment. The ground truth map (highlighted in black) was built by registering of over 100 largely-overlapped scans taken by Leica BLK360 terrestrial 3D scanner. The trajectories of the three separate flights and the onboard-built maps are highlighted in blue, red, and green respectively. The global registration of the maps was done in the post-processing phase.

(b) Localization accuracy during all three separate exploration deployments colored with respect to (a). The quantification of mission durations is given in Table 3.

(c) Localization accuracy during a simulated mission highlighted in green: showcase of a position drift in the z-axis due to lack of salient features in structured laser data.

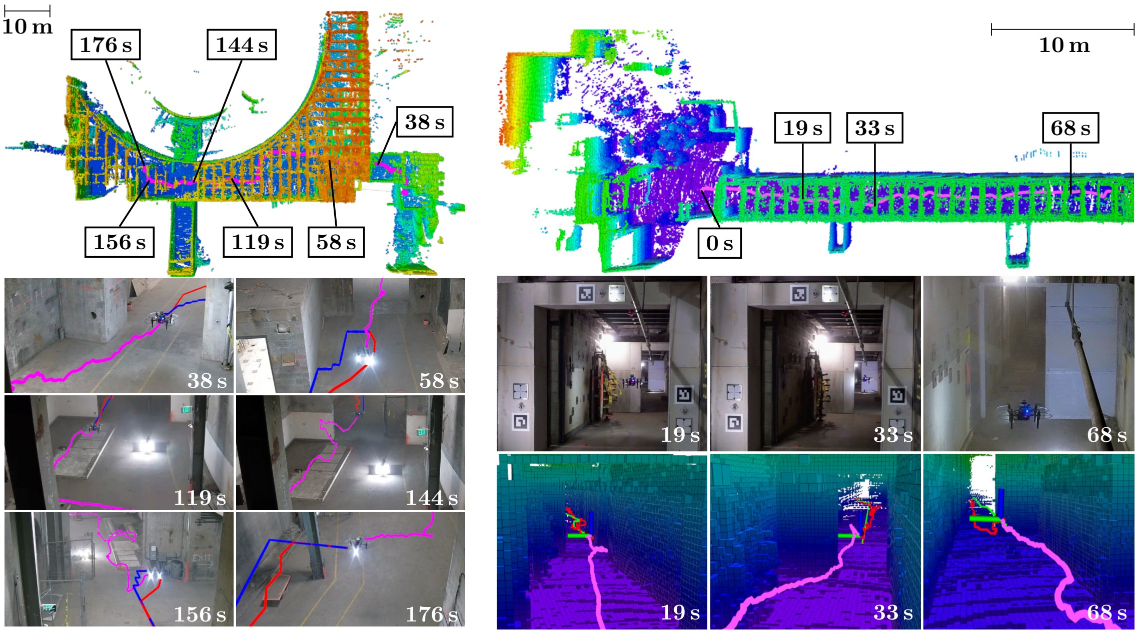
Figure 20: Autonomous exploration of an apriori unknown cave environment by a single UAV in three separate flights. The figure (a) shows the map of the environment, the UAV trajectories, and the onboard-built maps of the cave during the three independent exploration tasks. The localization accuracy of the employed SLAM (see section 3.2) is shown during all the three deployments (b). The figure (c) shows the position drift per each spatial axis appearing in an environment lacking salient features in structured laser data.

as part of a multi-robot rescue team. The robot count of the team reached a maximum of 6 UGV and 2 UAV vehicles deployed at once. As part of the CTU-CRAS-NORLAB team, the proposed UAV system helped to win first place among DARPA non-funded teams and third place overall within the Urban Circuit of the competition.

The Alpha course was an open space scenario reachable through wide hallways and gates. The artificial terrain barriers in this course extended the time required for navigation of ground robots, but this did not affect the proposed aerial system. This showed the superior advantage of the presented UAV system in the fast exploration of open-space environments, as it managed to travel 93 m inside the building in just 200 s. Within this short period of time, a single fully-autonomous UAV explored the space of approximately 2867 m^3 of one floor of the reactor building. On the other hand, it took about half an hour to reach the same distance with the semi-autonomously operated ground robots. The flight trajectory of a single UAV and the volumetric onboard-built map obtained during the flight lasting less than 240 s is shown in Fig. 21a.

The Beta course is characterized by a confined corridor with two 100 cm wide doors at 10 m and at 30 m right after the starting gate. Although the systems of most of the participating teams managed to navigate through these narrow entrances with ground robots (often teleoperated), none of the other teams succeeded in flying through with multi-rotor aerial robots. To the best of our knowledge, only the system proposed here was capable of taking off autonomously with aerial robots at the start location and of reaching the area behind the first extremely narrow gate (see Fig. 21b and Fig. 22). However, since the dimensions of the UAV with propeller guards were only 26 cm less than the width of the door, the performance of the system was tested to its limits, as analyzed in Fig. 14b.

Different deployment strategies were used for the two runs on the Alpha course. In the first run (see Fig. 21a), the exploration policy maximized the amount of explored space and relied on communication with the ground station via a low-bandwidth mesh network. Due to the poor reliability of this network, the strategy changed significantly for the second run. As other members of the multi-robot team relied on the communication network, the first UAV was delegated to



(a) first run on Alpha course (b) second run on Beta course

Figure 21: Single UAV flights within the first run on the Alpha course (a) and within the second run on the Beta course (b). Figure (a) shows the onboard-built map and in-mission snapshots of the UAV with its flight trajectory (pink) and path-planning output – vanilla A* (blue) and its post-processed path (red). Figure (b) shows the onboard-built maps, and in-mission snapshots of the UAV (upper row) navigating through the confined hallway containing narrow passages, which are only 26 cm wider than the footprint of the UAV. The bottom row of figure (b) shows the onboard-built map together with the UAV trajectory (pink) and the path-planning output – vanilla A* (green) and the post-processed path (red).

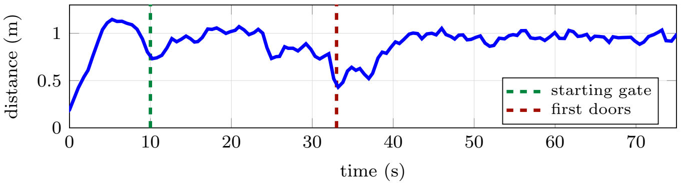


Figure 22: Distance between the center of the UAV and the nearest obstacle throughout the flight during the second run on the Beta course.

land at an advantageous position specified by the mission operator, and served as a data re-translation node for the rest of the team. The first UAV navigated autonomously to a position located approximately 30 m away from the staging area in 72 s after the entire system was booted, it landed, and it re-translated data from all the robots to the ground-station till the end of the mission (1 hour). This strategy significantly increased the range of communication for the ground robots, allowing them to explore further areas of the course. The second UAV followed an exploration policy with the intention of returning to the starting location, transmitting the onboard-built map and detection data to the ground station, and repeating the mission. This robot flew approximately 163 m in 393 s; however, the robot was affected by a hardware failure during the returning phase and its mission was interrupted.

6 CONCLUSION

In this paper, we have presented UAV system for fast autonomous exploration and inspection of large complex unknown indoor environments. The entire system has been developed with a view to combining the required ability to pass through narrow passages, to react properly to the presence of dynamic obstacles, and to withstand real-world conditions. As a part of the presented system, we have introduced a computationally undemanding algorithm for UAV path planning in general 3D cluttered environments, which has a key role in the increase of robustness of the entire system. Further, we have presented a frontier-based exploration strategy that can be applied by a team of robots in fully communication-less environment. The proposed system has been evaluated in a highly thorough experimental analysis of the aerial system in varying environments. The extensive validation was performed by the deployment of the system in single and multi-UAV scenarios, as well as in small and large-scale experiments in simulated and real-world urban and cave environments. In addition to complex validation, the system was evaluated in numerous experiments focused on real-time properties of the system and on thorough evaluation of the ability of the system to fly through confined areas and to withstand challenging real-world conditions including dynamic obstacles. The computational requirements analysis verifies that none of the subsystems show a significant dependence on the size of the explored environment. Thus, the system proves to be applicable independently on the length of the missions. The performance of the entire system was showcased in the DARPA SubT Challenge Urban Circuit where the system was part of the UGV-UAV team that won 1$^{\mathrm{st}}$ place among self-funded teams and 3$^{\mathrm{rd}}$ place overall.

MULTIMEDIA MATERIALS

The multimedia materials are available at `http://mrs.felk.cvut.cz/papers/darpa-urban`. The underlying control pipeline is available as an open-source at `https://github.com/ctu-mrs/`. The planning and exploration part of the system will be also available as an open-source at `https://github.com/ctu-mrs/` after the final round of the DARPA SubT challenge.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENT

This work was supported by CTU grant no SGS20/174/OHK3/3T/13, by OP VVV project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics", by the Czech Science Foundation under research project No. 20-29531S and by the Defense Advanced Research Projects Agency (DARPA).

References

Abouzahir, M., Elouardi, A., Latif, R., Bouaziz, S., and Tajer, A. (2018). Embedding SLAM algorithms: Has it come of age? *Robotics and Autonomous Systems*, 100:14–26.

Achtelik, M. W., Lynen, S., Weiss, S., Chli, M., and Siegwart, R. (2014). Motion- and Uncertainty-aware Path Planning for Micro Aerial Vehicles. *Journal of Field Robotics*, 31(4):676–698.

Alismail, H., Kaess, M., Browning, B., and Lucey, S. (2017). Direct Visual Odometry in Low Light Using Binary Descriptors. *IEEE Robotics and Automation Letters*, 2(2):444–451.

Angermann, M., Frassl, M., and Lichtenstern, M. (2012). Mission review of aerial robotic assessment — Ammunition explosion cyprus 2011. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*.

Báča, T., Heřt, D., Loianno, G., Saska, M., and Kumar, V. (2018). Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Báča, T., Petrlík, M., Vrba, M., Spurný, V., Pěnička, R., Heřt, D., and Saska, M. (2021). The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, 102(26):1–28.

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016). Receding Horizon "Next-Best-View" Planner for 3D Exploration. In *IEEE International Conference on Robotics and Automation*.

Briod, A., Kornatowski, P., Zufferey, J.-C., and Floreano, D. (2014). A Collision-resilient Flying Robot. *Journal of Field Robotics*, 31(4):496–509.

Cadena, C., Carlone, L., Carrillo, H., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332.

Cantelli, L., Mangiameli, M., Melita, C. D., and Muscato, G. (2013). UAV/UGV cooperation for surveying operations in humanitarian demining. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*.

Casagli, N., Frodella, W., Morelli, S., Tofani, V., Ciampalini, A., Intrieri, E., Raspini, F., Rossi, G., Tanteri, L., and Lu, P. J. (2017). Spaceborne, UAV and ground-based remote sensing techniques for landslide mapping, monitoring and early warning. *Geoenvironmental Disasters*, 4:1–23.

Cieslewski, T., Kaufmann, E., and Scaramuzza, D. (2017). Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Cover, H., Choudhury, S., Scherer, S., and Singh, S. (2013). Sparse Tangential Network (SPARTAN): Motion planning for micro aerial vehicles. In *IEEE International Conference on Robotics and Automation*.

Cvišić, I., Ćesić, J., Marković, I., and Petrović, I. (2018). SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics*, 35(4):578–595.

Dang, T., Khattak, S., Mascarich, F., and Alexis, K. (2019). Explore Locally, Plan Globally: A Path Planning Framework for Autonomous Robotic Exploration in Subterranean Environments. In *International Conference on Advanced Robotics*.

Dang, T., Papachristos, C., and Alexis, K. (2018). Autonomous exploration and simultaneous object search using aerial robots. In *IEEE Aerospace Conference*.

Delmerico, J. and Scaramuzza, D. (2018). A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. In *IEEE International Conference on Robotics and Automation*.

Droeschel, D., Nieuwenhuisen, M., Beul, M., Holz, D., Stückler, J., and Behnke, S. (2016). Multilayered Mapping and Navigation for Autonomous Micro Aerial Vehicles. *Journal of Field Robotics*, 33(4):451–475.

Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., Carlone, L., and Agha-mohammadi, A.-a. (2020). LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments. *arXiv e-prints*, page arXiv:2003.01744.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*.

Falanga, D., Kleber, K., Mintchev, S., Floreano, D., and Scaramuzza, D. (2019). The Foldable Drone: A Morphing Quadrotor That Can Squeeze and Fly. *IEEE Robotics and Automation Letters*, 4(2):209–216.

Fang, Z., Yang, S., Jain, S., Dubey, G., Roth, S., Maeta, S., Nuske, S., Zhang, Y., and Scherer, S. (2016). Robust Autonomous Flight in Constrained and Visually Degraded Shipboard Environments. *Journal of Field Robotics*, 34(1):25–52.

Gao, F., Wu, W., Lin, Y., and Shen, S. (2018). Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In *IEEE International Conference on Robotics and Automation*, pages 344–351.

Geneva, P., Eckenhoff, K., Yang, Y., and Huang, G. (2018). LIPS: LiDAR-Inertial 3D Plane SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Harabor, D. and Grastien, A. (2011). Online graph pruning for pathfinding on grid maps. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M. (2014). Autonomous Visual Mapping and Exploration With a Micro Aerial Vehicle. *Journal of Field Robotics*, 31(4):654–675.

Huang, G. (2019). Visual-Inertial Navigation: A Concise Review. In *IEEE International Conference on Robotics and Automation*.

Huang, Y.-W., Lu, C.-L., Chen, K.-L., Ser, P.-S., Huang, J.-T., Shen, Y.-C., Chen, P.-W., Chang, P.-K., Lee, S.-C., and Wang, H.-C. (2019). Duckiefloat: a Collision-Tolerant Resource-Constrained Blimp for Long-Term Autonomy in Subterranean Environments. *arXiv e-prints*, page arXiv:1910.14275.

Jung, S., Song, S., Kim, S., Park, J., Her, J., Roh, K., and Myung, H. (2019). Toward Autonomous Bridge Inspection: A framework and experimental results. In *International Conference on Ubiquitous Robots*.

Kothari, M. and Postlethwaite, I. (2013). A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-Exploring Random Trees. *Journal of Intelligent & Robotic Systems*, 71(2):231–253.

Krátký, V., Petráček, P., Spurný, V., and Saska, M. (2020). Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles. *IEEE Robotics and Automation Letters*, 5(2):2302–2309.

Kruijff, G. M., Pirri, F., Gianni, M., Papadakis, P., Pizzoli, M., Sinha, A., Tretyakov, V., Linder, T., Pianese, E., Corrao, S., Priori, F., Febrini, S., and Angeletti, S. (2012). Rescue robots at earthquake-hit Mirandola, Italy: A field report. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*.

Kruijff-Korbayova, I., Colas, F., Gianni, M., Pirri, F., Greeff, J., Hindriks, K., Neerincx, M., Ogren, P., Svoboda, T., and Worst, R. (2015). TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response. *KI - Künstliche Intelligenz*, 29:193–201.

Lin, Y., Gao, F., Qin, T., Gao, W., Liu, T., Wu, W., Yang, Z., and Shen, S. (2018). Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1):23–51.

Lindner, G., Schraml, K., Mansberger, R., and Hübl, J. (2015). UAV monitoring and documentation of a large landslide. *Applied Geomatics*, 8:1–11.

Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C. J., and Kumar, V. (2017). Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. *IEEE Robotics and Automation Letters*, 2(3):1688–1695.

Loianno, G., Brunner, C., McGrath, G., and Kumar, V. (2017). Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU. *IEEE Robotics and Automation Letters*, 2(2):404–411.

Ma, R., Li, X., Sun, M., and Kuang, Z. (2018). Experiment of Meteorological Disaster Monitoring on Unmanned Aerial Vehicle. In *International Conference on Agro-geoinformatics*.

Mac, T. T., Copot, C., Tran, D. T., and Keyser], R. D. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., and Tadokoro, S. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841.

Miller, I. D., Cladera, F., Cowley, A., Shivakumar, S. S., Lee, E. S., Jarin-Lipschitz, L., Bhat, A., Rodrigues, N., Zhou, A., Cohen, A., Kulkarni, A., Laney, J., Taylor, C. J., and Kumar, V. (2020). Mine Tunnel Exploration Using Multiple Quadrupedal Robots. *IEEE Robotics and Automation Letters*, 5(2):2840–2847.

Mohta, K., Watterson, M., Mulgaonkar, Y., Liu, S., Qu, C., Makineni, A., Saulnier, K., Sun, K., Zhu, A., Delmerico, J., Karydis, K., Atanasov, N., Loianno, G., Scaramuzza, D., Daniilidis, K., Taylor, C., and Kumar, V. (2018). Fast, autonomous flight in gps-denied and cluttered environments. *Journal of Field Robotics*, 35(1):101–120.

Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63.

Nieuwenhuisen, M. and Behnke, S. (2015). 3D planning and trajectory optimization for real-time generation of smooth MAV trajectories. In *European Conference on Mobile Robots*.

Nieuwenhuisen, M., Quenzel, J., Beul, M., Droeschel, D., Houben, S., and Behnke, S. (2017). ChimneySpector: Autonomous MAV-based indoor chimney inspection employing 3D laser localization and textured surface reconstruction. In *International Conference on Unmanned Aircraft Systems*.

Perez-Grau, F. J., Ragel, R., Caballero, F., Viguria, A., and Ollero, A. (2018). An architecture for robust UAV navigation in GPS-denied areas. *Journal of Field Robotics*, 35(1):121–145.

Petráček, P., Krátký, V., and Saska, M. (2020). Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments. *IEEE Robotics and Automation Letters*, 5(2):2078–2085.

Petrlík, M., Báča, T., Heřt, D., Vrba, M., Krajník, T., and Saska, M. (2020). A Robust UAV System for Operations in a Constrained Environment. *IEEE Robotics and Automation Letters*, 5(2):2169–2176.

Qin, H., Meng, Z., Meng, W., Chen, X., Sun, H., Lin, F., and Ang, M. H. (2019). Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments. *IEEE Transactions on Vehicular Technology*, 68(2):1339–1350.

Quan, L., Han, L., Zhou, B., Shen, S., and Gao, F. (2020). Survey of UAV motion planning. *IET Cyber-systems and Robotics*, 2(1):14–21.

Ratliff, N., Zucker, M., Bagnell, J. A., and Srinivasa, S. (2009). Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation*.

Richter, C., Bry, A., and Roy, N. (2016). Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. *Robotics Research*, 114(1):649–666.

Rokhmana, C. A. and Andaru, R. (2016). Utilizing UAV-based mapping in post disaster volcano eruption. In *International Annual Engineering Seminar (InAES)*.

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Pomerleau, F., Kubelka, V., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., and Krajník, T. (2019). DARPA Subterranean Challenge: Multi-robotic Exploration of Underground Environments. In *Modelling and Simulation for Autonomous Systems*.

Saska, M., Krátký, V., Spurný, V., and Báča, T. (2017). Documentation of Dark Areas of Large Historical Buildings by a Formation of Unmanned Aerial Vehicles using Model Predictive Control. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*.

Selin, M., Tiger, M., Duberg, D., Heintz, F., and Jensfelt, P. (2019). Efficient Autonomous Exploration Planning of Large-Scale 3-D Environments. *IEEE Robotics and Automation Letters*, 4(2):1699–1706.

Tordesillas, J., Lopez, B. T., and How, J. P. (2019). FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Usenko, V., von Stumberg, L., Pangercic, A., and Cremers, D. (2017). Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Yang, L., Qi, J., Song, D., Xiao, J., Han, J., and Xia, Y. (2016). Survey of Robot 3D Path Planning Algorithms. *Journal of Control Science and Engineering*, 2016:1–22.

Ye, H., Chen, Y., and Liu, M. (2019). Tightly Coupled 3D Lidar Inertial Odometry and Mapping. In *International Conference on Robotics and Automation*.

Yuan, C., Liu, Z., and Zhang, Y. (2015). UAV-based forest fire detection and tracking using image processing techniques. In *International Conference on Unmanned Aircraft Systems*.

Zhang, J. and Singh, S. (2014). LOAM: Lidar Odometry and Mapping in Real-time. In *Proceedings of Robotics: Science and Systems Conference*.

Zhao, Y., Zheng, Z., and Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowledge-Based Systems*, 158:54–64.

Zhou, B., Gao, F., Wang, L., Liu, C., and Shen, S. (2019). Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters*, 4(4):3529–3536.

Zhou, B., Pan, J., Gao, F., and Shen, S. (2020). Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *arXiv e-prints*, page arXiv:2007.03465.

Zuo, X., Ye, W., Yang, Y., Zheng, R., Vidal-Calleja, T., Huang, G., and Liu, Y. (2020). Multimodal localization: Stereo over LiDAR map. *Journal of Field Robotics*. Online first.

Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles

Pavel Petráček[a], Vít Krátký[a], Matěj Petrlík[a], Tomáš Báča[a], Radim Kratochvíl[b], and Martin Saska[a]

***Abstract*—This paper presents a self-contained system for the robust utilization of aerial robots in the autonomous exploration of cave environments to help human explorers, first responders, and speleologists. The proposed system is generally applicable to an arbitrary exploration task within an unknown and unstructured subterranean environment and interconnects crucial robotic subsystems to provide full autonomy of the robots. Such subsystems primarily include mapping, path and trajectory planning, localization, control, and decision making. Due to the diversity, complexity, and structural uncertainty of natural cave environments, the proposed system allows for the possible use of any arbitrary exploration strategy for a single robot, as well as for a cooperating team. A multi-robot cooperation strategy that maximizes the limited flight time of each aerial robot is proposed for exploration and search & rescue scenarios where the homing of all deployed robots back to an initial location is not required. The entire system is validated in a comprehensive experimental analysis comprising of hours of flight time in a real-world cave environment, as well as by hundreds of hours within a state-of-the-art virtual testbed that was developed for the DARPA Subterranean Challenge robotic competition. Among others, experimental results include multiple real-world exploration flights traveling over 470 m on a single battery in a demanding unknown cave environment.**

***Index Terms*—Aerial Systems: Applications; Field Robots; Aerial Systems: Perception and Autonomy; Multi-Robot Systems; Mapping**

Multimedia Materials

The paper is supported by the multimedia materials available at mrs.felk.cvut.cz/papers/ral-2021-caves. The implementation is also publicly available at github.com/ctu-mrs.

I. Introduction

HUMAN exploration of complex cave systems has occurred for thousands of years. However, there are still entire cave systems and individual subterranean voids, shafts, and cavities that are yet uncovered. This is primarily due to the dangerous nature of subterranean exploration in environments like natural caves, although man-made cellars, drainages, and mines pose similar risks. These environments contain sediments such as debris, rocks, sand, clay, ice, decomposed organic matter, human waste, and even various forms of speleothems in limestone caves. Considering the absolute darkness, lack of GNSS signals, flowing and dripping water, humid air, and the possible presence of poisonous gases, wind gusts, hanging ropes, and wildlife, there is excessive risk to the lives of human explorers in the exploration of new environments, as well as in search & rescue missions. Given the current state-of-the-art technology in robotics, many dangerous areas of subterranean systems are safely reachable using mobile robots, with the greatest focus being on vertical exploration using aerial vehicles. In contrast to human exploration, the use of such technology presents several advantages in the form of accessibility, safety, speed, instantaneous environment visualization, and precise quantification. On the other hand, challenges to the operation of mobile robots in such an environment lies in the uncertainty, lack of light, high humidity, and diversity of space in the form of narrow and/or low passages, canyons, large domes, high chimneys, and deep abysses.

Fig. 1: Robotic exploration of the Bull Rock Cave (central Moravian Karst, Czech Republic) by a fully autonomous aerial vehicle.

The challenges to deployment of aerial vehicles in subterranean environments with respect to robot control, communication, sensor fusion, and positioning are described thoroughly in [1]. These specific challenges continue to be relevant even after substantial progress in the field of mobile robotics. However, in contrast to [1], our motivation includes minimizing the need for communications required for operator control and instead focuses on the full autonomy of robots and autonomous cooperation among members of a robotic team. The restriction of communication in subterranean environments introduces challenges to the maximization of system robustness and the use of efficient decision making in the form of adaptable exploration strategies in harsh unknown environments.

Manuscript received March 1, 2021; Revised May 6, 2021; Accepted July 8, 2021. This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments.

The work was supported by the Czech Science Foundation (GAČR) under research project no. 20-29531S, by CTU grant no. SGS20/174/OHK3/3T/13, by the Defense Advanced Research Projects Agency (DARPA), and by OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics".

[a] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 166 36 Prague 6, Czech Republic {pavel.petracek|vit.kratky|matej.petrlik|tomas.baca|martin.saska}@fel.cvut.cz,

[b] Brno University of Technology, Faculty of Civil Engineering, Institute of Geodesy, Czech Republic r_kratochvil@fce.vutbr.cz.

Digital Object Identifier (DOI): see top of this page.

A. *Related work*

In a non-robotic context, wild caves are explored by modernly termed *cavers*. However, the human surveying and mapping of caves is known to have existed for thousands of years for purposes ranging from dwelling to speleology. The significance of cave exploration and cave mapping to scientific research is a thoroughly studied inquiry in literature, e.g., in [2].

In the work presented here, we focus mainly on the robotic point of view within the scope of the application domain. One of the first cave-mapping approaches using robotic solutions was proposed in [3], where the authors employed hand-held laser scanners, which are limited in speed, accuracy, and safety. In the context of mobile robotics, topics like the automatic control of an unstable dynamic system such as an aerial multirotor vehicle [4], the fusion of inertial, visual, and laser information for localization and mapping [5], and path planning in dynamic environments [6] have been addressed in order to achieve faster and safer methodology than mapping done with hand-held devices, as proposed in [3].

Within the scope of subterranean environments, the DARPA Subterranean Challenge competition has pushed the state of the art of autonomous exploration in human-made mines [7]–[10]. Although these systems have provided interesting solutions with great potential, the authors of [7]–[10] rely on the predictable structure of underground mines, such as using the protraction of human-made tunnels to mark the furthest depth data as frontiers or predefining turns at junctions in [9]. Since the complexity and diversity of natural caves is extensive, more robust solutions with a minimum number of environmental assumptions are required. This was tackled in [11] where the authors introduced a possible way for applying autonomous drones as a technology to assist speleologists and archaeologists. Although an interesting read, the proposed methods only constitute a preliminary discussion that presents neither novel technology nor applied results. A similar discussion focusing on the state of robotic problems within the application of subterranean exploration with UAVs is presented in [12]. In contrast to [11], the authors of [12] present a set of preliminary experiments in laboratory conditions and two dimensional space. Unfortunately, the assumption of a planar world is highly restrictive within the scope of real-world deployment due to the complex character of natural subterranean environments.

The precise localization of mobile robots is crucial to autonomous navigation in such complex environments. Among existing state-of-the-art literature, the LOCUS algorithm [13] achieves the lowest localization error at the cost of high computational demands. Unlike with ground robots, this method might be unsuitable for aerial robots as the computational resources on lightweight UAVs are scarce due to their limited payload. In [14], the authors demonstrated that localization performance can be further improved by dropping range beacons. This is a viable strategy for heterogeneous robotic teams, but unfeasible for teams of only lightweight UAVs.

The use of robotic teams for cooperative exploration has been addressed mostly in planar worlds with recurrent connectivity constraints [15] or with the requirement of a centralized element [16]. A similarly defined task to our problem of team homing — respecting intermittent communication, need for decentralization, and limited operation time of aerial robots — is proposed in [17], where the robots gather and share data during the mission and return all the way back to the base before their operation times out. In contrast to [17], we propose homing coordination that lands each aerial robot at a position expanding a communication relay graph, thereby increasing the time for mere exploration in tasks where return to the starting position is not required. Related to the scope of search & rescue, the authors in [18] propose to re-position robots in a relay-chain formation to enable data transmission over longer distances once an object of interest is found. Our solution reports the position of the objects once the explorer robot connects to the relay graph during homing. The recently developed fast exploration technique in [19] maximizes explored volume over battery-limited flight time. The method is based on data only from an RGBD camera with a limited field of view (FoV). In comparison to LiDAR-based methods, we have experimentally verified that RGBD cameras are sub-optimal sensors for the exploration of large-scale caves due to their limited range and FoV.

B. *Contributions*

First, we propose a fully autonomous system enabling multimodal mapping, fast and efficient planning with sensoric field-of-view constraints for safe movement in 3D, robust localization, and adaptable decision making. Second, a multi-robot cooperation for the efficient homing of a team of autonomous explorer robots is proposed. Third, the system has been validated through hundreds of hours of testing in a state-of-the-art virtual testbed developed for the DARPA Subterranean Challenge robotic competition, as well as through hours of flight time in the real world. To the best of our knowledge, the presented large-scale experimental deployment of autonomous aerial robots in a natural cave environment goes beyond the current state of the art in autonomous robotics. Lastly, we present and share the experience obtained during this comprehensive experimental deployment that was carried out in close cooperation with speleologists.

II. Experience Gained

A. *Speleology motivation*

From the speleological point of view, aerial systems are crucial for pushing exploratory state-of-the-art methods to provide assistance in efficient scouting of difficult-to-access areas in vertical environments, as well as for the quick inspection of known areas using onboard sensors only. These systems minimize risks for humans by reducing the need to climb or to swim in cold water reservoirs, and also through the detection of poisonous gases or even radioactive waste. Furthermore, this enables the preservation and protection of natural environments against human influence, including ancient sediment forms, floor dripstone formations, paleontological and archaeological sites, and sources of potable water.

In contrast to well-established methods of subterranean documentation (i.e., theodolite and level/distance meter, compass, and clinometer), modern technology employs stationary and mobile laser scanners to produce a dense 3D model of the environment. Due to the complexity of natural environments, the use of stationary scanners is time-consuming because of the necessity of eliminating occluded spaces. Although hand-held mobile scanners are more time-efficient in this context, their use is limited to areas accessible to humans. This limitation opens the door for mobile robotics which is able to tackle this challenge and to provide optimized 3D mapping. State-of-the-art mapping in such environments reaches decimeter level precision, which is less precise than stationary scanners, yet sufficient for the majority of speleological needs. Moreover, the common issue of mapping drift accumulation in long-corridor spaces can be minimized using reference measurements by precise stationary scanners or man-measured control points to obtain accurate results.

B. System requirements

The primary prerequisite of a team of aerial explorers that can be deployed in caves involves the ability to adapt to diverse, unknown environments lacking sources of light and access to GNSS. This general description requires the abilities to

- be deployed in constrained cavities, as well as in open caverns of natural caves,
- map and visualize the environment in a fast, quantified manner in the form of dense point clouds and image streams,
- seamlessly infuse an arbitrary exploration strategy for more efficient mission operation within the scope of individual environments (policy selection is discussed in Sec. IV),
- return to the mission operator and promptly visualize the environment for human supervision, and
- maximize operation capabilities in terms of coverage when a team of robots is employed.

C. Depth estimation in high humidity

The performance of the PMD pico flexx time-of-flight (ToF) camera and the Intel Realsense D435 stereo camera have been analyzed as complementary sensors to the primary LiDAR for the purpose of improving the sensory FoV coverage. Although ToF cameras generally outperform stereo cameras in terms of distance measurement precision and density of measurement points [20], the high humidity typically present in natural caves causes dispersion of light emitted from ToF cameras by small water droplets. This effect significantly degrades the acquired measurements. As was verified empirically, ToF cameras can produce false-negative measurements of obstacles situated behind clouds of water droplets. The use of stereo cameras (e.g., Realsense) is recommended for its robustness to environmental conditions within natural caves. Nevertheless for large cave systems, such a sensor needs to be combined with 3D LiDARs in order to comply with the requirements of speleologists and first responders.

III. System Architecture

The system of the proposed autonomous explorer robot is divided into multiple groups of individual interconnected modules to be described in this section. All components and their relations are visualized in Fig. 2.

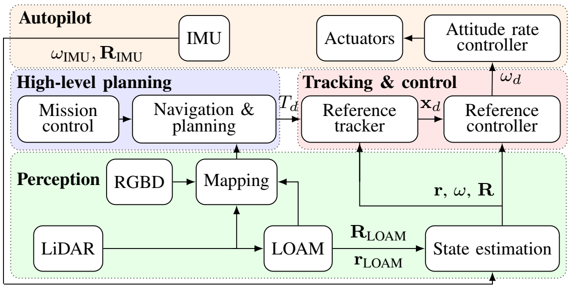


Fig. 2: Individual interconnected modules form the system architecture of the autonomous explorer robot. The *High-level planning* modules focus on achieving the mission objectives by generating references for the *Tracking & control* modules based on the map built by the *Perception* modules. This also provides a state estimate for closing the control feedback loop. All modules except the *Autopilot* group are handled by the main onboard computer.

A. Perception

The perception of the proposed system is based on a multi-channel *LiDAR* sensor that is used for both building the spatial representation of the surrounding environment in the *Mapping* module, as well as for the motion estimation in the *LOAM* module. Obtaining the full-state estimate is realized within the *State estimation* module, where multiple sources of incomplete state measurements are fused together using a bank-of-filters estimator.

The vertical navigation capabilities of the system can be greatly improved by equipping the robot with vertically-facing *RGBD* cameras that are able to fill in the blind spots in the limited vertical FoV of the *LiDAR*. Apart from navigation, these optional sensors may be used for detecting objects of interest in caves in search & rescue scenarios or for visual documentation of newly explored cave systems.

1) LiDAR: Even though our system is not tied to a specific LiDAR model, there are certain important parameters that can affect the performance and capabilities of the platform.

To reliably stabilize the UAV, the time delay of the estimated state must stay below the threshold of a certain critical value depending on the type of controller and gains. When this threshold is exceeded, the UAV begins oscillating and eventually automatically lands when the control error is too large to continue the mission safely. We have found experimentally that for most combinations of localization methods and controllers, the critical value ranges from 100 ms to 200 ms. Thus, 10 Hz is the lowest rotation frequency that can be used without employing methods of delay compensation.

The typical values of a vertical field of view (VFoV) of 3D LiDARs are in the 30° to 90° range. The higher VFoV values improve vertical mobility in constrained spaces, however with a low VFoV, it is impossible to safely navigate narrow vertical shafts as it is not known whether the space above the UAV is free and safe to fly through, or whether it contains an obstacle.

2) RGBD: The regions above and below the UAV that are not covered by the LiDAR can be captured using a depth camera or by spinning the LiDAR sensor around a vector that is orthogonal to the axis of scanning, as seen in [21]. However, such a solution adds additional weight to the sensor, which decreases the available flight time. A blind spot also still remains as part of the laser rays is blocked by the frame of the UAV. Alternatively, lightweight depth cameras can be mounted on opposite sides of the body frame in order to cover most blind spots of the LiDAR. Additional sensing modality is gained by combining an RGB and depth camera in a single sensor (RGBD) with a slight weight increase.

3) Localization: For localization of the UAV, we have adapted the LOAM algorithm [22]. This state-of-the-art method is very precise (0.55 % translation error [23]) while attaining real-time performance. In our adapted version of the open-source implementation, the algorithm is optimized on CPU and employs parallel computing, which enables us to deploy and use the localization in the real-time position control feedback loop onboard fast-moving aerial vehicles.

4) Mapping: The LOAM-based algorithm builds a sparse internal representation of the environment consisting of edge and planar features. However, this sparse map is unsuitable for navigation purposes. Additionally, the LOAM map does not consider the probabilistic nature of the sensor, nor does it distinguish free and unknown space. Both of these factors are necessary in exploration techniques for reliable navigation and consistent frontier selection.

In the proposed system, the environment is represented by a dense probabilistic volumetric map, which consists of cubic cells with one of 3 states: free, occupied, or unknown. The map is kept in the octree structure to facilitate the Bayesian integration of new measurements and efficient access to individual cells of the probabilistic map. The high-level systems, such as grid-based path planning or inter-robot map registration, also benefit from quick access to the dense environment representation. This approach is capable of multi-modal fusion by integrating the data from all available onboard sensors and outputting point-cloud measurements. If high-level path planning is constrained by the field of view of onboard sensors (tackled in [24] and also in Sec. III-C), the multi-modality of mapping enables arbitrary movement in 3D.

5) Sensor processing: The targeted subterranean environments may have high humidity or may contain large clouds of whirling dust. The water and dust particles can then produce erroneous measurements for the LiDAR-based sensors. Assuming a partial reflection from water or dust particles and a large energy dissipation of distant reflections, these erroneous measurements can be filtered with respect to the measured intensity of returning light rays. As has been empirically verified, a simple threshold-based filtration over the intensity channel within the local neighborhood of the sensor is sufficient for filtering out false-positive measurements. The idea of the local filtration is to filter out particles gusting through the surrounding air due to the aerodynamic influence of the propellers. Although the cutoff threshold of the intensity magnitude is environment-specific, filtering out measurements below the 10^{th} percentile of the intensity distribution per each laser scan proved to be a reliable solution, even in the dustiest real-world environments. Such processing is unavailable for camera-based systems that may require thorough, computationally-expensive solutions to overcome these challenges.

6) State estimation: The reference controller (see Sec. III-B2) requires a position estimate of the UAV body frame in the world frame $\mathbf{r} = (x, y, z)$, the velocity of the body frame $\dot{\mathbf{r}}$, rotation $\mathbf{R}$ from the UAV body frame to the world frame, and angular velocity $\boldsymbol{\omega}$ in the body frame in order to close the feedback loop. The LOAM localization method provides 6-DoF pose estimate, i.e., $\mathbf{r}_{\mathrm{LOAM}}$, $\mathbf{R}_{\mathrm{LOAM}}$, which are fused in the *State estimation* block with interoceptive measurements from the *IMU* of the *Autopilot* to obtain the rest of the state variables.

The details about the estimation process are described in [25]. Nevertheless, it is worth highlighting the importance of the fusion of orientation $\mathbf{R}_{\mathrm{IMU}}$ and $\mathbf{R}_{\mathrm{LOAM}}$ in cave environments. While $\mathbf{R}_{\mathrm{IMU}}$ is very precise and without delay, the heading of the UAV (i.e., the measured direction of the body-fixed, forward-facing axis) is unreliable due to the presence of ferromagnetic ores in the cave rocks that cause deviations in the magnetometer measurements. By correcting these errors with the heading from $\mathbf{R}_{\mathrm{LOAM}}$ in the estimation process, the resulting orientation $\mathbf{R}$ is robust to changes in the erratic magnetic field in subterranean environments.

B. Tracking & control

The safe navigation of constrained environments with low obstacle clearance imposes the requirements of precise trajectory tracking with minimal control error, as any deviation from the desired state could potentially result in a collision. The *Reference controller* is responsible for minimizing the control error around the desired control reference that is provided by the *Reference tracker*. The controller outputs an attitude rate reference for the low-level *Attitude rate controller* in the *Autopilot*.

1) Reference tracker: The *Reference tracker* is essential in providing the *Reference controller* with smooth and feasible references to ensure a safe flight. The tracker based on the model predictive control (MPC) simulates an ideal virtual model of the UAV with constrained translational states up to jerk, together with heading and heading rate. The input can be either a single pair of desired 3D position $\mathbf{p}_d$ and heading η_d, or a trajectory T_d in the form of a sequence of such pairs with a specified sampling rate. The full state of the virtual model is then sampled at 100 Hz, and $\mathbf{r}_d$, $\dot{\mathbf{r}}_d$, $\ddot{\mathbf{r}}_d$, $\dddot{\mathbf{r}}_d$, η_d, $\dot{\eta}_d$ are passed to the *Reference controller* as reference $\mathbf{x}_d$.

2) Reference controller: The agile *SE(3)* geometric state feedback controller [26] minimizes the position and velocity errors. To compensate imperfect calibration and external forces acting upon the UAV, the controller is extended with the body and world disturbance terms described in [25]. The output attitude rate reference $\boldsymbol{\omega}_d$ is tracked by the *Autopilot*.

C. Path planning

The planning approach used to safely navigate through apriori unknown environments must fulfill requirements of real-time responsiveness and efficient global planning in order to fully exploit the limited flight time of UAVs. For this purpose, fast iterative post-processing is applied to the output of an optimal grid-based planner in order to increase the UAV-obstacle distance above a minimum threshold [27]. The grid-based planner and the iterative post-processing do not apply an optimistic assumption that the unknown space is collision-free. Although this visibility-constrained precondition requires high sensory coverage around the robot to allow for arbitrary movement in 3D, it consequently prevents collisions of the trajectory being followed, even if replanning would fail. This methodology improves safety and robustness of the overall flight, allows for deployment in completely unknown environments without any apriori information, and permits seamless navigation in open spaces, as well as safe movement through narrow passages.

Common grid-based planning methods require preprocessing of an employed map representation, such as determining and applying the 3D distance transform for obstacle growing. This may introduce significant computational overhead by bottle-necking system performance, as the map must then be processed in every planning step. Such a computationally expensive task contradicts the requirements for responsiveness within evolving dynamic environments. To minimize the overall time required for a single planning iteration, a local KD-tree representation of the environment is used to decide the feasibility of particular cells within a voxel grid. This approach shifts the largest load from the pre-processing phase to the planning phase, which is beneficial especially to shorter plans that require searching only a small part of the environment. The low computational demands of the applied planning approach enable frequent replanning the global plan, which is also crucial for the efficient use of newly-discovered collision-free space.

To effectively exploit the limited flight time of aerial explorers, all mid-flight stops are eliminated by computing in parallel the next exploration goal during path following. The path to the next goal is efficiently appended to the rest of the current reference trajectory T_d using the prediction horizon of the MPC (see Sec. III-B). The need for precise locomotion control in complex natural caves makes uniform path-sampling unfeasible with respect to the dynamic constraints of a UAV and fast, collision-free trajectory tracking. Therefore, the reference trajectory T_d provided by the *Navigation & planning* module to the *Reference tracker* is computed based on the following process.

Given the dynamical constraints of the robot, the generated path is uniformly sampled with a sampling distance adapted to the maximum velocity magnitude v_{max} of the UAV. Based on this initial trajectory T_i, the required acceleration magnitudes a_n between consequent transition points are computed by velocity differentiation as

$$a_n(k) = \frac{||\mathbf{v}_i(k+1) - \mathbf{v}_i(k)||_2}{t_s}, \tag{1}$$

where $\mathbf{v}_i(k)$ is the required velocity vector for transition from a transition point $t_i(k)$ to $t_i(k+1)$ on the initial trajectory T_i and t_s is a constant sampling period. The new velocity for a k-th segment is then given by

$$v_k = \begin{cases} \max\left(v_{max}\frac{a_{max}}{a_n}, v_{min}\right) & \text{if} \quad a_n(k) > a_{max}, \\ v_{max} & \text{if} \quad a_n(k) \leq a_{max}, \end{cases} \tag{2}$$

where the minimum velocity v_{min} serves as a parameter balancing the precision and the time needed for trajectory tracking. By this step, the velocities for particular segments are set so that the maximum velocity is applied in straight segments, while lower velocities are applied in curved segments of any given path.

To further improve trajectory sampling and to achieve smoother changes in velocities, the sampling distance on particular segments is computed so that the motion along each segment has the constant acceleration

$$\overline{a}_k = \frac{|v_{k+1} - v_k|}{t_{acc,k}}, \tag{3}$$

where $t_{acc,k}$ is the time available for acceleration on the k-th segment. The time $t_{acc,k}$ is obtained from the length l_k of the segment k and the required change of the velocity. The number of transition points N_k on the k-th segment of the initial trajectory T_i is given as

$$N_k = \begin{cases} \left\lceil \frac{l_k}{v_k t_s} \right\rceil & \text{if} \quad \overline{a}_k = 0, \\ \left\lceil \frac{t_{acc}}{t_s} \right\rceil & \text{if} \quad \overline{a}_k > 0, \end{cases} \tag{4}$$

where the desired constant acceleration is adapted to meet the velocity v_{k+1} at the end of each segment as

$$a_k = \frac{\overline{a}_k}{N_k t_s}. \tag{5}$$

The sequence of sampling distances for the k-th segment of T_i is then given by

$$d_{k,i} = v_k t_s + i a_k {t_s}^2, \; i \in \{1, \cdots, N_k\}. \tag{6}$$

The trajectory sampled with sampling distances defined by (6) is passed to the *Reference tracker* [25] as a reference trajectory T_d in order to generate a feasible reference $\mathbf{x}_d$ for the *Reference controller*. Despite its simplicity, the described sampling method achieves better results within the scope of the proposed application than the optimization-based trajectory generation methods proposed in [28], [29]. In contrast to the proposed method, the problem in [28], [29] is defined in such a way that the exact positions of all the path waypoints must be visited, generating significantly slower trajectories.

IV. Exploration Policy

Cave environments are naturally diverse and require various different mission strategies suitable for specific environments. Deriving the optimal policy is thereby dependent on various factors, such as the expected mission output, mission-specific constraints, the complexity and the specifics of the environment, and the number of available robots. For this reason, our system is designed so that any arbitrary policy can be utilized within the scope of an autonomous mission.

Nevertheless, two exploratory mission types are of the most use in practice: deep cave exploration and full-coverage exploration. These missions are used for scouting previously uncovered areas in order to obtain a general overview of the environment, monitor environmental changes such as gas leaks, detect natural water reservoirs, discover new possible passages, or assess the structural state of cavern walls and other objects of interest. The former approach maximizes the explored volume of space in the entire environment, while the latter minimizes the blind spots missed by onboard cameras with a constrained FoV.

The capabilities of a robotic mission are furthered with the use of multiple cooperating robots. To show an example of such improvement using a team of agents as opposed to a single agent, a homing strategy that maximizes the flight time of aerial robots during a multi-robotic mission is proposed in the following subsection. During the proposed coordination, continuous exploration is not assumed and distance-constrained ad-hoc communication is used. The robots are homogeneous and generate their behaviors in a decentralized manner based on their current state and the available information from other robots (only positions in a shared frame are required).

A. *Multi-robot homing strategy*

A cooperative operation maximizing the flight time of a multi-robot team is proposed for applications where homing all the deployed robots to an initial location is not required. This strategy is suitable for tasks where the possible gained information is superior to the cost of the robots, such as in search & rescue scenarios. This method assumes there is access to a low-bandwidth communication link among any two robots within an omnidirectional communication radius.

To maximize the flight time, the robots utilize local communication to plan the homing path such that a group of robots is able to build up a communication tree with the base station as the root communication node. This allows the robots to optimize their flight time by navigating back to a location in the proximity of another communication node (a landed robot, base station, or self-sustaining communication node deployed by other robots) when the battery capacity becomes drained. This entire homing strategy is showcased in an example scenario for two independent robots in Fig. 3.

In the proposed strategy, each robot constructs a navigation homing tree using nodes created from the set of past poses of the robot. This online-built tree has edges valued by the required flight time between two nodes and is used to estimate required homing time to the proximity of a communication node. The pose nodes are connected such that each path leaf-to-communication is the shortest (see Fig. 3a). A homing path is constructed recursively as a sequence of tree nodes from the current robot position (a leaf) to the nearest communication node, with the landing position being within communication range of the nearest communication node (see Fig. 3b). The tree is shared among the robots deployed in the same mission. The knowledge from the previous explorers is integrated to prolong their flight time (see Fig. 3c), thus causally maximizing the time capacity for the exploration task. When a communication node (e.g., a robot landing pose) is integrated into the homing tree, it is linked exclusively to another communication node to join the retranslation chain (see Fig. 3d). Consequently, the parents of neighboring pose nodes are updated so that each pose node has a parent with the minimal accumulated cost to any communication node (see Fig. 3b and Fig. 3d). The process of inserting pose nodes as well as communication nodes into the homing tree is described in Alg. 1.

V. Experimental Analysis

The entire proposed system has been validated through hours of flight time in the real world, as well as in hundreds of hours in various virtual subterranean environments. The results of these experimental analyses are presented hereafter.

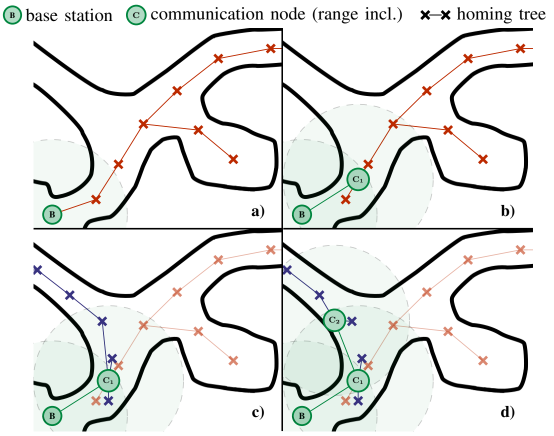


Fig. 3: An example scenario of the homing strategy for two robots (red and blue) that maximizes flight time by landing at feasible positions while building a communication chain to a base station.

Algorithm 1: Insertion of a node into the onboard-built homing tree. Function cost(n_a, n_b) returns an estimate of flight time among nodes n_a and n_b, function accumulatedCost(n_a) returns the required flight time from node n_a to the nearest communication node, and function freeRay(n_a, n_b) returns true if a linear path between nodes n_a and n_b is collision-free in 3D.

```
procedure InsertNodeToHomingTree
    Input:
        N                                  ▷ Node to be inserted
        $\mathcal{C}, \mathcal{P}$          ▷ Sets of communication and pose nodes
        $d_e$                              ▷ Minimum edge length
    if N.$type$ == COMMUNICATION then
        N.$parent \leftarrow \arg\min_{c \in \mathcal{C}}$ cost(N, $c$)
        for $p \in \mathcal{P}$ do          ▷ Update parents of neighboring pose nodes
            if cost(N, $p$) $<$ accumulatedCost($p$) then
                $p.parent \leftarrow$ N
        $\mathcal{C} \leftarrow \mathcal{C} \cup$ N
    else
        $\mathcal{V} \leftarrow \mathcal{C} \cup \mathcal{P}$
        if $\min_{v \in \mathcal{V}} (||\mathrm{N} - v||_2) \geq d_e$ then
            $\overline{\mathcal{V}} = \{v \mid$ freeRay(N, $v$), $\forall v \in \mathcal{V}\}$
            if $\overline{\mathcal{V}} \neq \emptyset$ then
                N.$parent \leftarrow \arg\min_{v \in \overline{\mathcal{V}}}$ [cost(N, $v$) + accumulatedCost($v$)]
                $\mathcal{P} \leftarrow \mathcal{P} \cup$ N
```

A. *Real-world environment*

To analyze the properties of the system, a fully autonomous aerial robot (see Fig. 4) was deployed for several hours of flight time in the Bull Rock Cave located in the central Moravian Karst of the Czech Republic (see Fig. 1 and the attached multimedia materials).

During multiple autonomous exploratory missions, a single explorer (see the hardware components of the robot in Fig. 4) was deployed to validate the proposed system in various exploratory scenarios. The flight trajectories from all missions are visualized in Fig. 5a and the mission statistics and performance metrics of the mapping module are summarized in Table I. A greedy frontier-navigation policy was employed such that the frontier closest to the lateral direction of flight (A, B), the highest frontier (C), and frontier with the largest ratio of unknown to free cells in a bounded area (D) was selected as the next goal. With respect to these experiments in a harsh subterranean environment, we have

- validated the performance of the system by flying in large cave domes, as well as in narrow corridors just 70 cm wider than the dimensions of the robot,

Trial	Flight time (s)	Trajectory length (m)	Explored volume (m^3)	Mapping accuracy μ (m)	σ (m)
A	731	476	7463	0.57	0.59
B	935	473	11 403	0.53	0.56
C	359	71	551	0.23	0.26
D	749	602	3756	0.33	0.38
E	386	233	3055	0.39	0.39
F	633	256	2579	0.21	0.22
G	638	261	3650	0.27	0.33
H	297	142	1682	0.25	0.41
I	129	121	3326	0.19	0.22
J	425	233	4388	0.25	0.29

TABLE I: Quantitative evaluation on multiple autonomous exploratory missions within the Bull Rock Cave system. The flight trajectories and qualitative analysis of the mapping accuracy are shown in Fig. 5.

- validated the real-time performance and robustness of the system in multiple autonomous horizontally-deep flights longer than 470 m using just a single battery and reaching a maximal velocity up to 2 m s^{-1},
- validated the ability to autonomously explore natural domes in terms of vertical depth,
- verified the ability to perform a full mission and return to an initial location with the obtained information,
- quantified the accuracy of the onboard-built maps with respect to a ground truth map of the environment, and
- obtained feedback from speleologists in order to design the system following their requirements.

The dense onboard-built maps (20 cm resolution) from all the experiments were merged (manual global registration with local ICP refinement) during post-processing to obtain the map of the environment $\mathcal{M}$. The reference ground truth map $\mathcal{M}_{gt}$ was built by registering over 100 largely overlapping scans taken by a Leica BLK360 terrestrial 3D scanner. The mapping accuracy over all the experiments reached mean $\mu = 0.37$ m and standard deviation $\sigma = 0.46$ m using the point-to-point Euclidean error metric between each point in $\mathcal{M}$ and the corresponding closest point in $\mathcal{M}_{gt}$. The distribution of the mapping errors throughout all flights is visualized in Fig. 5b. As specified by the end-users, the decimeter-level mapping precision achieved over the course of these exceptionally fast and extensive flights is sufficient for the majority of speleological needs.

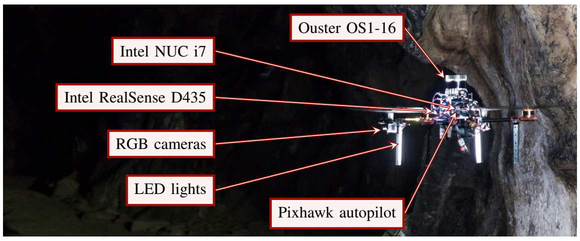
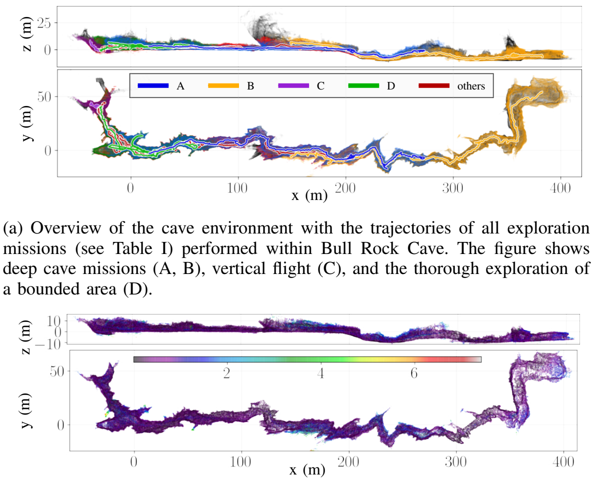


Fig. 4: General hardware components of an autonomous explorer robot. All data are processed and reasoned over with an onboard processing unit. The main source of data comes from the top-mounted LiDAR.

B. Virtual environment

To validate the proposed methodology for multi-robot coordination using a local low-bandwidth communication network, a team of aerial robots was deployed for hundreds of hours of flight in a virtual environment using a virtual testbed developed for the DARPA Subterranean Challenge competition.



(a) Overview of the cave environment with the trajectories of all exploration missions (see Table I) performed within Bull Rock Cave. The figure shows deep cave missions (A, B), vertical flight (C), and the thorough exploration of a bounded area (D).

(b) Visual analysis on the mapping accuracy – the distribution of mapping errors during all autonomous exploration tasks as summarized in Table I. The color bar legend represents the mapping error in meters using the point-to-point euclidean error metric.

Fig. 5: Full-coverage exploration of the Bull Rock Cave system (located in the central Moravian Karst, Czech Republic) with autonomous aerial explorers. Full resolution figure is available within the attached multimedia.

This state-of-the-art testbed consists of several large-scale cave environments containing dynamic obstacles and models of real-world interference, such as sensor discrepancies, communication schemes, and battery longevity.

In contrast to real-world experiments, the virtual environment is larger and allows for the seamless verification of multi-robotic cooperation. To demonstrate the performance of the proposed homing strategy, a selected example scenario of such an operation is presented in Fig. 6. This experiment highlights the positive influence of the homing strategy in a search & rescue scenario where the three explorers were able to exploit the increased flight time. With a 50 m communication range and 1.2 m s^{-1} average velocity for each robot, the homing cooperation increased the available flight time for exploration by 40 s and 80 s, respectively. Moreover, the experiment shows the influence of multi-sensor mapping, which allowed the black robot to single-handedly explore the upper floor of the virtual environment. The final exploratory trajectories of the cooperating robots during the presented mission reached lengths of 715 m, 1349 m, and 1405 m.

The influence of the homing strategy on the time available for mere exploration is also quantitatively analyzed in Table II. The results were averaged over six separate deployments, each with five cooperating robots. Identical mission parameters were set to all the robots for the baseline [17], as well as for the proposed method. The data show an increasing trend in the available mission time for belated explorers for which the effective exploration phase is consequently prolonged during their entire operation time.

VI. Conclusion

This letter presents a comprehensive study on the use of autonomous aerial explorers as an assisting technology for the exploration of natural cave environments. This study also

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2021. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2021.3098304

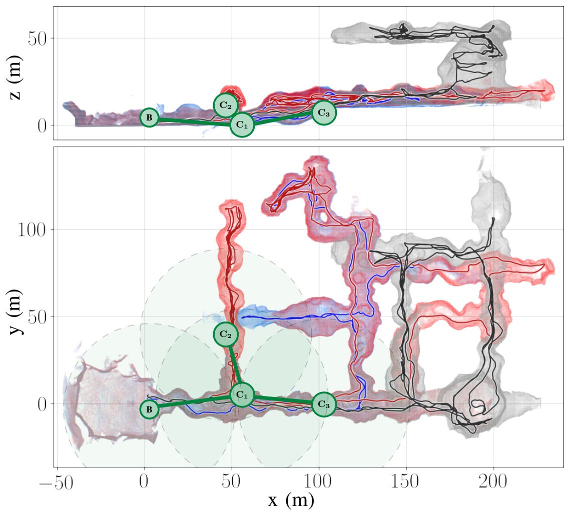


Fig. 6: Three autonomous explorers deployed in a virtual cave world within the DARPA simulation testbed. The robots finished their missions by building a communication tree with maximal edge length of $d_c = 50\,\mathrm{m}$.

Robot	1st	2nd	3rd	4th	5th
Exploration time before homing (secs)	316	330	360	380	389
Exploration time increase (%)	−1.5	2.8	12.2	18.4	21.0

TABLE II: Influence of the homing strategy on the flight time available for mere exploration. Comparison with a baseline time of 321 s (averaged over 10 flights) where a robot returned to base before its operation timed out.

shares the experience acquired during the technology's development in close cooperation with a team of speleologists, cavers, and first responders.

The proposed self-sustaining system interconnects solutions for all crucial robotic tasks in order to enable full autonomy in complex unknown subterranean environments without access to GNSS. Among others, this includes laser-data processing which copes with high humidity and dustiness within subterranean environments and robust path-planning for unknown dynamic environments to allow for flights in constrained cavities, as well as in open caverns of natural caves. Moreover, a multi-robot cooperation is proposed for the efficient homing of a team of robots for applications where the possible information gain is superior to the costs of the robots, such as search & rescue scenarios in cave systems. The performance of the entire applicable system was validated in one of the most large-scale experimental analyses ever conducted, consisting of hours of flight time in Bull Rock Cave (Czech Republic, Moravian Karst) and in hundreds of hours in the state-of-the-art virtual testbed developed for the DARPA Subterranean Challenge. This presented analysis of the entire system proves that it is a robust solution capable of reliable planning with sensoric field-of-view constraints and accurate mapping. The accuracy of localization and mapping was evaluated with respect to a ground-truth map of the cave environment and reached mean precision below 40 cm in real-world conditions. This performance has satisfied the requirements of speleologists and first responders.

References

[1] R. La Scalea *et al.*, "Opportunities for autonomous UAV in harsh environments," in *IEEE ISWCS*, 2019, pp. 227–232.
[2] P. Kambesis, "The importance of cave exploration to scientific research," *Journal of Cave and Karst Studies*, vol. 69, pp. 46–58, 04 2007.
[3] R. Zlot *et al.*, "Three-Dimensional Mobile Mapping of Caves," *Journal of Cave and Karst Studies*, vol. 76, pp. 191–206, 2014.
[4] T. P. Nascimento *et al.*, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Annual Reviews in Control*, vol. 48, pp. 129–146, 2019.
[5] C. Debeunne *et al.*, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.
[6] Y. Zhao *et al.*, "Survey on computational-intelligence-based UAV path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.
[7] K. Ebadi *et al.*, "LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments," in *IEEE ICRA*, 2020, pp. 80–86.
[8] T. Dang *et al.*, "Explore Locally, Plan Globally: A Path Planning Framework for Autonomous Robotic Exploration in Subterranean Environments," in *IEEE ICAR*, 2019, pp. 9–16.
[9] I. D. Miller *et al.*, "Mine Tunnel Exploration Using Multiple Quadrupedal Robots," *IEEE RA-L*, vol. 5, no. 2, pp. 2840–2847, 2020.
[10] M. Petrlík *et al.*, "A Robust UAV System for Operations in a Constrained Environment," *IEEE RA-L*, vol. 5, no. 2, pp. 2169–2176, 2020.
[11] G. Zhang *et al.*, "SmartCaveDrone: 3D cave mapping using UAVs as robotic co-archaeologists," in *IEEE ICUAS*, 2017, pp. 1052–1057.
[12] J. G. R. III *et al.*, "Distributed subterranean exploration and mapping with teams of UAVs," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, vol. 10190, 2017, pp. 285–301.
[13] M. Palieri *et al.*, "LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time," *IEEE RA-L*, vol. 6, no. 2, pp. 421–428, 2020.
[14] N. Funabiki *et al.*, "Range-Aided Pose-Graph-Based SLAM: Applications of Deployable Ranging Beacons for Unknown Environment Exploration," *IEEE RA-L*, vol. 6, no. 1, pp. 48–55, 2020.
[15] Y. Pei *et al.*, "Coordinated Multi-Robot Real-Time Exploration with Connectivity and Bandwidth Awareness," in *IEEE ICRA*, 2010, pp. 5460–5465.
[16] W. Burgard *et al.*, "Coordinated Multi-Robot Exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
[17] K. Cesare *et al.*, "Multi-UAV Exploration with Limited Communication and Battery," in *IEEE ICRA*, 2015, pp. 2230–2235.
[18] J. Scherer *et al.*, "An Autonomous Multi-UAV System for Search and Rescue," in *DroNet*, 2015, p. 33–38.
[19] B. Zhou *et al.*, "FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning," *IEEE RA-L*, vol. 6, no. 2, pp. 779–786, 2021.
[20] F. Pece *et al.*, "Three depth-camera technologies compared," in *First BEAMING Workshop, Barcelona*, vol. 2011. Citeseer, 2011, p. 9.
[21] J. Williams *et al.*, "Online 3D Frontier-Based UGV and UAV Exploration Using Direct Point Cloud Visibility," in *IEEE MFI*, 2020, pp. 263–270.
[22] J. Zhang *et al.*, "Low-drift and real-time lidar odometry and mapping," vol. 41, no. 2, 2017, pp. 401–416.
[23] A. Geiger *et al.*, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE CVPR*, 05 2012, pp. 3354–3361.
[24] M. Beul *et al.*, "Fast Autonomous Flight in Warehouses for Inventory Applications," *IEEE RA-L*, vol. 3, no. 4, pp. 3121–3128, 2018.
[25] T. Baca *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *J Intell Robot Syst*, vol. 102, no. 26, pp. 1–28, 2021.
[26] T. Lee *et al.*, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE CDC*, 2010, pp. 5420–5425.
[27] V. Krátký *et al.*, "An Autonomous Unmanned Aerial Vehicle System for Fast Exploration of Large Complex Indoor Environments," *JFR*, vol. 5, no. 2, pp. 1–24, 2021.
[28] C. Richter *et al.*, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.
[29] M. Burri *et al.*, "Real-Time Visual-Inertial Mapping, Re-localization and Planning Onboard MAVs in Unknown Environments," in *IEEE IROS*, Sept 2015, pp. 1872–1878.

Special Issue: DARPA Subterranean Challenge, Advancement and Lessons Learned from the Finals (DARPA SubT Final)

Field Report

UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT

Matěj Petrlík, Pavel Petráček, Vít Krátký, Tomáš Musil, Yurii Stasinchuk, Matouš Vrba, Tomáš Báča, Daniel Heřt, Martin Pecka, Tomáš Svoboda and Martin Saska
Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

Abstract: This paper presents a novel approach for autonomous cooperating UAVs in search and rescue operations in subterranean domains with complex topology. The proposed system was ranked second in the Virtual Track of the DARPA SubT Finals as part of the team CTU-CRAS-NORLAB. In contrast to the winning solution that was developed specifically for the Virtual Track, the proposed solution also proved to be a robust system for deployment onboard physical UAVs flying in the extremely harsh and confined environment of the real-world competition. The proposed approach enables fully autonomous and decentralized deployment of a UAV team with seamless simulation-to-world transfer, and proves its advantage over less mobile UGV teams in the flyable space of diverse environments. The main contributions of the paper are present in the mapping and navigation pipelines. The mapping approach employs novel map representations—SphereMap for efficient risk-aware long-distance planning, FacetMap for surface coverage, and the compressed topological-volumetric LTVMap for allowing multirobot cooperation under low-bandwidth communication. These representations are used in navigation together with novel methods for visibility-constrained informed search in a general 3D environment with no assumptions about the environment structure, while balancing deep exploration with sensor-coverage exploitation. The proposed solution also includes a visual-perception pipeline for on-board detection and localization of objects of interest in four RGB stream at 5 Hz each without a dedicated GPU. Apart from participation in the DARPA SubT, the performance of the UAV system is supported by extensive experimental verification in diverse environments with both qualitative and quantitative evaluation.

Keywords: Unmanned Aerial Vehicles, Search and Rescue, DARPA, SubT, autonomy, exploration, navigation, deployment, subterranean environment, degraded sensing

Support materials

The paper is supported by the multimedia materials available at mrs.felk.cvut.cz/fr2022darpa. Open-source implementation of the core of the UAV system is available at github.com/ctu-mrs/

Received: 6 June 2022; revised: 7 October 2022; accepted: 29 November 2022; published: 2 January 2023.

Correspondence: Matěj Petrlík, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic, Email: matej.petrlik@fel.cvut.cz

DOI: https://doi.org/10.55417/fr.2023001

mrs_uav_system. The SLAM datasets are available at github.com/ctu-mrs/slam_datasets. The visual detection datasets are available at github.com/ctu-mrs/vision_datasets.

1. Introduction

The research of new robotic technologies and solutions is accelerating at an unprecedented rate mainly in case of aerial robotics. Technological development is improving many areas of our lives and, hopefully, even the future of humanity. The authors of (Shakhatreh et al., 2019) reviewed current research trends and future insights on potential Unmanned Aerial Vehicle (UAV) use for reducing risks and costs in civil infrastructure. The survey of UAV applications is accompanied by a discussion of arising research challenges and possible ways to approach them.

This paper focuses on a robotic system developed to autonomously search subterranean environments. The motivation behind searching subterranean environments is to gain situational awareness and assist specialized personnel in specific missions. Such missions may include: assessing the structural integrity of collapsed buildings, tunnels, or mines; exploration of a newly discovered branch in a cave network; or searching for lost persons. These tasks can often be life-threatening to human workers as many hazards are present in subterranean environments. In order to reach survivors quickly in unstable caves or partially collapsed burning buildings, first responders, such as emergency rescuers and firefighters, may potentially put their lives at risk. In firefighting tasks, fires can be either localized and reported to personnel by robots or the robots can even directly begin extinguishing flames if the presence of human firefighters is too risky (Spurny et al., 2021; Pritzl et al., 2021; Martinez-Rozas et al., 2022). In such scenarios, ceilings can suddenly collapse, toxic gas can appear in a mine, flames can extend to an escape corridor, or a cave cavity can flood with water. In distress situations, it is essential to swiftly coordinate the rescue operation as the survivors of a catastrophe might need acute medical assistance or have a limited amount of resources available, namely oxygen and water. However, without conducting a proper reconnaissance of the environment and assessing the potential risks prior to the rescue mission, the involved rescuers are exposed to a much higher probability of injury.

To reduce the possibility of bodily harm or to avoid risks altogether, a robotic system can be sent on-site before the rescuers in order to either quickly scout the environment and report any hazards detected by the onboard sensors, or directly search for the survivors. The rescue mission can be further sped up by deploying a team of robots capable of covering larger areas and offer redundancy in case of losses of some robot units in harsh environments. Multirobot teams can also consist of heterogeneous agents with unique locomotion modalities to ensure traversability of various terrains, including muddy ground, stairs, and windows, which is discussed in the overview of collaborative Search and Rescue (S&R) systems (Queralta et al., 2020). Similarly, sensing modalities can be distributed among individual robots to detect various signs of hazards, such as increased methane levels or the potential presence of survivors deduced from visual or audio cues. Mounting all sensors on a single platform would negatively affect its dimensions and, consequently, its terrain traversability as it may not be able to fit into narrow passages, such as crawlspace-sized tunnels or doorways. It would also mean a single point of failure for the rescue operation. On the other hand, the operation of a single robot can be managed by just one person, while commanding a robot team may be unfeasible for a single operator. Assigning each robot to an individual operator would also be an ineffective allocation of resources. Moreover, the range of the robot would be limited by the communication link to the operator. To provide a valuable tool for the rescue team, the robots must be able to move through the environment on their own and infer about the environment using their sensor data. The rescuer can then also act as an operator, providing only high-level inputs to the robotic system to bias their behavior based on *a priori* information (e.g., someone was last seen on the east side of the third floor). The research and development of such autonomous systems for assisting first responders is the primary focus of the S&R robotics, and also the motivation for the S&R UAV system presented in this paper.

The robotic platforms typically considered for S&R tasks are categorized into wheeled, tracked, legged, marine, and aerial platforms (Delmerico et al., 2019). Among these locomotive modalities, aerial robots are considered to have the highest traversal capabilities since they can fly over most obstacles which are untraversable by other platforms. One example of an autonomous aerial research platform for S&R is found in (Tomic et al., 2012). The mobility of UAVs also surpasses other robot types thanks to its dynamic flight which can achieve large velocities and accelerations. These qualities make UAVs ideal for swift environmental scouting for gaining initial knowledge about a situation. As such, the aerial platform is predetermined to be deployed as the first robot during the first minutes of the rescue operation. A team deployed in an outdoor multi-UAV disaster response task (Alotaibi et al., 2019) can effectively cover a large search area and minimize the time to find and reach survivors. On the other hand, UAVs cannot operate for extended periods of time due to their limited flight time, and the sensory equipment is limited by the maximum payload of the UAV. Some sensing modalities might even be unsuitable for the use on aerial robots due to their propulsion system, e.g., detecting gas due to the aerodynamic effects of the propellers, or sound detection due to noisy operation. Due to the aforementioned pros and cons of UAV platforms, it is convenient to combine the capabilities of other robot types to form a heterogeneous robotic team.

This manuscript proposes an autonomous cooperative UAV approach for S&R. The approach used by Unmanned Ground Vehicles (UGVs) is not presented here because it is vastly different from the UAV system and as such would not fit into the scope of this article, which is already moderately extensive as we did not want to omit any details about the deployed system. The UGV solution was developed by our colleagues who are acknowledged at the end of this article. The proposed UAV together with legged, wheeled, and tracked UGVs formed the CTU-CRAS-NORLAB team, which participated in the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge (SubT). The team consisted of Czech Technical University in Prague (CTU) and Laval University.

1.1. DARPA SubT challenge

After major success in accelerating the development of self-driving cars in the Grand Challenges of 2004 and 2005 and the Urban Challenge in 2007, DARPA announced the Subterranean Challenge (SubT) (Orekhov and Chung, 2022) for the years 2017-2021 to advance the state of the art of S&R robotics. Participants had to develop robotic solutions for searching subterranean environments for specific objects that would yield points if reported with sufficient accuracy. To achieve the task at hand, the competitors had to develop complex multirobot systems spanning nearly all research areas of mobile robotics, from design of the robotic platforms to high-level mission planning and decision-making.

The rules of the competition can be summarized in a few points. Each team has a dedicated time slot, or *run*, to send their robots into a previously unvisited course and search for specific objects, referred to as artifacts (Figure 1). Each run starts at a predefined time and ends exactly one hour later. A single team is present on the course at a time during which they can deploy an unconstrained number of robots of arbitrary size. The movement of team personnel and their handling of robots is allowed only in the area in front of the entrance to the course, as shown in Figure 2. Only robots can enter the course and only one human operator/supervisor can command the robots and access

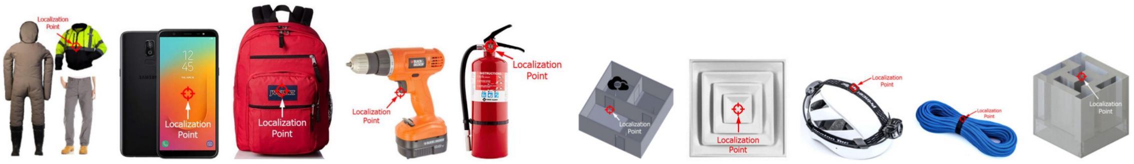


Figure 1. All 10 artifacts searched for in the Final Event of DARPA SubT (image courtesy of DARPA). The operator had to submit the position of the identified artifact with accuracy better than 5 m. While the first three artifacts (survivor, cellphone, and backpack) were present in all circuits, the drill and the fire extinguisher were tunnel-specific. Similarly, the gas and vent were located in the urban environment, and the helmet with rope could be found in the caves. The last artifact (the cube) was introduced only for the Final Event.

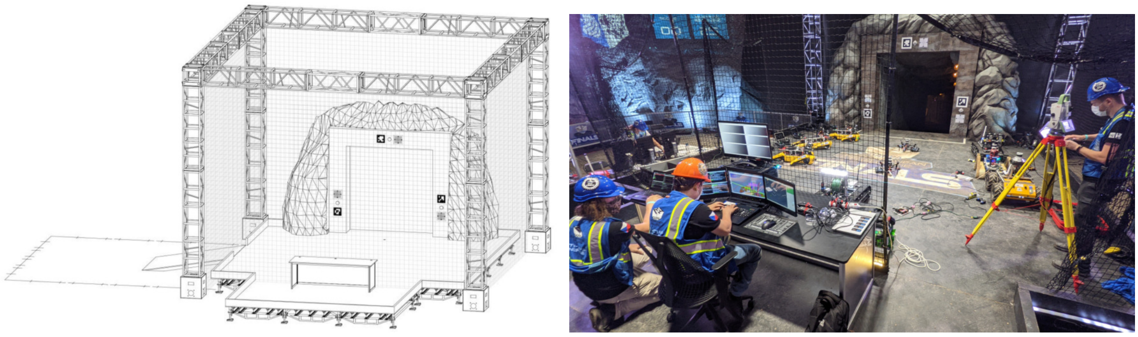


Figure 2. The bounded staging area (image courtesy of DARPA) is the only place where the human crew members can handle the robots. The person sitting behind the displays is the operator who is the only one allowed to issue commands to the team of robots, and also to view and interpret mission data.

Table 1. The prize money awarded for achieving the first three places in the Final Event.

Place	Systems Track	Virtual Track
1.	$2M	$750K
2.	$1M	$500K
3.	$500K	$250K

the data they acquire during the run. These conditions should mimic the conditions of a real S&R robotic mission. The operator can report the type and position of an artifact. If the type was correct and the reported position was not further than 5 m from the true position, the team was awarded one point. The team with the highest score wins the prize according to Table 1. For a more detailed description of the challenge, see (Orekhov and Chung, 2022).

To encourage the development of high-level components without worrying about the resilience of the hardware in harsh subterranean conditions and also to enable teams without many resources and/or physical robots to compete, a virtual version (Virtual Track) of the competition was run in parallel to the physical Systems Track. The solutions of the Virtual Track were uploaded as Docker images (one image per robot) to the Gazebo-based Cloudsim simulation environment, where the entire run was simulated. Every team could use the Cloudsim simulator to test their approaches in practice worlds prior to the actual competition.

The competition was further subdivided into individual circuits, which were events in the specific subterranean environments of a tunnel, cave, and urban space. Examples of each environment are shown in Figure 3. The surroundings were chosen to correlate with typical real S&R sites to assure the applicability of the systems developed during the competition. Every type of environment differs in size, geometric dimensions, traversability conditions, and requirements on perception modalities. The specifics of tunnel-like environments are summarized in (Tardioli et al., 2019) with 10 years of experience in S&R ground robots research. The role of mobile robots in rescue missions after mine disasters is discussed in (Murphy et al., 2009). The Final Event combined all of the previous environments for the ultimate challenge.

We participated in the competition first as a nonsponsored team. In the Tunnel Circuit, we won 1st place among the nonsponsored teams and 3rd place in total, which earned us $200 000. The aerial robots explored 340 m of tunnels and found 3 artifacts out of the 10 artifacts discovered by all our robots (Petrlik et al., 2020). This success was repeated in the Urban Circuit with the same place achieved but this with time larger prize money $500 000. The UAVs proved their suitability for quick scouting of the environment thanks to their advantage in mobility when the first deployed UAV managed to travel 93 m inside the building in just 200 s while it took about half an hour to reach the same area with the semi autonomously operated ground robots as reported in (Kratky et al., 2021a). One of the deployed UAVs also served as a retranslating station for other robots

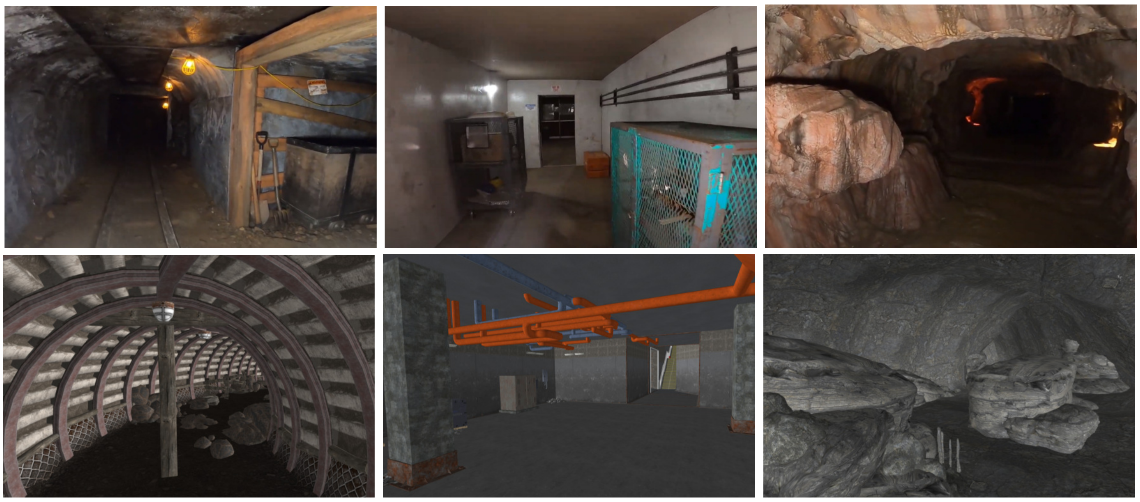


Figure 3. Three types of subterranean environments found in the competition, each challenging for the robot team in a different way. From left to right: tunnel, urban, and cave. The top row shows examples of environments from the Systems Track of the Final Event, while the virtual worlds are pictured in the bottom row.

after navigating to and landing at a strategic position. Thanks to consistent performance in both circuits, DARPA awarded our team the funding for the Final Event, which allowed us to acquire more capable hardware. In the Virtual Track, the UAVs were used as the primary platform for finding artifacts thanks to their high travel speed and the ability to fly over terrain untraversable by UGVs. The ground robots supported longer flights of the UAVs by extending the communication network with breadcrumbs. In total, 215 artifacts were found by the UAVs in the competition worlds (8 artifacts less than the winner). The performance of UAVs in the confined environment of the Systems Track was worse than in the Virtual Track. Nevertheless, while the UGVs detected 5 out of 7 scored artifacts, the aerial robots managed to add 2 unique artifacts not seen by other robots but 1 of them had the wrong class and image and thus could not score. The last point was scored manually by the operator by matching a detection from a UGV that had inconsistent map with the same position in a correct map. The approach presented in this paper is the result of UAV research, development, and testing over the whole 3-year-long period.

2. Related work

The state of the art in rescue robotics is coherently summarized in the survey (Delmerico et al., 2019), which concerns both hardware and software. On the hardware side, different robot morphologies, locomotion types, and platform designs are categorized. Regarding software, the survey concerns perception and control algorithms. The authors interviewed experts on disaster response and humanitarian aid to understand the situation and needs of rescuers.

Here, we provide an overview of the solutions for perception in adverse conditions of the underground environments, methods of localization and mapping for precise and reliable navigation, and techniques for safe traversal of narrow corridors. A summary of systems deployed in previous circuits of DARPA SubT follows. Finally, relevant datasets are referenced in order to prompt further research effort in the S&R area.

2.1. Degraded sensing

Perception in subterranean environments faces constant degradation of the sensor outputs due to the harsh conditions of such places. The underground climate is often filled with impervious dust (particularly in mines), where any movement agitates the settled layer of fine dirt and mineral particles. On the other hand, caves are typically humid ecosystems, where dense mud replaces the dust layer found in mines. However, the elevated humidity forms droplets of fog, which corrupt the measurements of most visible or Near Infrared (NIR) light-based sensor modalities, and also causes frequent reflections on wet surfaces. Radars can reliably penetrate smoke, dust, and fog, and after postprocessing using, e.g., Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), a 2D occupancy grid for navigation (Lu et al., 2020) can be constructed. Another reliable sensing modality for when images from color (RGB) cameras are polluted by dust or fog is thermal imaging, which, in (Khattak et al., 2019), is used for the localization of robots in areas with airborne obscurants. Our approach goes beyond these works by employing intensity-based filtering of the Light Detection and Ranging (LiDAR) data, and thus no additional sensors are necessary even in dense clouds of dust.

2.2. Localization and mapping

Recent developments in S&R robotics sparked the research of more precise local pose estimation algorithms (also referred to as odometry), as well as long-term globally consistent trajectory and multirobot map fusion of all agents of the robotic team. The state-of-the-art methods were published in (Cadena et al., 2016), where the challenges and future direction of the Simultaneous Localization and Mapping (SLAM) development are also identified. The demands on low control error and robustness to degraded sensor data in the narrow subterranean environments present in the DARPA SubT pushed all contesting teams to either adapt and improve an existing method to be usable in the extreme conditions, or to develop a new SLAM tailored to this specific domain. SLAM methods used by the teams in the Final Event are summarized in (Ebadi et al., 2022) along with expert opinions about the present maturity and future outlook of the field.

Team CoSTAR developed a LiDAR odometry solution (Palieri et al., 2020) based on Generalized Iterative Closest Point (GICP) matching of LiDAR scans with initialization from Inertial Measurement Unit (IMU) and wheel odometry, including the possibility of extension to other odometry sources, such as Visual-Inertial Odometry (VIO). The method is shown to outperform state-of-the-art localization methods on the datasets from Tunnel and Urban circuits. An ablation study presents the influence of individual components on the total Absolute Position Error (APE). The second improved version (Reinke et al., 2022), which was released as open-source is less computationally demanding, less memory intensive, and more robust to sensor failures. All presented experiments are conducted with ground robots. The localization of aerial vehicles is handled by a resilient HeRo state estimation system (Santamaria-Navarro et al., 2019). The state estimation stack considers heterogeneity and redundancy in both sensing and state estimation algorithms in order to ensure safe operation, even under the failure of some modules. Failures are detected by performing confidence tests on both data and algorithm health. If a check does not pass successfully, the resiliency logic switches to the algorithm with the best confidence, similar to our previous solution published in (Baca et al., 2021). The local odometry of (Palieri et al., 2020; Santamaria-Navarro et al., 2019) is accompanied by loop closure detection and pose graph optimization locally on each robot, as well as globally on the base station. This optimizes the trajectories of all robots for a multirobot centralized SLAM solution (Ebadi et al., 2020). After improving the performance of the multirobot loop closure generation and pose estimation, especially in large-scale underground environments, the solution was open-sourced (Chang et al., 2022) and released together with a multirobot dataset from the subterranean environment. A technique for loop closure prioritization (Denniston et al., 2022) improves the Absolute Trajectory Error (ATE) of the multirobot SLAM by prioritizing loop closures based on observability, graph information, and Received Signal Strength Indicator (RSSI) criteria. A decentralized SLAM solution for UAVs (Lajoie et al., 2020) performs distributed outlier-resilient pose graph optimization when another agent is within communication range. This method can be used with either a stereo camera or a LiDAR, and is evaluated on a dataset from the Tunnel Circuit.

The long, featureless corridors that are often present in man-made tunnels lead to unobservability of the motion along the degenerate direction, which leads to significant drift. Promising approaches, such as (Shan et al., 2020; Xu et al., 2022), constrain the solution of the optimization problem using the preintegrated IMU measurements. This helps to reduce the localization drift under unfavorable environmental geometry. Nevertheless, the vibrations induced by spinning propellers degrade the inertial measurements, and can thus negatively affect the localization precision. Approaches, such as those seen in (Ebadi et al., 2021), detect the geometrical degeneracy using the ratio of the most observable and the least observable directions. This ratio is then used to determine loop closure candidates to reduce the drift along the degenerate direction. Similarly, (Zhang et al., 2016) handles environment degeneracy in state estimation by not updating the solution in detected degenerate directions. Another possibility is to combine the 3D LiDAR method with a direct visual odometry method [e.g., (Alismail et al., 2016)], which tracks image patches by minimizing the photometric error. This approach, which is shown in (Shin et al., 2020), has the advantage over feature-based methods like that of (Zhang and Singh, 2015) in that it provides low drift, even when salient image and geometric features are lacking. The disadvantage is that localization performance is worsened when whirling dust is present in the camera image, as reported in (Petrlik et al., 2020).

Team CERBERUS developed a complementary multimodal sensor fusion (Khattak et al., 2020). The odometry estimated by visual/thermal inertial odometry is used as a prior for LiDAR scan-to-scan and scan-to-map matching. The VIO/TIO priors constrain the scan matching optimization problem, thus reducing drift in a degenerate environment significantly, which is demonstrated in an experiment conducted in a self-similar environment.

Another multimodal approach is the Super Odometry (Zhao et al., 2021) of team Explorer, which was deployed on aerial robots in the tunnel and urban circuits of DARPA SubT. The core of the method is the IMU odometry with biases constrained by VIO and LiDAR-Inertial Odometry (LIO), which are initialized with preintegrated inertial measurements of the constrained IMU. The relative pose factors of VIO and LIO are weighted based on the visual and geometrical degradation, respectively.

Team MARBLE first relied on visual SLAM (Kramer et al., 2021), but after Subterranean Integration Exercise (STIX), they transitioned to the LiDAR-based Cartographer (Hess et al., 2016) due to unstable tracking of motion under poor illumination, reflections, dust, and other visual degradation.

Wildcat SLAM (Hudson et al., 2022) of the CSIRO Data61 team is a multiagent decentralized solution, where each agent computes a global map using the currently available data shared among the robots. The odometry of each agent is based on the work of (Bosse et al., 2012).

Our approach is similar to the other teams' as we also use primarily LiDAR for localization and mapping. An improvement over the state of the art is the compensation of the delay (Pritzl et al., 2022a) caused by the LiDAR scan processing and the delay of the localization itself.

2.3. Mobility

Deploying aerial robots has one great advantage over ground robots due to their full terrain traversability. A UAV can fly over terrain that would compromise the safety of an UGV, e.g., steep decline, mud, water, etc. This allows to neglect the traversability problem necessarily tackled in solutions to UGV navigation (Fan et al., 2021), as the only movement constraint of aerial platforms flying through an enclosed environment is the minimum size of a passage that the robot can safely pass through. The dimensions of such passages depend largely on the size of the UAV, but also on the precision of the pose estimation, the control error of onboard regulators, the map quality, and the reactive behavior in close vicinity of obstacles. Some platforms also tolerate contact with obstacles in the sense that the contact does not endanger the continuation of the mission (Huang et al., 2019). Other types of platforms adapt their morphology and/or locomotion modality to their current surroundings and obstacles (Fabris et al., 2021). In voxel-based map representations, the size of a narrow passage is represented too conservatively, i.e., the size of the narrow passage in the voxel map is the lower bound of the true size. However, in practice, the narrow passage can be up to twice the map resolution larger than its voxel representation, which prevents traversing passages that are well within the physical limits of the UAV. To better approximate the true shape of the narrow passage, (O'Meadhra et al., 2018) propose continuous representation based on Gaussian Mixture Models (GMM) (Reynolds, 2009), which is converted to a voxel map of arbitrary resolution when queried. However, the information about the exact structure of the environment is lost due to the approximation by Gaussian distribution. We took another approach of locally increasing the resolution of the occupancy voxel map when the size of the environment requires it, which preserves all details.

To fully exploit the capabilities of UAV's full terrain traversability, the path planning and trajectory generation algorithms have to work with a full 3D representation of the environment and fulfil the real-time requirements. Although several excellent works on planning in constrained environments were recently published (Zhou et al., 2021a; Tordesillas et al., 2022), they focus primarily on generating fast trajectories while the presented application requires maximizing the reliability of the system in the presence of uncertainties imposed by a harsh dynamic environment. In addition, deploying the planning algorithm as part of the complex system running on board UAV with limited computational resources motivates the use of computationally undemanding algorithms. Similarly to team Explorer (Scherer et al., 2022), we make use of a multistage approach consisting of extensively validated computationally undemanding algorithms well-integrated into presented system (Baca et al., 2021; Kratky et al., 2021a).

2.4. DARPA SubT approaches

This paper primarily focuses on the approach developed for and experimentally verified in the Final Event of DARPA SubT. As mentioned, these results are built upon the experience in using the approaches developed for the tunnel and urban circuits. The practical verification of the developed solutions in challenging environments justifies the robustness of these algorithms. Valuable insights on the future of S&R robotics can be drawn from lessons learned by the teams.

Team CoSTAR relied on their uncertainty-aware framework, NeBula, in the tunnel and urban circuits (Agha et al., 2021). The framework supports multimodal perception and localization including radar, sonar, and thermal cameras. Aerial robots were part of their heterogeneous team in STIX and the tunnel circuit, mainly for exploring areas inaccessible to ground robots and data muling with distributed data sharing (Ginting et al., 2021; Saboia et al., 2022). A reactive autonomy approach COMPRA (Lindqvist et al., 2021) was also proposed for UAV underground S&R missions. Their solution gained 2nd and 1st place in the tunnel and urban circuits respectively.

Team Explorer developed a system (Scherer et al., 2022) that achieved 1st place in the tunnel circuit and 2nd place in the urban circuit. Their collision-tolerant platform "DS" with flight time of 13 min was carried on top of a UGV and could be launched by the operator when needed. The authors identified the challenge of combined exploration and coverage problem when their UAVs with limited camera Field Of View (FOV) missed some artifacts along their flight path. The frontier-based exploration pipeline used a custom OpenVDB mapping structure (Museth, 2013) for sampling frontier-clearing viewpoints. Paths to found viewpoints were planned using bidirectional RRT-Connect.

Team CERBERUS deployed legged ANYMAL robots and aerial DJI Matrice M100 robots in the tunnel circuit. Their graph-based system for the autonomous exploration of subterranean environments called GBPlanner was deployed in multiple locations. The exploration of Edgar mine during STIX and the National Institute for Occupational Safety & Health (NIOSH) mine during the tunnel circuit are documented in (Dang et al., 2020b). Specifically, the exploration method for aerial robots (Dang et al., 2019a) consists of a local fast-response layer for planning short collision-free paths and a global layer that steers the exploration towards unvisited parts of the map. This method is part of the solution for underground search by aerial robots found in (Dang et al., 2020a). A mapping and navigation approach (Papachristos et al., 2019a) for autonomous aerial robots based on the next-best-view planner (Papachristos et al., 2017; Bircher et al., 2016) was also proposed, but was later outperformed by the GBPlanner (Dang et al., 2020b). The uncertainty in localization and mapping is taken into account during the planning in (Papachristos et al., 2019b) in such a way that among all trajectories arriving to the reference waypoint, the one that minimizes the expected localization and mapping uncertainty is selected. To unify the exploration framework across both legged and aerial platforms, (Kulkarni et al., 2021) have revised (Dang et al., 2020b) and added a cooperation framework that identifies global frontiers in a global graph built from the sub-maps of individual robots. The unified strategy for subterranean exploration using legged and aerial robots in tunnel and urban circuits is presented in (Tranzatto et al., 2022b). Team CERBERUS won in the Systems Track of the Final Event and (Tranzatto et al., 2022a) describes their approach that led to this success.

Team MARBLE presents their system deployed to STIX, the tunnel circuit, and the urban circuit in (Ohradzansky et al., 2021). The aerial robots relied on direct vision-based local reactive control and map-based global path planning. Global path planning is common with ground and aerial robots. Viewpoints are selected based on the frontier voxels covered by the camera FOV and the approximate travel time. In the tunnel circuit, the local reactive control generates velocity commands by steering the UAV towards a look-ahead point from the global path, while being repulsed by nearby obstacles. With this planner, traversing narrow passages was problematic due to noise in the depth image. Thus a new planner was developed for the urban circuit based on voxel-based probabilistic tracking of obstacles (Ahmad et al., 2021). In the Systems Track of the Final Event, team MARBLE gained 3rd place.

A heterogeneous team of robots including UAVs was also deployed by team CSIRO Data61 (Hudson et al., 2022), both in the tunnel and urban circuits. The aerial part of the team consisted of a DJI M210 equipped with the commercially available payload of Emesent Hovermap, and a custom gimballed camera. To explore the environment of the urban circuit, the autonomy utilized an approach based on the direct point cloud visibility (Williams et al., 2020). Team CSIRO Data61 achieved 2nd place in the Systems Track of the Final Event.

Although team NCTU did not participate in the Final Event, their solution (Chen-Lung et al., 2022) to the tunnel and urban circuit showcased originality in the form of autonomous visually localized blimps (Huang et al., 2019). Their navigation was based on policies learned by deep reinforcement learning with simulation-to-world transfer.

Our CTU-CRAS-NORLAB team first participated in the STIX event with a hexarotor platform localized by optic flow (Walter et al., 2018) of the downward-facing camera. The reactive navigation used LiDAR scans to stay in the middle of the tunnel and move forward in a preferred direction at an intersection. The predictive controller (Baca et al., 2016) was forgiving to imprecise localization caused by strenuous optic flow estimation in the whirling dust of the tunnels. The heterogeneous team that secured 3rd place in the tunnel circuit (Roucek et al., 2019) consisted of wheeled, tracked, and aerial robots with different sensor payloads. Instead of unreliable optic flow, the localization of the UAV system (Petrlik et al., 2020) was revamped to rely on 2D LiDAR, HectorSLAM (Kohlbrecher et al., 2011), and state estimation (Petrlik et al., 2021). The hardware platform was also downscaled to a 450 mm diameter quadrotor. The vertical element of the urban circuit called for upgrading the LiDAR to a 3D one, which consequently required a redesign of the whole navigation pipeline (Kratky et al., 2021a) to allow for six Degrees of Freedom (DOF) mobility through the 3D environment. Physically, the platform was based on the same frame as what was used in the tunnel circuit, however prop guards were added to reduce the chance of destructive collision while flying through doors. The CTU-CRAS-NORLAB approach to the urban circuit, which we completed in 3rd place, is described in (Roucek et al., 2020). Although the cave circuit was canceled, extensive preparations were still performed in the sizable Bull Rock cave in South Moravia (Petracek et al., 2021). The exploration depth of the UAV team was greatly extended by a multirobot coordinated homing strategy that focused on extending the communication range of the base station by landing the returning UAVs on the edge of the signal. Based on the lessons learned during these competition and testing deployments (during the 3 years of development UAVs of the CTU-CRAS-NORLAB team achieved >400 flights and traveled >50 km in demanding real world environments) the new approaches presented in this paper were designed.

2.5. Datasets

Due to the challenging nature of the subterranean environments, such as narrow passages, degenerate geometry, and perception degradation, datasets that were collected by the competing teams are valuable to the community as the algorithms can be evaluated on demanding data degraded by the previously mentioned issues. In contrast to the verification often conducted under artificially ideal lab conditions, these datasets present a fair way to compare algorithms in realistic conditions. A SLAM dataset (Rogers et al., 2020a) collected during the tunnel circuit and STIX consists of LiDAR scans, images from a stereo camera and thermal camera, IMU measurements, and RSSI, together with a professionally surveyed ground truth map and measured artifact positions. The dataset from the urban circuit (Rogers et al., 2020b) was recorded using the same sensors with the exception of an added carbon dioxide (CO_2) sensor and the lack of a thermal camera. Data from sensors used for autonomous navigation including color-depth (RGBD) camera, event camera, thermal camera, 2D and 3D LiDARs, IMU, and Ultra-Wide Band (UWB) positioning systems were collected (Koval et al., 2022) by a mobile robotic platform moving through a subterranean environment. Another dataset (Kasper et al., 2019) for comparison of VIO methods contains outdoor, indoor, tunnel, and mine sequences, with ground truth poses obtained by laser tracking the sensors rig. Aerial datasets consisting of unsynchronized LiDAR scans and IMU measurements from UAVs flying in the cave, tunnel, and mine environments are included in this paper,[1] with ground truth poses estimated using a professionally surveyed ground truth map. We also publish the labeled visual detection datasets[2] consisting of images from both UAV and UGV cameras that were used for training of the artifact detection Convolutional Neural Network (CNN). Images from the Tunnel and Urban circuits, Bull Rock Cave, and industrial buildings are included.

3. Contributions

An approach for cooperative exploration of demanding subterranean environments by a team of fully autonomous UAVs in S&R tasks is presented in this paper. Deployment of this approach in the DARPA SubT virtual competition was awarded by 2nd place. The simulation model of the UAV platform designed by our team was used by seven out of nine teams. The crucial contributions of the developed system can be summarized in the following list:

- **A complex approach that can serve as a guide for building a system for Global Navigation Satellite System (GNSS)-denied operations.** The proposed approach was extensively verified in numerous simulated worlds and real physical environments ranging from vast caves, industrial buildings, tunnels, and mines to large outdoor openings. Most importantly, the UAVs were deployed into the intentionally harsh conditions of the DARPA SubT to push them to their limits. The experience gained from hundreds of flights in such conditions are condensed into the lessons learned presented in this paper, which we deem valuable for the field robotics community.
- **Novel mapping structures** are proposed for safety-aware reactive planning over large distances, for compact volumetric inter-robot information sharing, for storing coverage of surfaces by onboard sensors, and for finding a suitable landing spot.
- **Maximization of the probability of detecting a nearby artifact** by searching not only the unexplored space, but also visually covering known surfaces while respecting the limited field of view of the onboard sensors. The detection is coupled with probabilistic estimation of artifact positions based on multitarget tracking and detection-to-hypothesis association, which improves the precision of artifact localization while the robot is moving around the artifact.
- **A novel safety-aware approach to planning** that considers the risk of planned trajectories in addition to the path length in the optimized cost function. In contrast to the state-of-the-art

[1] github.com/ctu-mrs/slam_datasets
[2] github.com/ctu-mrs/vision_datasets

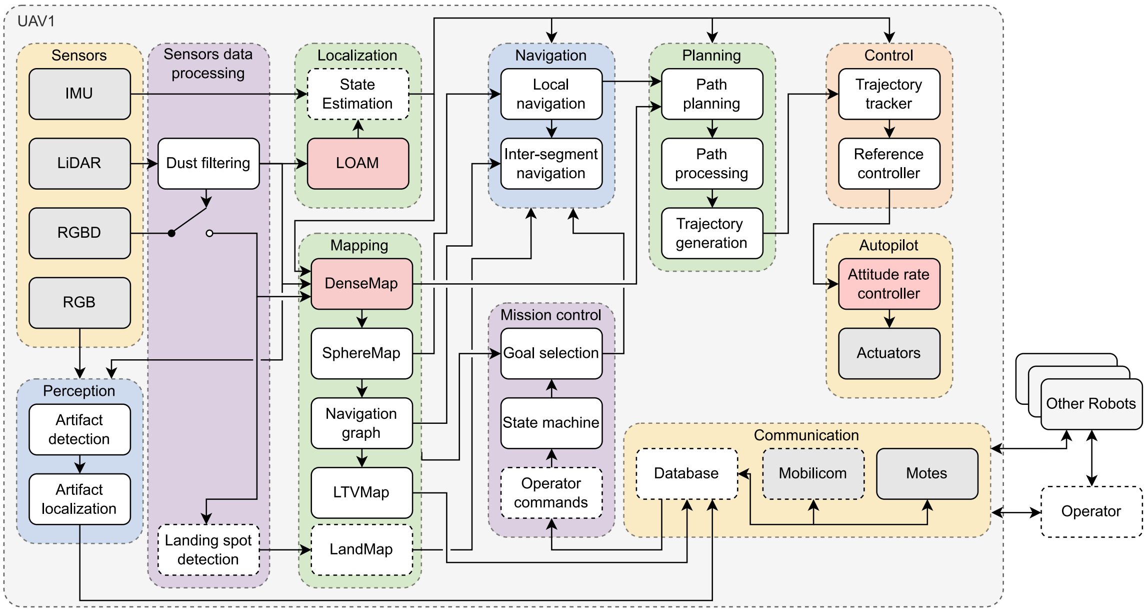


Figure 4. The diagram shows individual modules of the UAV system architecture (as deployed on the hardware platform) grouped into logical categories. Hardware modules are filled with gray, and red distinguishes open source modules not developed by us. The modules marked by dashed lines were used only in Systems Track but not in Virtual Track.

methods, longer paths are selected if the estimated risk of collision is lower than the risk of a shorter path.

- **Full autonomy of the UAV** allows for scalability of the size of the deployed fleet without placing additional workload on the operator. Nevertheless, the operator can override the autonomy with one of the available special commands to change the default search behavior when the UAV is in communication range.
- **The multirobot autonomous search** benefits from a higher number of deployed UAVs that share their topological representations of the environment to cooperatively cover a larger area by biasing the search towards parts unvisited by other agents.

4. System architecture overview

The whole autonomous system of a single UAV consists of software modules, each with different inputs, outputs, and purpose. These modules and their interconnections are depicted in Figure 4 with the individual modules grouped into more general logical categories. The first category includes the physical *Sensors* (Section 5) of the UAV—the IMU, LiDAR, RGB, and RGBD cameras. The description of the important parameters of the used sensors is available in Section 12. Measurements from IMU and LiDAR enter the *Localization* group (Section 6), where a full-state estimate of the UAV is obtained. LiDAR is also used in combination with the RGBD camera for building maps in the *Mapping* module group (Section 7). The *Perception* (Section 10) category focuses on detection and localization of artifacts using all the available sensor data.

Autonomous search through the environment is governed by the *Mission control* category (Section 11), which selects goals (Section 8) based on the current state of the state machine, models of the environment from the *Mapping* group, and possibly also commands from the operator. A coarse path consisting of waypoints to the selected goals is found by the *Navigation* (Section 9.1) and further refined and time-parametrized in the *Planning* modules (Section 9) in order to produce a safe and dynamically feasible trajectory. The *Control* blocks (Baca et al., 2021) track the trajectory and generate attitude rate references for the low-level *Autopilot* that controls the actuators (Section 12).

The operator receives crucial mission status data, topological maps, and, most importantly, detected artifacts through the *Communication* layer (Roucek et al., 2020). This also allows the operator to influence or override the autonomous behavior of the UAV. All transmitted data are received by other UAVs (or other robots, in the case of a heterogeneous team) in the communication range, which serves two purposes: one, the receiving agent can propagate the message further down the network, and, two, the topological maps allow penalizing goals already visited by other robots to better allocate resources over a large area.

5. Spatial perception

The equipment on board UAV platforms within our research group is modular and replaceable to support a wide spectrum of research areas (Hert et al., 2022). In the proposed system for agile subterranean navigation, however, the aerial platform is fixed to ease fine-tuning of the on-board-running algorithms. From the point of perception, it relies heavily on 3D LiDAR from Ouster (SLAM, dense mapping, and artifact localization), and utilizes vertically oriented RGBD cameras for filling space out of FOV of the primary LiDAR sensor, and uses two RGB Basler cameras for artifact detection, supported by powerful LEDs illuminating the scene. The flow of sensory data within the entire system are shown directly in Figure 4.

5.1. Sensors calibration

The intrinsics of LiDAR sensor and RGBD cameras are factory-calibrated whilst monocular RGB cameras are calibrated with standard OpenCV calibration tools, assuming the pinhole camera model. The extrinsics of the sensors (cameras-to-LiDAR and LiDAR to the flight control unit) are given by the CAD model of the robot. To mitigate the effects of inaccuracies in 3D printing, modeling, and assembly, all the camera-to-LiDAR extrinsics are fine-calibrated using a checkerboard camera calibration pattern with known dimensions. The fine-calibration pipeline detects the pattern in both modalities (LiDAR data and RGB image), finds mutual correspondences, and estimates the extrinsics by defining the problem as perspective-n-point optimization minimizing the reprojection error of the mutual correspondences with Levenberg-Marquardt method.

5.2. Filtering observation noise

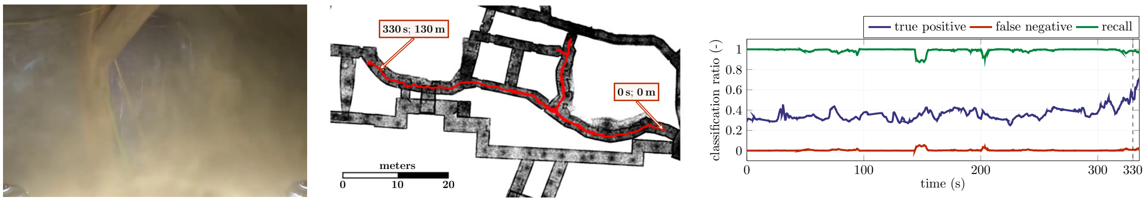
The aerodynamic influence of a multirotor UAV on the environment is not negligible, particularly in confined settings. The fast-rotating propellers generate airflow lifting up light particles of dust and whirling them up in clouds. In environments where the clouds are not blown away but are rather rebounded back to the UAV, the effect on sensory performance might be crippling. To minimize deterioration in perception and its dependent systems (e.g., mapping, localization), the incident noise is filtered out from local LiDAR data.

The idea of robust filtering of dust is based on the method presented in (Kratky et al., 2021a) in which LiDAR data are sorted by the intensity field (measured intensity of the reflected light for a given point) and 10 % of the lowest-intensity data in a local radius from the sensor are removed. In contrast to the baseline method, simpler thresholding is adopted such that a subset $\mathcal{P}_F \subseteq \mathcal{P}$ of LiDAR data $\mathcal{P}$ is preserved. The absence of data sorting lowers the computational load and reduces delay in data processing. The set is given as $\mathcal{P}_F = \mathcal{P}_D \cup \mathcal{P}_I$, where

$$\mathcal{P}_D = \{\mathbf{p} \mid \|\mathbf{p}\| \geq \kappa,\, \mathbf{p} \in \mathcal{P}\}\,, \tag{1}$$

$$\mathcal{P}_I = \{\mathbf{p} \mid \mathcal{I}(\mathbf{p}) > \Upsilon,\, \mathbf{p} \in \mathcal{P} \setminus \mathcal{P}_D\}\,. \tag{2}$$

$\mathcal{I}(\mathbf{p})$ $\left(\mathrm{W\,m^{-2}}\right)$ is the intensity of the reflected light from a point $\mathbf{p}$, κ (m) is a local radius of a filtering sphere with LiDAR data origin at its center, and Υ $\left(\mathrm{W\,m^{-2}}\right)$ is the minimal intensity of preserved



(a) Dense cloud dust around the UAV as viewed in onboard RGB camera at time 330 s. **(b)** Top view on the UAV trajectory. **(c)** Performance of noise classification in LiDAR data in 3 m local radius from the sensor. Average recall reached 99 %.

Figure 5. LiDAR-data noise filtration running onboard a UAV during a 154 m flight in the mine part (the dustiest part) of the DARPA SubT finals environment. The true positive classification in (c) denotes the ratio of correctly classified noise whereas the false negative represents the ratio of noise preserved after the filtration process (i.e., the unfiltered noise) to the size of the point cloud. The data for the classification analysis (c) were obtained by spatially comparing the sensor measurements with the map of the environment provided by the organizers.

data points. With n data points within a radius κ, the computational complexity is reduced to $\mathcal{O}(n)$ from baseline $\mathcal{O}(n \log(n))$. Although to achieve optimal performance the method requires calibration to given environmental conditions, a set of reasonable parameters ($\kappa = 5\,\mathrm{m}$ and $\Upsilon = 30\,\mathrm{W\,m^{-2}}$ throughout many of our real-world deployments in the harshest dust conditions) suffices in the majority of applications. The performance of the dust filtering is analyzed in Figure 5 on an example UAV flight in the mine part (the dustiest zone) of the DARPA SubT finals environment.

The above method is utilizable only for sensory data containing information about the intensity of the reflected light. The rest of the sensors (RGBD cameras) are not processed, but their fusion and utilization are controlled by the amount of filtered noise in the primary LiDAR. Having the cardinality of the point sets defined in (2), the estimated amount of noise can be represented as

$$r_d = 1 - \frac{|\mathcal{P}_I|}{|\mathcal{P} \setminus \mathcal{P}_D|}, \tag{3}$$

where $r_d \in \langle 0, 1 \rangle$ is the ratio of the filtered-out observations to all the observations within the local radius κ. The RGBD cameras are then classified as unreliable (and not used in mapping or for detecting landing feasibility, as marked in Figure 4) if

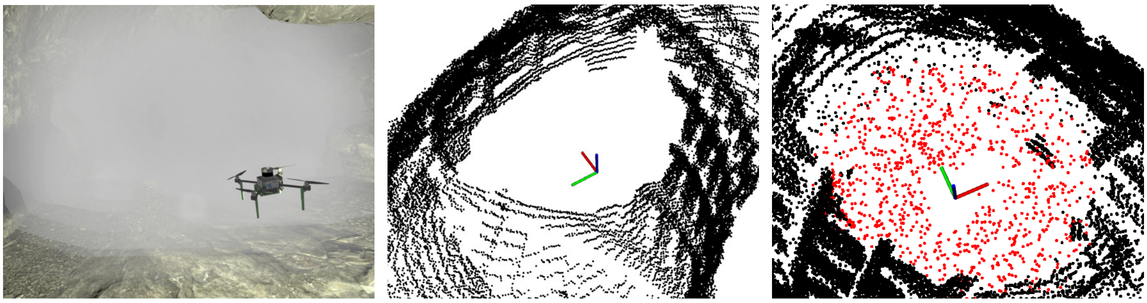
$$r_d > \lambda_d, \tag{4}$$

where $\lambda_d \in \langle 0, 1 \rangle$ is a unitless user-specified threshold. The lower value of λ_d the less amount of noise is integrated into mapping, while greater λ_d lets the connected modules handle the noise themselves. We empirically set the threshold high to $\lambda_d = 0.4$ in our final setup, since our probabilistic mapping pipeline is quite robust to the stochastic noise.

5.2.1. Detecting artificial fog in the virtual environment

The virtual competition contained a fog emitter plugin (see Figure 6) to mimic environmental perception degradation arising from observing smoke, dust, and fog. The plugin spawned a fog cloud when a robot reached the proximity of the emitter. Although our localization pipeline was able to cope with local noise, the inability to filter out the fog particles in a robust way led to a degradation of the local DenseMap, and consequently to blocking local planning which respects strict requirements on collision-free path planning. Thus, in our setup for the virtual challenge, the navigation stack did not try to enter through the fog areas but detected them, maneuvered out of them, and blocked the areas for global planning.

To detect the presence of the UAV within such a fog cloud, a discretized occupancy voxel grid is built from a set of data within a local radius (example data within a radius are shown in Figure 6c). Within this radius is compared the occupancy ratio r (number of occupied voxels to all voxels in the local grid) with maximum occupancy R given by the field of view of the sensor producing the



(a) Visualization of virtual fog in Ignition Gazebo. **(b)** Example 3D LiDAR data outside fog. **(c)** Example 3D LiDAR data inside fog (fog colored locally in red).

Figure 6. Simulated fog and its effect on sensory perception in the virtual environment. A fog cloud (a) spawns when a robot reaches its proximity. The cloud then affects the sensory inputs such that a uniform-distribution noise emerges in LiDAR data corresponding to the fog (c).

data. For each LiDAR or depth sensor, the sensor is classified as being in fog if

$$r_f > \lambda_f R, \tag{5}$$

where $\lambda_f \in \langle 0, 1\rangle$ is a unitless multiplier converting $\lambda_f R$ to a maximal occupancy ratio threshold. The multiplier was set empirically to $\lambda_f = 0.7$ in our final setup.

For depth cameras that are not used for self-localization of the UAV, the in-fog classification solely controls whether the depth data are integrated within the mapping pipeline. However, if a localization-crucial 3D LiDAR is classified to be in fog, a backtracking behavior is triggered within the mission supervisor (see Section 11). The primary purpose of the backtracking is to prevent being stuck in fog and thus the UAV is blindly navigated out of the fog through the recent history of collision-free poses, ignoring occupied cells in the DenseMap (including possible noise from fog measurements). Lastly, detection of fog in a 3D LiDAR blocks the area in global planning.

5.3. Detecting spots safe for landing

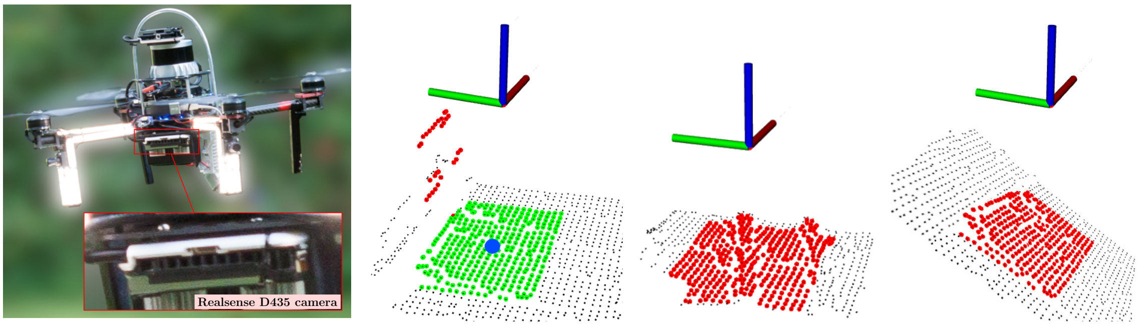
For purposes of artifact detection and spatial mapping, the UAV carries a downward-facing RGBD camera, shown in Figure 7a. Apart from mapping the space below the UAV, the depth data of this camera are used in locating areas safe for landing throughout the UAV flight. If the sensor is marked as reliable according to (4), its depth-data frames are continuously fitted with a plane model whose coefficients are used in the binary classification of safe or unsafe landability respecting the plane-fit quality and deviation of its normal vector from the gravitational vector. The process of deciding on safe landability given a single depth-data frame is visualized in Figure 7 and described in Algorithm 1. The classification assumes that the data frame can be transformed into a gravity-aligned world coordinate frame. Inputs to Algorithm 1 are the square size of safe landing spots $s\,(\mathrm{m})$, the minimal ratio of inliers in RANSAC plane fitting $I_{\min} \in \langle 0, 1\rangle$, and the minimal z-axis component of unit plane-normal vector $N^z_{\min} \in \langle 0, 1\rangle$. The square size s specifies the width of an area on which a UAV can land safely. Selection of s constraints the minimal height above the ground in which a safe landing spot detection may occur. Having a sensor with the minimal field of view $\theta_{\min}$ observing an even planar surface, the condition on line 12 in Algorithm 1 will be true for distance of the camera from the surface $d > d_{\min}$, where

$$d_{\min} = \frac{s}{2\tan\left(\frac{\theta_{\min}}{2}\right)}. \tag{6}$$

The maximal distance $d_{\max}$ is then given by the maximal range of the sensor. In our setup, we utilized Realsense D435 camera with $\theta_{\min} = 58°$ and set $s = 1.2\,\mathrm{m}$ to be marginally greater than the dimensions of our UAV platform (approximately 0.8 m wide). According to (6), the given parameters

Algorithm 1. Detecting spots safe for UAV landing in downward-facing RGBD camera.

```
 1: Input:
 2:     $\mathcal{D}$                                              ▷ Depth-data frame in sensor coordinate frame
 3: Output:
 4:     $\mathcal{L}$                                              ▷ Binary classification for landing: {SAFE, UNSAFE}
 5:     $\mathbf{p}_W$                                             ▷ Position of landing area in the world coordinate frame
 6: Parameters:
 7:     $s$                                                        ▷ Square-size of safe landing spot in meters
 8:     $I_{\text{min}}$                                           ▷ Minimal ratio of inliers in plane fitting
 9:     $N^z_{\text{min}}$                                         ▷ Minimal z-axis component of the normalized plane-normal vector
10: Begin:
11:     $\mathcal{S}$ := cropFrameAtCenter($\mathcal{D}$, $s$)     ▷ Crop frame-centered square with size $s$
12:     if height($\mathcal{S}$) < $s$ or width($\mathcal{S}$) < $s$ then
13:         return: {$\mathcal{L}$ = UNSAFE, $\mathbf{p}_W$ = N/A} ▷ Not safe to land: too close to the ground to decide
14:     $\mathcal{P}$ := fitPlaneWithRANSAC($\mathcal{S}$)         ▷ Fit data with plane using RANSAC
15:     if inliers($\mathcal{P}$) / count($\mathcal{S}$) < $I_{\text{min}}$ then
16:         return: {$\mathcal{L}$ = UNSAFE, $\mathbf{p}_W$ = N/A} ▷ Not safe to land: data are not planar
17:     $\mathcal{P}_W$ := transformToWorldFrame($\mathcal{P}$)    ▷ Transform plane to gravity-aligned frame
18:     if |normal($\mathcal{P}_W$).z| < $N^z_{\text{min}}$ then
19:         return: {$\mathcal{L}$ = UNSAFE, $\mathbf{p}_W$ = N/A} ▷ Not safe to land: ground is too steep for landing
20:     $\mathcal{S}_W$ := transformToWorldFrame($\mathcal{S}$)
21:     $\mathbf{p}_W$ := centroid($\mathcal{S}_W$)                ▷ Express landing spot as the centroid of the depth data in the world
22:     return: {$\mathcal{L}$ = SAFE, $\mathbf{p}_W$}
```



(a) Downward-facing RGBD camera used for landability detection mounted on our UAV platform.

(b) Even planar surface: safe for landing.

(c) Nonplanar surface (rails): unsafe for landing.

(d) Uneven surface: unsafe for landing.

Figure 7. Deciding on landability of a UAV from downward-facing depth data—binary classification to safe (b) and unsafe [(c) and (d)] landing areas. In [(b)–(d)], the UAV is represented by Cartesian axes whereas the depth data are colored in black. The blue sphere in the safe classification (b) denotes the centroid of the plane inliers (colored in green) passed as a feasible landing position to LandMap (see Section 7.5).

yield the minimal distance of the sensor from the ground in detecting the landability to $d_{\text{min}} = 1.08\,\text{m}$, with $d_{\text{max}} = 10\,\text{m}$ specified by the manufacturer. If the input data frame $\mathcal{D}$ contain noise with nonplanar distribution, the condition on line 15 will classify the data as unsafe. The plane-fit and landability classification parameters $I_{\text{min}} = 0.9$ and $N^z_{\text{min}} = 0.7$ were found empirically for the given sensory setup. Positions classified as safe for landing on line 21 are passed to LandMap described in Section 7.5.

6. Localization

Accurate and reliable localization is critical for most other parts of the system. The ability of the reference controller to track the desired state depends largely on the quality of the available

Table 2. Approximate distribution of the environment cross-section as announced by the organizers before the Final Event.

Cross-section (m^2)	Distribution
<5	65%
5-100	20%
>100	15%

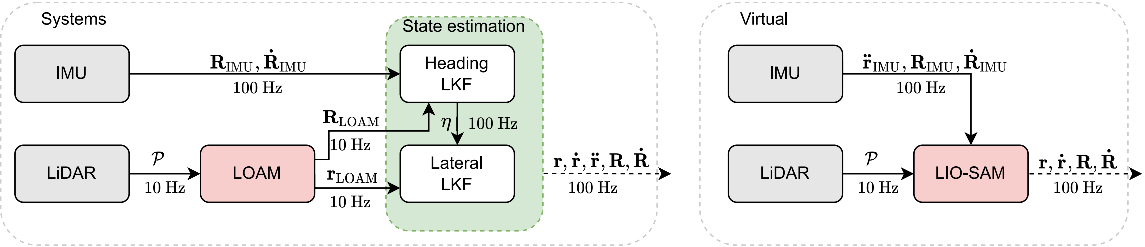

Systems
IMU
$\mathbf{R}_{\text{IMU}}, \dot{\mathbf{R}}_{\text{IMU}}$
100 Hz
State estimation
Heading LKF
$\mathbf{R}_{\text{LOAM}}$
10 Hz
η
100 Hz
LiDAR
$\mathcal{P}$
10 Hz
LOAM
$\mathbf{r}_{\text{LOAM}}$
10 Hz
Lateral LKF
$\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \mathbf{R}, \dot{\mathbf{R}}$
100 Hz
Virtual
IMU
$\ddot{\mathbf{r}}_{\text{IMU}}, \mathbf{R}_{\text{IMU}}, \dot{\mathbf{R}}_{\text{IMU}}$
100 Hz
LiDAR
$\mathcal{P}$
10 Hz
LIO-SAM
$\mathbf{r}, \dot{\mathbf{r}}, \mathbf{R}, \dot{\mathbf{R}}$
100 Hz

Figure 8. The diagram shows the flow of data among individual localization modules for the Systems Track (left) and Virtual Track (right). The 3D LiDAR supplies A-LOAM or LIO-SAM with the laser scans in the point cloud form $\mathcal{P}$. Assisted by the orientation $\mathbf{R}$ from the IMU, A-LOAM produces a position estimate $\mathbf{r} = [x, y, z]^T$ that is fed into the *State estimation* block, which outputs the full state estimate. In the case of the virtual pipeline, the IMU data fusion is executed in LIO-SAM, and thus the state estimation module is not needed thanks to the sufficient accuracy of both lateral and heading components.

state estimate. In the narrow environments which are often present in subterranean environments (see Table 2 for cross-section distribution in the Final Event), minimizing the control error is crucial to avoid collisions. Multirobot cooperation assumes the consistency of maps created by individual robots. If the maps of two robots are not consistent due to errors in localization, the multirobot search might be suboptimal. For example, an unvisited goal can be rejected as already reached by a robot with an inconsistent map. Moreover, the localization accuracy influences the position error of a reported artifact. A UAV with localization drift over 5 m can detect and perfectly estimate the position of an artifact. Nevertheless, the report may never score a point since the position of the UAV itself is incorrect.

Our approach relies on a LiDAR sensor for localization as the laser technology proved to be more robust to the harsh conditions of the subterranean environment than the vision-based methods. We have been using LiDAR since the Tunnel circuit (Petrlik et al., 2020) where a lightweight 2D LiDAR aided by a rangefinder for measuring above ground level (AGL) height was sufficient for navigation in tunnels with a rectangular cross-section. The more vertical environment of the urban circuit required redesigning the localization system to use 3D LiDAR for navigating in 3D space (Kratky et al., 2021a).

The localization system deployed in the Final Event and presented in this manuscript builds upon the solution proposed in (Kratky et al., 2021a) and is divided into two modules: the localization algorithm and the state estimation method. Figure 8 shows the data flow in the localization pipeline. We have based the localization on the A-LOAM implementation of the LiDAR Odometry and Mapping (LOAM) algorithm (Zhang and Singh, 2014) for the Systems Track and the LiDAR Inertial Odometry via Smoothing and Mapping (LIO-SAM) (Shan et al., 2020) for the Virtual Track. Our implementation[3] has been tested in a real-time UAV control pipeline throughout multiple experimental deployments as part of our preliminary works (Kratky et al., 2021a; Petracek et al., 2021) and in the DARPA SubT competition.

[3] github.com/ctu-mrs/aloam

6.1. A-LOAM

The A-LOAM implementation of the LOAM (Zhang and Singh, 2014) algorithm utilizes the laser scans from a multiline LiDAR to obtain its 6-DOF pose. To achieve real-time performance and accurate pose estimation at the same time, the method is divided into two parts.

The first part of the algorithm processes the incoming data at the rate of their arrival and estimates the rigid motion between the consecutive point clouds $\mathcal{P}_k$ and $\mathcal{P}_{k+1}$ obtained at the timestamps t_k and t_{k+1}, respectively. The process starts with finding geometric features in the input point cloud $\mathcal{P}_{k+1}$. The points are first sorted by the smoothness of their local neighborhood, and then those which are the least and most smooth are selected as edge and planar features, respectively. To achieve a more uniform distribution of features, the point cloud is divided into regions of the same size, and each region can contain only a limited number of edge and planar feature points. A point cannot be chosen as a feature point if there is already a feature point in its local neighborhood. A correspondence is found in $\mathcal{P}_k$ for each edge/planar point from $\mathcal{P}_{k+1}$. These correspondences are then weighted by their inverse distance, and correspondences with the distance larger than a threshold are discarded as outliers. Finally, the pose transform $\mathbf{T}^L_{k+1}$ between $\mathcal{P}_{k+1}$ and $\mathcal{P}_k$ is found by applying the Levenberg-Marquardt method to align the correspondences.

The second part estimates the pose of the sensor in the map $\mathcal{M}_k$, which is continuously built from the feature points found by the first part of the algorithm. First, $\mathcal{P}_{k+1}$ is projected into the map coordinate system to obtain $\mathcal{P}^W_{k+1}$. Then, feature points are searched similarly to as is done in the first part, with the difference being that 10 times more features are found. Their correspondences are found in $\mathcal{M}_k$, which is divided into cubes with 10 m edges. The correspondences are searched for only in the cubes intersected by the $\mathcal{P}^W_{k+1}$ to keep the run-time bounded. The transform $\mathbf{T}^W_{k+1}$ between $\mathcal{P}^W_{k+1}$ and $\mathcal{M}_k$ is obtained with the same steps as in the first part. Due to the 10-times greater amount of correspondences and search through a potentially larger map, this is a much slower process than the first part.

Thanks to the combination of both parts, the algorithm outputs the pose estimate of the rate of the LiDAR, with drift bounded by slower corrections that snap the pose to the map.

6.2. State estimation

For precise and collision-free navigation through a cluttered narrow environment, which typically appears in subterranean S&R scenarios, the control stack requires a smooth and accurate state estimate at a high rate (100 Hz). The *State estimation* module provides such an estimate through the fusion of data from Advanced implementation of LOAM (A-LOAM) and IMU. It also does this by applying filtering, rejection, and prediction techniques. We provide only a brief description of the estimation process as it is not viewed as the primary contribution and has already been presented in (Baca et al., 2021).

The state vector of the UAV is defined as $\mathbf{x} = [\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \mathbf{R}, \dot{\mathbf{R}}]^T$. The position $\mathbf{r} = [x, y, z]^T$, its first two derivatives of $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$, the orientation in the world frame $\mathbf{R}$, and the angular velocities $\dot{\mathbf{R}}$ include all the dynamics required by other onboard algorithms. Even though the position $\mathbf{r}$ is provided by the A-LOAM algorithm, the rate of the position updates is too low for the control loop. Furthermore, the velocity and acceleration vector is not known, and must thus be estimated. A Linear Kalman Filter (LKF) of a point mass model with position, velocity, and acceleration states is employed to estimate the unknown variables at the desired rate.

While the IMU of the onboard autopilot provides the orientation $\mathbf{R}$, the heading[4] η is prone to drift due to the bias of the gyroscopes in Micro-Electromechanical Systems (MEMS) IMUs. We correct this drift in a standalone heading filter, which fuses $\dot{\mathbf{R}}$ gyro measurements with A-LOAM

[4] Heading is the angle between the heading vector and the first world axis. The heading vector is the direction of the forward-facing body-fixed axis projected onto the plane formed by the horizontal axes of the world frame, as formally defined in (Baca et al., 2021).

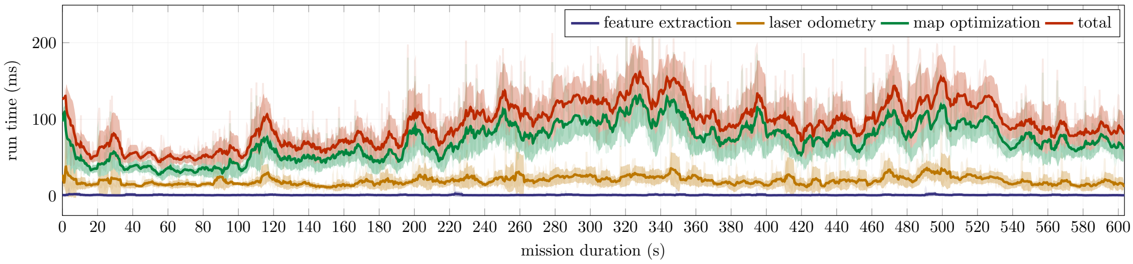


Figure 9. The computation time of the most demanding parts of the A-LOAM algorithm is plotted with respect to the time in the mission that was conducted in simulation. The total time is the sum of all three parts. The darkest colors depict moving mean, the medium dark bands represent the moving standard deviation, and raw data are shown by the lightest colors. The moving statistics are calculated over 1 s long time window. On average, the feature extraction takes 1 ms, the laser odometry 19 ms, the map optimization 91 ms, and, in total, the pose estimate is obtained in 111 ms.

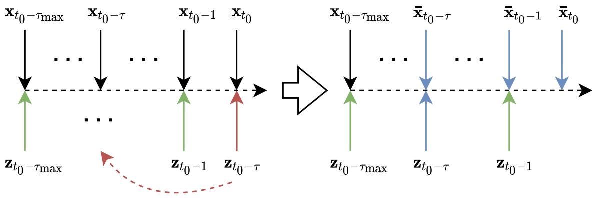


Figure 10. The left time sequence shows the situation in the filter after the arrival of delayed correction $\mathbf{z}_{t_0-\tau}$ at time t_0. The green arrows represent corrections applied at the correct time. The delayed $\mathbf{z}_{t_0-\tau}$ would be fused at t_0 in a traditional filter, resulting in a suboptimal state estimate. However, thanks to the buffering of state and correction history, it is fused into the correct state at time $t_0 - \tau$. The states after $t_0 - \tau$ had to be recalculated to reflect the correction $\mathbf{z}_{t_0-\tau}$, which is shown by the blue color in the right time sequence.

η corrections. Corrections from the magnetometer are not considered, due to the often-occurring ferromagnetic materials and compounds in subterranean environments.

The processing of a large quantity of points from each scan and matching them into the map takes 111 ms on average (see Figure 9 for run time analysis) for the onboard Central Processing Unit (CPU). The empirical evaluation shows that the controller of the UAV becomes increasingly less stable when the state estimate is delayed for more than 300 ms. To reduce the negative effect of the delay on the control performance, we employ the time-varying delay compensation technique (Pritzl et al., 2022a). We define the delay as $\tau = t_{\mathbf{T}_{k+1}} - t_{\mathcal{P}_{k+1}}$, i.e., the time it took LOAM to compute the pose transform after receiving the point cloud from LiDAR. The core of the method is a buffer $\mathbf{Q_x}$ containing the past states $\mathbf{x}_{\langle t_0-\tau_{\max}, t_0\rangle}$, and buffer $\mathbf{Q_z}$ having the past corrections $\mathbf{z}_{\langle t_0-\tau_{\max}, t_0\rangle}$ of the filter. The length of the buffer is not fixed, but data older than the expected maximum delay $\tau_{\max}$ are discarded to keep the buffer size bounded. When a new delayed measurement $\mathbf{z}_{t_0-\tau}$ arrives at time t_0, it is applied as a correction to the state $\mathbf{x}_{t_0-\tau}$ in $\mathbf{Q_x}$. The corrected state $\bar{\mathbf{x}}_{t_0-\tau}$ replaces $\mathbf{x}_{t_0-\tau}$. All subsequent states $\mathbf{x}_{\langle t_0-\tau, t_0\rangle}$ are discarded from $\mathbf{Q_x}$, and replaced by the states $\bar{\mathbf{x}}_{\langle t_0-\tau, t_0\rangle}$ propagated from $\bar{\mathbf{x}}_{t_0-\tau}$, using regular prediction steps of the filter with all corrections from $\mathbf{Q_z}$. Figure 10 visualizes the sequence of performed actions. Thus we acquire a time-delay compensated state estimate which, when used in the feedback loop of the UAV controller, allows for stable flight with a delay of up to 1 s. The effect that increasing the delay has on the control error is plotted in Figure 11.

6.3. LIO-SAM

LIO-SAM (Shan et al., 2020), used in the Virtual Track approach, utilizes IMU integration on top of dual factor-graph optimization. The first factor-graph optimization is similar to the A-LOAM

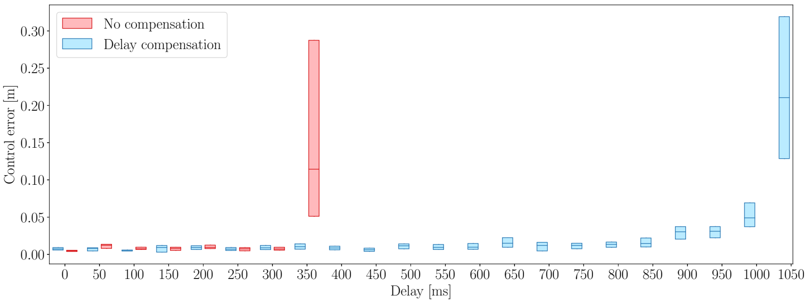


Figure 11. The box plot shows the median with lower and upper quartiles of the control error with respect to the delay of the position estimate used in the feedback loop. The data were obtained in simulation by artificially increasing the delay of ground truth position in 50 ms increments. Without compensation, the system becomes unstable after exceeding 300 ms delay, which results in oscillation-induced control error at 350 ms. The control error for the longer delay is not shown, because the high amplitude of oscillations led to a collision of the UAV. The highest delay with compensation is 1000 ms when the system has over a 5 cm control error, but is still stable. The UAV stability is lost at 1050 ms delay.

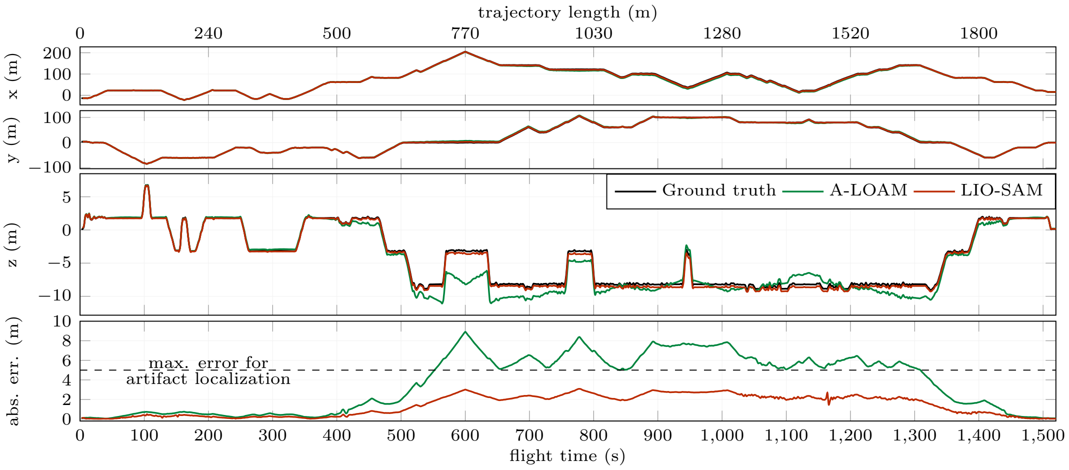


Figure 12. The performance of A-LOAM and LIO-SAM during a single flight within *Finals Prize Round World 01* (see Figure 46) of the DARPA SubT virtual environment. A-LOAM does not fuse the inertial measurements which assist LIO-SAM during LiDAR-scan matching in areas of the environment where such matching suffers from geometric degeneration, in the context of solving optimization problems. The selected environment contains a variety of narrow vertical passages where the performance of narrow-FOV LiDAR perception is limited, leading to drift in the ego-motion estimation that is clearly visible in the A-LOAM method. The LIO-SAM method was shown to achieve sufficient accuracy and low drift during long-term and arbitrary 3D navigation within a simulated environment.

mapping pipeline as it first extracts geometrical features out of raw LiDAR data and registers them to a feature map, with the motion prior given by the second optimization pipeline. The second factor-graph optimization fuses the mapping output with IMU measurements and outputs fast odometry used in the state estimation pipeline. The first graph is maintained consistently throughout the run, whereas the second graph optimization is reset periodically to maintain real-time properties.

In a simulated environment, LIO-SAM yields greater accuracy than A-LOAM for its fusion of inertial measurements with precisely modeled and known characteristics. A comparison of both the methods within the simulated environment is summarized in Figure 12. In the real world, the measurements of an IMU rigidly mounted on board a UAV contain a wide spectrum of large stochastic noise. During empirical testing, the integration method in LIO-SAM was shown to not be

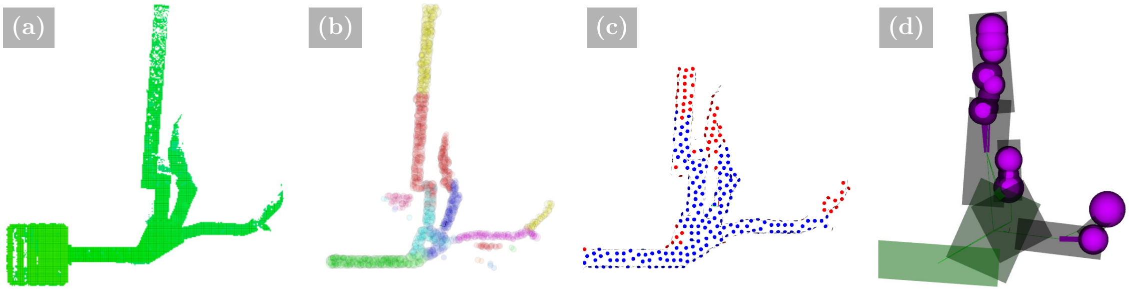


Figure 13. Top view of the used mapping structures from the intersection of the virtual Final Event map. DenseMap (a) is used for short-distance planning, SphereMap (b) for safety-aware long-distance planning, FacetMap (c) for storing surface coverage, and LTVMap (d) for compact topological information sharing among robots.

robust towards the unfiltered noise while frequency-band and pass filters induced significant time delays, destabilizing the pipeline completely. For the inability to accurately model the noise, real-world laser-inertial fusion is done manually by smoothing over a short history of past measurements (see Section 6.2).

7. Mapping

In this section, we present our approach to mapping the explored environments. As each task has specific requirements on the map properties, we designed multiple spatial representations, each of which is structured for a particular task. In particular, DenseMap (Figure 13a) is utilized for short-distance path planning; FacetMap (Figure 13b) for surface coverage tracking; SphereMap (Figure 13c) for fast and safe long-distance path planning; lightweight topological-volumetric map (LTVMap) (Figure 13d) for compressed, topological, and mission-specific information sharing between robots in low bandwidth areas; and LandMap (Figure 15) for representing feasible spots for safe UAV landing. These maps and the methods for building them are presented in this section.

7.1. DenseMap

Local information of the UAV is combined within a dense map to serve as the basis for the entire navigation stack, as described in (Kratky et al., 2021a). The map integrates information in a dense, probabilistic manner using an efficient octree structure implemented within the OctoMap (Hornung et al., 2013) library. During the map update, the data of each input modality producing spatial measurements are used to update the map with respect to the pose estimate correlating to the timestamp of the respective measurement. The data to be integrated are first cleared of any observation noise (see Section 5). The ray of each remaining spatial measurement is integrated within a discretized representation of the environment using the Bayes rule and ternary classification to the unknown, free, and occupied voxels. The output of dense mapping is convertible to other navigation representations and serves as the fundamental structure for local planning and dynamic obstacle detection.

To retain maximum information under constraints on real-time performance, the voxelization resolution is selected such that a scan insertion is processed at 5 Hz, at worst. The resolution can be locally increased if path planning demands a decrease in discretization errors. This is a useful feature for improving safety and repeatability in navigating highly narrow passages. To maintain the map structure, the local resolution is controlled by a factor n such that the local resolution equals $r/2^n$ with r being the default resolution of the dense map. In our sensory and computation setup, the default resolution is empirically set to 20 cm, reduced by a factor of $n = 2$ to 5 cm for navigating narrow passages, if required. The integrated data consist of LiDAR measurements and depth estimates of two RGBD cameras. These sensors are mounted on-board UAVs so that the spatial observations cover roughly all directions around the robot, enabling almost arbitrary UAV-motion planning in collision-free 3D space.

7.2. SphereMap

To enable the UAV to quickly evaluate the travel time and risk caused by flying near obstacles while also pursuing any given goal, we developed a multilayer graph structure that uses volumetric segmentation and path caching, called SphereMap (Musil et al., 2022). All three layers of the SphereMap are updated near the UAV in every update iteration, which runs at approximately 2 Hz.

Path planning in the SphereMap depends on only one parameter c_R, which we call *risk avoidance*. It is used to trade path safety for path length. For long-distance planning, we disregard UAV dynamics and only take into account the path length and obstacle clearance along the path. We define the path cost between points $\mathbf{p}_1$ and $\mathbf{p}_2$ as

$$D(\mathbf{p}_1, \mathbf{p}_2) = L + c_R R, \tag{7}$$

where L is the path Euclidean length summed over all edges of the path in the sphere graph, and $R \in [0, L]$ is a risk value computed by examining the radii of the spheres along the path. For example, a path with all spheres with radii at the minimal allowed distance from obstacles would have $R = L$, and a path through open space with large sphere radii would have $R = 0$.

The lowest layer of the SphereMap is a graph of intersecting spheres, shown in Figure 13b. It is constructed by filling the free space of an obstacle k-d tree built from the DenseMap with spheres at randomly sampled points. The graph is continuously built out of intersecting spheres, and then by pruning the spheres that become unsafe or redundant. The radii of the spheres carry obstacle clearance information, which is used for path risk evaluation.

The second layer of the SphereMap is a graph of roughly convex segments of the sphere-graph. It is updated after every update of the sphere graph by creating and merging segments until every sphere in the graph belongs to a segment.

The third and last layer of the SphereMap is a navigation graph. For every two adjacent segments, we store one sphere-sphere connection, which we call a *portal* between the segments, as in (Blochliger et al., 2018). These portals form the vertices of the navigation graph. At the end of every SphereMap update iteration, we compute which paths are optimal according to the path cost from (7) between all pairs of portals of a given segment. The paths are computed only inside that given segment. If the segments are kept small (tens of meters in length), the recomputation is reasonably fast. The optimal portal-portal paths form the edges of the navigation graph. The UAV uses the navigation graph to quickly find long-distance paths between any two points in the known space by planning over the edges of the navigation graph, and then by only planning over the sphere graph in the first and last segments of the path.

7.3. FacetMap

The occupancy octree and SphereMap maps are sufficient for volumetric exploration. However, the goal of the DARPA SubT challenge was to locate artifacts, most of which could be detected only from cameras. Because the FOV of our UAVs' cameras did not cover the entire FOV of the LiDAR and depth cameras, not all occupied voxels in the occupancy map could be considered as "covered by cameras." For this reason, we developed another map, called FacetMap, illustrated in Figure 14. This map is a simple surfel map, with the facets stored in an octree structure, each having an orientation, a coverage value, and a fixed size. The FacetMap is built by computing the normals of the occupancy map at sampled occupied points, and creating facets with a set resolution if there are no existing facets with a similar normal nearby. The facets are updated (i.e., added or deleted) periodically at approximately 2 Hz in a cube of pre-defined size around the UAV.

Each facet holds a coverage value that is, for simplicity, defined as binary. A facet is marked as covered if the facet center falls into the FOV of any camera, and the ray from the camera to

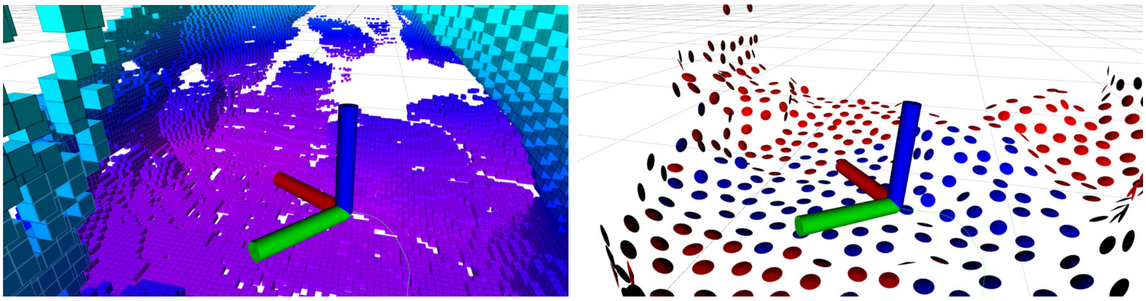


Figure 14. Illustration of the FacetMap in simulation as described in Section 7.3. The map is built from the DenseMap (left) by finding normals of sampled points. The orientation of the visualization discs (right) is determined by the facet's normal, and the color by whether the facet was covered by the UAV's front-facing cameras or not.

the facet center is at an angle lower than a defined threshold from the facet's normal, so as to not mark surfaces as covered if they are viewed at a very skewed angle. The angle threshold was set empirically to 78° in the competition. Angles larger than the threshold reduced the probability of successfully detecting artifacts. The covered facets stay in the map even if the underlying occupancy map shifts (e.g., when an obstacle moves). As described in Section 8.2.3, one strategy used in our system uses this map to cover as much of the surface as possible while flying between volumetric exploration viewpoints. The strategy in Section 8.2.2 uses this map to completely cover surfaces of a dead-end corridor before backtracking to search a different area. Coverage of entire regions of the SphereMap can also be easily computed and then stored in the LTVMap, as described in Section 7.4.

7.4. LTVMap

Distributing all of the maps described in this chapter among the UAVs would be highly demanding for the communication network. As such, we have developed the lightweight topological-volumetric map (LTVMap), which combines the necessary mission-related information from the other maps and can be quickly extracted from the SphereMap and sent at any time.

This map consists of an undirected graph, where each vertex is created from a free-space segment in the original SphereMap and the edges are added for all of its adjacent segments. Each vertex holds an approximation of the segment's shape. In our implementation, we use four DOF bounding boxes (with variable size and rotation along the vertical axis) for shape approximation, though any other shape could be used.

For cooperative exploration purposes, the frontier viewpoints (described in Section 8.1) found by a given UAV are also sent in the LTVMap, with each viewpoint being assigned an information value and segment from which the viewpoint is reachable. For surface coverage purposes, every segment in the LTVMap also holds a single numerical value representing the percentage of relevant surfaces covered in that segment. This value is computed by projecting points from the facets of the FacetMap and counting the points that fall into every segment. Further description and analysis of LTVMaps can be found in (Musil et al., 2022). These LTVMaps are shared among robots, and are used for cooperative search planning onboard UAVs, as described in Section 8.3.

7.5. LandMap

As described in Section 5.3, a downward-facing RGBD camera detects areas safe for landing. These areas are continuously collected within an unconnected set and stored in a sparse point-cloud manner with minimum mutual distance 5.0 m, low enough for avoiding unnecessary long paths to the nearest landing spot while keeping the LandMap memory-light even for large environment. An example of

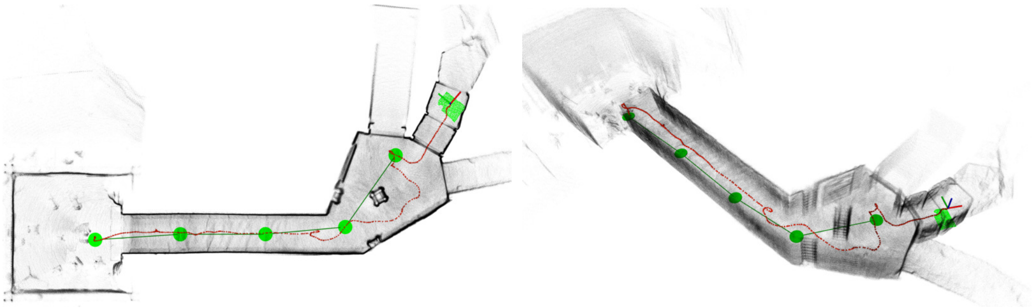


Figure 15. Example of the LandMap with resolution of 5 m built in the beginning of the DARPA SubT systems Final Event after 70 s of a UAV flight. The UAV is represented by the Cartesian axes with its trajectory colored in red. The LandMap incorporates the spots classified as safe for UAV landing (green circles) which are used during the UAV homing phase of the mission to ensure safety during the landing procedure.

the LandMap is shown in Figure 15. During the homing phase of the mission, the UAV navigates to an area connected to the ground station via the communication network (see Section 11.4). After reaching this area, the UAV navigates towards a safe landing spot as indicated by the LandMap, which is closest to its current pose (see mission state machine in Figure 27). While flying towards the LandMap-selected spot, the UAV lands sooner if the ground below the UAV is classified as safe-for-landing in the current RGBD data. The landing spots previously identified as safe are, once more, verified before landing in order to ensure safety in dynamic environments. If the spot is no longer safe for landing, it is invalidated and the UAV is navigated to the next closest landing spot.

8. Autonomous search

Since communication between robots in subterranean environments can never be ensured, the UAVs in our system operate completely autonomously and only use information from other robots to update their goal decision (e.g., blocking frontiers leading to areas explored by other robots). The system can also be controlled at a very high level by the human operator, which is described in Section 11.2. This section describes the high-level search autonomy of our system.

8.1. Informative viewpoint computation and caching

For exploration purposes, the UAVs in our system do not consider the information gain along trajectories, but rather sequences of discrete viewpoints, so that we can have a unified goal representation for both local and global search planning. These viewpoints are divided into places at which a UAV could obtain some volumetric information, called *frontier viewpoints*, and the points at which a UAV could cover some not-yet-covered surfaces with its cameras, called *surface coverage viewpoints*. Each viewpoint ξ, comprising of position $\mathbf{p}_\xi$ and heading φ_ξ, is therefore assigned some information value $I(\xi)$. In our approach, the information gain of frontier viewpoints ξ_F and surface viewpoints ξ_S is computed as

$$I(\xi_F) = c_F \frac{n_{\text{unk}}}{n_{\text{rays}}}, \qquad I(\xi_S) = c_S n_{\text{unc}}, \tag{8}$$

where $n_{\text{unk}}/n_{\text{rays}}$ is the ratio of rays cast in the UAV's depth cameras' and LIDAR's FOVs that hit an unknown cell of the occupancy map before hitting an occupied one or going out of range. Similarly, n_{unc} is equal to the number of uncovered facets of the FacetMap, hit by rays that are cast in the UAV's RGB cameras' FOVs. The constants c_F and c_S are empirically tuned to alter the UAV's next viewpoint selection and hence, its behavior. Additionally, a positive or negative bias c_{FS} can be added to the information value of either function to make the UAV prefer one type of viewpoints more.

The UAV does not sample and evaluate viewpoints on-demand after reaching some viewpoint, rather it continually samples viewpoints in its vicinity at a given rate and stores them into a map of cached viewpoints. Only viewpoints that have $I(\xi)$ above some threshold, are safe, not too close to another informative viewpoint, and not blocked by mission control are stored. The viewpoints are also pruned from the map if they become uninformative or if a better viewpoint is nearby. Lastly, viewpoints that were found in a previous update and are now outside the local update box, are kept as global goals and are pruned more aggressively than the local goals. This approach continually produces a map of informative viewpoints that is denser near the UAV and sparse in the rest of the environment.

8.2. Single-UAV autonomous search planning

In our approach, the UAV can be in three states of autonomous search—*locally searching*, *traveling to goal* or *returning*, and the goal planning and evaluation is divided into local and global planning, as in (Dang et al., 2019b). In all of these states, reachability determination and path planning to any given goal is performed using the rapid long-range path finding provided by the SphereMap, described in Section 7.2. The transitions between the three states are fairly simple—if there are informative and reachable viewpoints near the UAV, the UAV is in the locally searching state and tries to always keep a sequence of two viewpoints. These are given to the trajectory planning pipeline so that the UAV doesn't stop at each viewpoint and compute the next best one. This is done by performing a local replanning of the sequence whenever the UAV is getting close to a viewpoint.

When there are no reachable viewpoints near the UAV or when new information is received from the operator or other robots, a global replanning is triggered.

The global replanning, inspired by (Dang et al., 2020b), computes paths to all stored informative viewpoints (not only in the local search box) and evaluates them. The best viewpoint is then set as a goal to the long-distance navigation pipeline described in Section 9.1. Finally, the *returning* state is triggered when the global planning does not find any reachable goals, or if the operator demands it, or if $t_{\text{home}} < c_H t_{\text{battery}}$, where t_{home} is the estimated time of flight needed to return to the base station, t_{battery} is the estimated remaining flight time, and c_H is an empirically tuned constant. The value of t_{home} is computed from the UAV's average flight speed, and a path found through the SphereMap to the base station. If there is no path to the base station, the UAV will instead try to return along a tree of visited positions, which is built specifically for this purpose, so that for example if a path is only temporarily blocked, the UAV will fly to the roadblock, and if it is removed, will continue flying to the base station. The UAV can also recover from this state, if it is returning due to having found no reachable goals, and suddenly some goals become reachable again. When the UAV gets close to the goal, it switches back to the *locally searching* state.

The reward functions used to evaluate goals govern the behavior of the UAV while searching the environment, and as such, they define the search strategy of the UAV. For simplicity, we made the local planner and global planner use the same reward function in a given strategy, with only one difference, that the local planner can add a penalty to local goals, based on the UAV's current momentum and heading, to allow for smoother local search, which is a highly simplified version of the local viewpoint tour planning done by (Zhou et al., 2021b). These strategies and their corresponding reward functions were utilized in the challenge:

8.2.1. Greedy search strategy (GS)

The chosen reward function for selecting the next best viewpoint ξ from the current UAV viewpoint ξ_{UAV} (the UAV's current position and heading) can be written as

$$R_{\text{GS}}(\xi_{\text{UAV}}, \xi) = I(\xi) - D(\xi_{\text{UAV}}, \xi), \tag{9}$$

where $I(\xi)$ is the information value of the viewpoint (described in Section 8.1) and D is the best path cost computed in the SphereMap (described in Section 7.2). This type of reward was selected for its simplicity, which allows easy debugging and tuning of UAV behavior. It is also easier to extend this reward function to the multi-UAV cooperation reward functions in Section 8.3. This reward function for controlling the next best goal selection thus depends on the constants c_F, c_S, c_{FS} described in Section 8.1 and the risk-awareness constant c_R used in path planning, which can be used to tune the search based on the desired behavior. The constants c_F, c_S, c_{FS} control whether and how much the UAV prefers frontier viewpoints or surface viewpoints, while c_R is set according to the desired risk avoidance. This reward function is very simple and can take the UAV in various directions, leaving behind uncovered surfaces in faraway places. The next strategy aims to solve this.

8.2.2. Dead end inspection strategy (DEI)

A more thorough reward function can be written as

$$R_{\mathrm{DEI}}(\xi_{\mathrm{UAV}}, \xi) = I(\xi) - D(\xi_{\mathrm{UAV}}, \xi) + (D(\mathbf{p}_{\mathrm{HOME}}, \xi) - D(\mathbf{p}_{\mathrm{HOME}}, \xi_{\mathrm{UAV}})). \tag{10}$$

This strategy adds the difference in path costs to the base station position $\mathbf{p}_{\mathrm{HOME}}$ from the evaluated viewpoint and from the UAV. This greatly increases the value of viewpoints that are deeper in the environment, relative to the UAV. Using this reward function, the UAV will most likely first explore frontiers until reaching a dead-end, and then thoroughly cover surfaces from the dead end back to the base, analogous to a depth-first search.

8.2.3. Viewpoint path enhancement strategy (VPE)

The third strategy used on the UAVs is not a change of the reward function, but rather a simple way to increase surface coverage when the UAV is flying through long stretches of explored but not perfectly covered space, either in the DEI or GS strategy. If VPE is enabled and the UAV is flying to a distant goal, then we periodically take the short-distance trajectory from the local path planner (described in Section 9), sample it into multiple viewpoints, and try to perturb these viewpoints to increase surface coverage, while not increasing the flight time too much. Thus we fully utilize the agility of quadcopter UAVs, as they can easily turn from side to side while flying in a given direction.

8.2.4. Comparison of the strategies

During pre-competition testing, the three strategies mentioned above proved to be nearly identical in the total amount of covered volume and surfaces. However, there are serious differences in the overall behavior and what it means for cooperation. The GS strategy on average covers the most volume and surfaces but leaves behind many patches of surfaces or frontiers in very far-away places, due to its greedy nature. The VPE strategy just slightly alters the GS strategy to cover more surfaces in total at the cost of less explored volume but also leaves unfinished goals behind. This has generally been very useful in areas with long corridors that have a high amount of short branches leading off from the main corridor, such as in tunnels or cramped urban areas because the VPE strategy will force the UAV to peek into the corridors, but not to rigorously explore them as with the DEI strategy. The DEI strategy usually covers less space and surfaces in total, but what is most important—it does not leave uncovered and unexplored parts of the environment behind, meaning that for cooperative missions, no other UAV needs to go to that space again, as that space has been completely covered. This is essential in longer missions to ensure complete coverage of the environment.

8.3. Probabilistic cooperative search planning

Our approach to multi-UAV search planning was to make the UAVs completely autonomous and decentralized by default, while also being able to share important information and use it for their own planning. Each UAV always keeps the latest version of the LTVMap (described in Section 7.4) received from a given UAV. When a new LTVMap is received, every newest received map currently being stored onboard the UAV is updated by every other newest received map, as well as by the LTVMap constructed from the UAV's own SphereMap.

The updating is done so that the frontier viewpoints, sent along with each LTVMap, which fall into explored space in other LTVMaps, are blocked. This is difficult to do in a deterministic manner due to map drift and other inaccuracies. Therefore, we approached this problem similarly as in (Burgard et al., 2005) by gradually decreasing the reward of frontier viewpoints whenever the viewpoint falls into the explored space of any segment's bounding box in a received LTVMap. Because the LTVMap bounding boxes are a very rough approximation of the segments' true shape, this reward decreasing is weaker at the edges of the bounding boxes and strongest when the viewpoint lies deep inside the bounding box. Each frontier viewpoint in any LTVMap is assigned a likelihood $l(\xi \in V_{\text{exp}})$ to represent how likely it is that the viewpoint has already been visited by any other UAV. The $l(\xi \in V_{\text{exp}})$ of any viewpoint is the maximum of a function describing the likelihood that the point lies in a given segment's bounding box, computed over all segments of all the other received LTVMaps. This likelihood function can be selected arbitrarily; for our approach, we selected a function, which is equal to 0 outside of the segment's bounding box, and grows linearly to 1 the closer it is to the center of the bounding box. The updates of these $l(\xi \in V_{\text{exp}})$ values for a three UAV mission can be seen in Figure 17.

For a frontier viewpoint ξ_L in the UAV's local map, which has $l(\xi_L \in V_{\text{exp}}) > 0$, the reward function changes into

$$R(\xi_{\text{UAV}}, \xi_L, \mathbb{M}) = l(\xi_L \in V_{\text{exp}}) R_R(\xi_{\text{UAV}}, \xi_L, \mathbb{M}) + (1 - l(\xi_L \in V_{\text{exp}})) R_L(\xi_{\text{UAV}}, \xi_L), \tag{11}$$

where R_L is the reward function defined by the employed single-UAV search strategy described in Section 8.2. This does not take into account any information from other UAVs. R_R is a reward function which takes into account other frontiers in received LTVMaps that could be reachable through ξ_L, as illustrated in Figure 16. If $l(\xi \in V_{\text{exp}}) = 0$, it means that the viewpoint does not fall into the space of any received LTVMap and the UAV only decides based on its own maps. If $l(\xi \in V_{\text{exp}}) = 1$, the viewpoint surely lies in explored space of another UAV, hence it does not

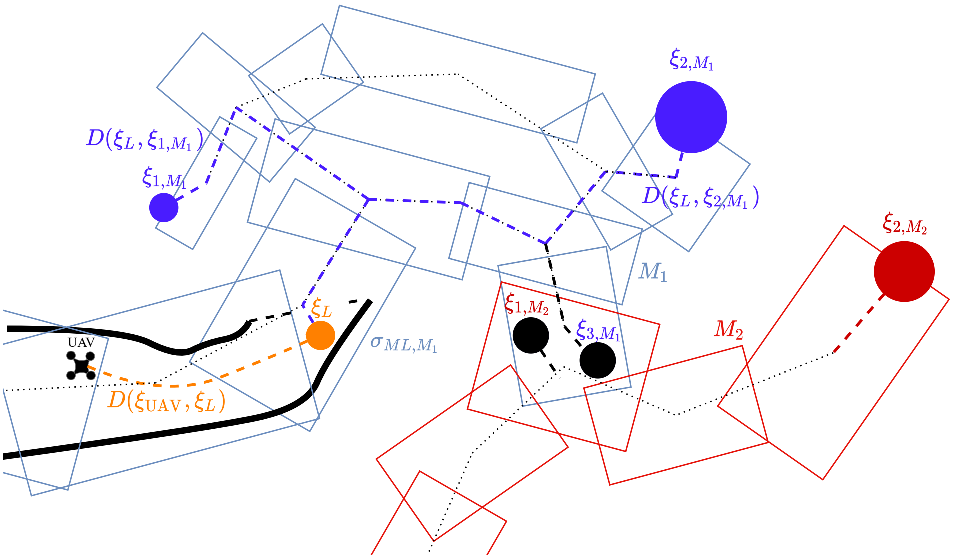


Figure 16. Diagram illustrating the computation of the cooperative exploration reward function, as described in (12). The image shows a UAV evaluating a frontier viewpoint ξ_L (orange) in its local occupancy map (black lines). The UAV has received two LTVMaps M_1, M_2 from two other UAVs. As the local map frontier ξ_L falls into one of the free space segments σ_{ML,M_1} of M_1, it is assigned as belonging to that segment and acts as an edge in planning paths between the local map and the received map M_1. Therefore, the frontier viewpoints $\xi_{1,M1}, \xi_{2,M2}$ should be reachable through ξ_L. A path to them is estimated across the centroids of the segments of M_1. The viewpoints $\xi_{3,M1}, \xi_{1,M2}$ (black) are marked as having $l(\xi \in V_{\text{exp}}) = 1$, since they fall deep into the explored space of the other received map, and are therefore not considered.

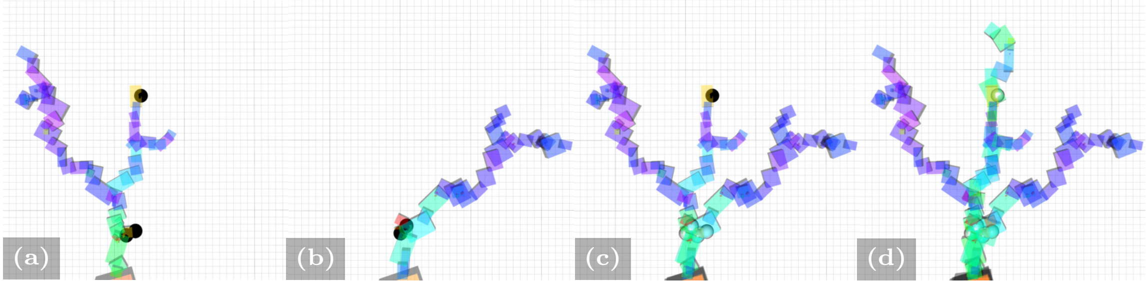


Figure 17. Illustration of LTVMap sharing and utilization during a cave exploration mission in simulation with three UAVs running the DEI strategy (described in Section 8.2.2). The heatmap color of the LTVMap segments shows surface coverage of the individual segments, with purple signifying complete coverage. The colors of the exploration viewpoints signify their $l(\xi \in V_{\mathrm{exp}})$ value, with white having a value equal to 1 and black being 0. Image (a) shows the LTVMap sent by UAV1 after returning to communication range with the base station. This map is given to UAV2, which then launches and chooses to explore the nearest unexplored frontier in the map of UAV1. Image (b) shows the LTVMap sent by UAV2 when it is returning. Image (c) then shows how the maps are co-updated onboard UAV3, which launches after receiving the LTVMap from UAV2. The only nonexplored viewpoint remaining is in the top part of the image. Image (d) shows the maps received by the base station from all three UAVs at the end of the mission with no unexplored viewpoints remaining.

bring any volumetric information to the team, so the UAV considers whether exploring it would eventually lead it to globally unexplored viewpoints. Figure 17 illustrates how sharing the LTVMap helps UAVs to not explore already explored parts of the environment. The function R_R that achieves this behavior was designed as

$$R_R(\xi_{\mathrm{UAV}}, \xi_L, \mathbb{M}) = \max_{M \in \mathbb{M}} \max_{\xi_R \in M} I(\xi_R) - D(\xi_{\mathrm{UAV}}, \xi_L) - \frac{D_R(\xi_L, \xi_R, \sigma_{ML,M})}{1 - l(\xi_R \in V_{\mathrm{exp}})}, \tag{12}$$

where $\mathbb{M}$ is the set of all received LTVMaps, and $\sigma_{ML,M}$ is the most likely segment that ξ_L belongs to in a map M. The function D_R is a special path cost function computed as a sum of Euclidean distances of segment centers in a given map, spanning from ξ_L, through the center of $\sigma_{ML,M}$, and towards a given frontier viewpoint ξ_R. The value of D_R is also scaled by a user-defined parameter. This is done so as to increase the cost of viewpoints in received maps as there is more uncertainty about the path to these viewpoints. The division by $1 - l(\xi_R \in V_{\mathrm{exp}})$ serves to gradually decrease the reward of exploring the viewpoint up to $-\infty$ when the viewpoint was surely explored by another UAV. Computation of this reward function is illustrated in Figure 16.

The percentage of covered surfaces inside segments received in the LTVMap is used for blocking the surface coverage viewpoints in segments, where the percentage is above a user-defined threshold. The segments with low surface coverage could be used as additional goals in a similar manner as shared frontiers in Figure 16. However, for simplicity, this was not implemented.

8.4. Autonomy robustness enhancements

One important problem is that in the case of dark and nonreflective surfaces (common in the DARPA SubT Finals course) the LiDAR beam does not return with enough energy. Such surfaces will not be marked as occupied and essentially become permanent frontiers, which means that some informative viewpoints, as defined in Section 8.2, are noninformative. To solve this, the UAV builds a map of visited positions. With time spent near a visited position, we linearly decrease the value of nearby viewpoints. After some time, the sampling is blocked near those positions completely.

Another problem arising is due to highly dynamic obstacles in the occupancy map, such as other robots, fog, or very narrow corridors where the discretization of occupancy can oscillate. As such, the reachability of a given viewpoint can oscillate. This was solved by putting a timeout on trying to reach a given viewpoint and was triggered if the UAV did not get closer to the goal within a defined time. After this timeout, an area around the viewpoint is blocked until the end of the mission, or until a manual reset by the operator. This approach may cause the UAV to block some goals that are only temporarily blocked by another robot in narrow passages, but it was deemed preferable rather than having the UAV permanently oscillate in such passages.

The autonomy system can be easily controlled by operator commands (described in Section 11.2) which can block viewpoints in a set cylinder in space, force the UAV to explore towards some goal, or simply move to a given position and stay there. In this way, problematic situations not covered by our solution, such as organizing multiple robots in a tight corridor, can be resolved by the operator.

9. Path planning, trajectory generation and tracking

Planning collision-free paths and generating dynamically feasible trajectories is another vital component of the presented UAV system operating in a constrained environment. The sequence of waypoints (a waypoint in this context is either only a point in space, when we do not care about the heading, or a point in space and heading, for example when using the VPE strategy in Section 8.2.3, that the local planner should move the UAV through) that efficiently guides the UAV through the environment is produced by the long-distance navigation module, described in Section 9.1. Given the navigation waypoints, a computationally undemanding multistage approach is applied to obtain a trajectory lying at a safe distance from obstacles, while also respecting dynamic constraints (limits on velocity, acceleration, and jerk) and minimizing the time of trajectory following. In particular, the solution can be divided into three modules: path planning to obtain the local reference path, path processing to increase the safety margin of the path, and the trajectory generation to obtain a time-parametrized trajectory respecting the dynamic constraints of the UAV. The diagram illustrating connections and data transfer between particular modules in path planning and trajectory generation pipeline is shown in Figure 18. The long-distance path found in SphereMap, the local path found in DenseMap, the postprocessed path, and the dynamic trajectory are depicted in Figure 19.

9.1. Safety-aware long-distance navigation

When a goal, or a sequence of goals, is set to the navigation stack, the long-distance navigation module computes a path through the SphereMap, optimal according to (7). The module then keeps this path and utilizes the trajectory planning and tracking modules to follow it. This is done simply by a "carrot and stick" approach, where the trajectory planning module is given a near waypoint

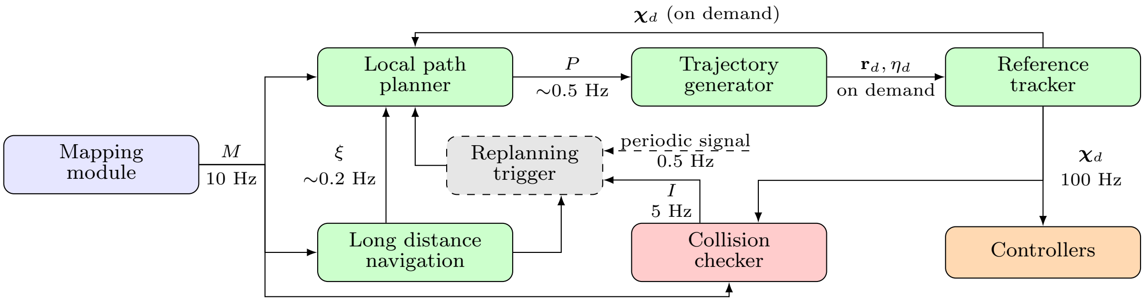


Figure 18. A diagram of the path planning and trajectory generation pipeline. Given 3D map M provided by *Mapping module*, the *Local path planner* produces path P connecting a start position for planning derived from full state reference χ_d provided by *Reference tracker*, with the viewpoints ξ supplied by *Long distance navigation* module. *Trajectory generator* produces feasible trajectory along the collision-free path P and supplies the position and heading reference $(\mathbf{r}_d, \eta_d)$ to a *Reference tracker*. *Reference tracker* creates a smooth and feasible reference for the reference feedback controllers. The *Local path planner* is triggered by a new set of goals, periodic signal or by an interrupt I generated by *Collision checker* responsible for detection of collisions with respect to most recent map of the environment.

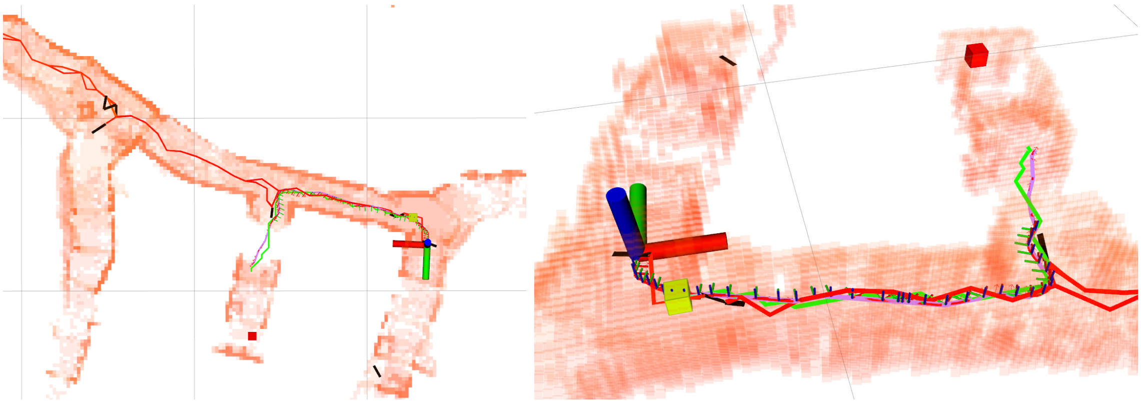


Figure 19. A two-view visualization of the path produced by all stages of the planning pipeline. The cached long-distance paths (——) between portals (——) are found in the SphereMap. A geometric path (——) is found in the DenseMap to the next waypoint given by the SphereMap. This path is then postprocessed (——) to be further away from obstacles, and a time-parametrized trajectory respecting the dynamics of the UAV is sampled (small axes). The small axes represent samples from the trajectory with constant time step, so axes further away from each other mean that the velocity of that part is higher. The current UAV pose is shown as large axes.

(approx. 20 m away from the UAV at maximum, to keep planning time short) on the path. This temporary goal waypoint is then slid across the path towards the goal.

If the trajectory planning and tracking modules cannot advance along the SphereMap path for a specified amount of time, which can be caused by a dynamic obstacle such as a rockfall, fog, or another robot, the SphereMap path following is stopped and an unreachability flag is raised. The UAV then chooses a different goal or tries to find a new path to the same goal based on the current state of mission control.

When the search planning requires the UAV to fly through multiple nearby goal viewpoints, such as when covering the surfaces in a room with cameras or when visiting multiple viewpoints while traveling and using the VPE strategy described in Section 8.2.3, the local path planning module is instead given a sequence of waypoints (containing both the goal viewpoints for surface coverage, which require heading alignment, and waypoints that do not require heading alignment and only serve to guide the local path planning). Thus the output of this module is always a sequence of one or more waypoints, which may or may not require heading alignment, and through which the local path planning module should find a path in a short time, which we can control by changing the look-ahead distance.

9.2. Local path planning

The grid-based path planner coupled with iterative path processing was adopted from (Kratky et al., 2021a) to obtain the primary reference path. The proposed approach presents a path planning and processing algorithm, which is based on the traditional A* algorithm applied on a voxel grid with several modifications to decrease the computational demands. The first modification lies in avoiding the computationally demanding preprocessing of the map representation (e.g., obstacle dilation by Euclidean distance field), which often requires more time than the actual planning on the grid. This holds true especially for shorter direct paths that leave a significant portion of the previously processed environment unexploited. For this reason, the presented approach builds a k-d tree representation of the environment which is then used to conclude the feasibility of particular cells, based on their distance to the nearest obstacle. As a result, the computational demands are partially moved from the preprocessing phase to the actual planning phase. This approach is particularly efficient in the case of paths that do not require exploiting a significant part of the environment. The second important modification is applying node pruning, similar to the jump point search algorithm (Harabor and Grastien, 2011). This modification helps to decrease the number of unnecessarily expanded nodes. As such, it lowers the computational time required for obtaining the solution. A detailed analysis of the influence of particular modifications on the performance of the planning algorithm is provided in (Kratky et al., 2021a).

To allow the generated paths to lead through narrow passages, the limits on safety distance are set to the dimension of the narrowest opening that is supposed to be safely traversable by the UAV. However, setting this distance to a value that ensures safety in the event of the maximum possible deviation from the path caused by any external or internal source would lead to the preclusion of entering narrow passages of the environment. On the contrary, setting this distance to a minimum value without considering safety margins would increase the probability of collision along the whole path. To balance the traversability and safety of the generated path, the minimum required UAV-obstacle distance applied in the planning process is set to the lowest traversability limit, and iterative path postprocessing is applied to increase the UAV-obstacle distance in wider parts of the environment. The employed postprocessing algorithm proposed in (Kratky et al., 2021a) iteratively shifts the path towards the free part of the environment, while continually maintaining the path's connectivity. As such, this anytime algorithm increases the average UAV-obstacle distance throughout the flight, which significantly improves the reliability of the navigation with respect to imprecisions in the reference trajectory tracking.

The generated path is periodically replanned at a rate of 0.5 Hz to exploit the newly explored areas of the environment and handle dynamic obstacles. The continuous path following is achieved by using the predicted reference generated by the MPC tracker (Baca et al., 2018) to identify the starting position for the planner at time T_s in the future. Apart from the periodic replanning, the planning is also triggered by the detection of a potential collision on the prediction horizon of the trajectory reference produced by the MPC tracker. The potential collisions are checked at a rate of 5 Hz by comparing the distance of particular transition points of the predicted trajectory to the nearest obstacle in the most recent map of the environment. Depending on the time left to the time instant of a potential collision, the UAV is either requested to perform a stopping maneuver or to trigger replanning with the most up-to-date map.

9.3. Trajectory generation

The path generated by the path planning pipeline is a series of waypoints, each consisting of a 3D position and heading. A trajectory (a series of dense time-parameterized waypoints) is generated for each new path, so that the motion of the UAV satisfies translational dynamics and dynamic constraints up to the 4th derivative of position. The dynamics of the trajectory can be changed according to the current safety distance limit. However, in the Final Event, this feature was disabled, as the UAV was already constrained to $1\,\mathrm{m\,s^{-1}}$ and further slowdown would unnecessarily prolong the time spent in a narrow passage, where the risk of collision is higher. The trajectory generation system is based on the polynomial trajectory generation approach (Richter et al., 2016; Burri et al., 2015), but it was significantly extended to perform in a constrained, real-world environment (Baca et al., 2021). This approach was modified to minimize the total flight time while still satisfying the dynamic constraints. Furthermore, an iterative sub-sectioning algorithm was added to force the resulting trajectory into a feasible corridor along the original path. Moreover, a fallback solver was added to cope with invalid QP solver results caused by numerical instabilities or in case of the solver timeout. The QP solver sometimes fails to produce a feasible trajectory, e.g., by violating the dynamic constraints, or by violating the corridor constraints. In such cases, we find an alternative solution by linearly sampling each original path segment. The time duration for each segment is estimated heuristically as an upper bound using the same method as in the initialization of the polynomial trajectory generation (Baca et al., 2021). Although the trajectory produced by this method violates the dynamics in each waypoint, the underlying MPC Trajectory tracking mechanism provides smooth control reference even at these points. Most importantly, despite the fallback solution not being optimal, it is tractable and is guaranteed to finish within a fraction of the

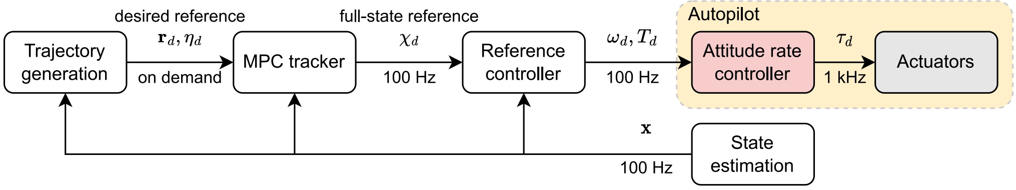


Figure 20. A diagram of the system architecture (Baca et al., 2021): *Trajectory generation* supplies the position and heading reference ($\mathbf{r}_d$, η_d) to the *MPC tracker*. The *MPC tracker* creates a smooth and feasible reference for the *Reference controller*. The *Reference controller* produces the desired angular velocities and thrust (ω_d, T_d) for the embedded *Attitude rate controller*, which sets the desired speed of the motors τ_d.

time of the polynomial optimization. Finally, a dynamic initialization mechanism and a time-outing system were added to cope with the nonzero trajectory generation and path planning computation times. Even though the path planning and the trajectory generation can last for several hundreds of milliseconds, the resulting trajectory always smoothly connects to the currently tracked trajectory. Therefore, no undesired motion of the UAV is produced. The updated trajectory generation approach was released and is maintained as part of the MRS UAV System (Baca et al., 2021).

9.4. Trajectory tracking and feedback control

The low-level guidance of the UAV is provided by a universal UAV control system, as developed by the authors of (Baca et al., 2021). The onboard control system supports modular execution of UAV reference generators, feedback controllers, and state estimators. During the SubT Finals, the system exclusively utilized the geometric tracking control on *SE(3)* (Lee et al., 2010) to follow the desired states generated by the MPC Tracker (Baca et al., 2018). First, the MPC Tracker is supplied with a time-parametrized reference trajectory, from which a smooth and feasible reference state consisting of position, velocity, acceleration, jerk, heading, and heading rate is generated by controlling a virtual model of the UAV. Second, the feedback controller minimizes the control error around the generated reference state and produces an attitude rate reference for the low-level attitude rate controller embedded in the Flight Control Unit (FCU). Figure 20 depicts the pipeline diagram of the control system with data flow among individual modules.

10. Artifact detection, localization, and reporting

Objects of interest (artifacts) in the explored area are detected visually using a CNN that processes images from several onboard RGB cameras covering the frontal, top, and bottom sectors of the UAV. The CNN detector is trained on our manually labeled dataset and outputs predicted bounding boxes and corresponding classes of the artifacts in the input images. To estimate the 3D positions of the detections, we have leveraged the onboard 3D LiDAR sensor and the mapping algorithm described in Section 7. These positions are processed by an artifact localization filter based on our previous work (Vrba et al., 2019), which fuses the information over time to filter out sporadic false positives and improve the localization precision. The artifact detection, localization, and filtering pipeline is illustrated in Figure 21.

Our approach consolidated into a similar principle of early recall and late precision proposed by (Lei et al., 2022). The CNN generates a high amount of detections to not miss any artifact at the cost of a high false positive rate. The false positives are later filtered out by the localization filter and the resulting hypotheses are further pruned by the human operator to improve the precision of the pipeline as a whole.

10.1. Artifact detection

The artifact detection is executed in parallel on image streams from all cameras at the same time, which would require a dedicated Graphical Processing Unit (GPU) onboard the UAV. Therefore,

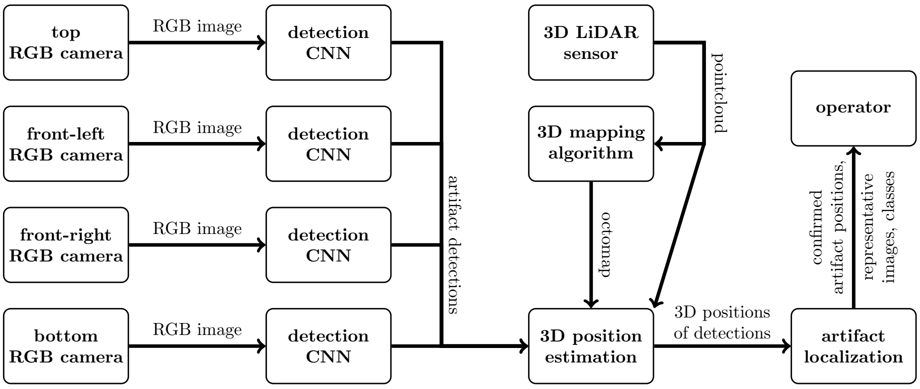


Figure 21. Schematic of the artifact detection and localization pipeline.

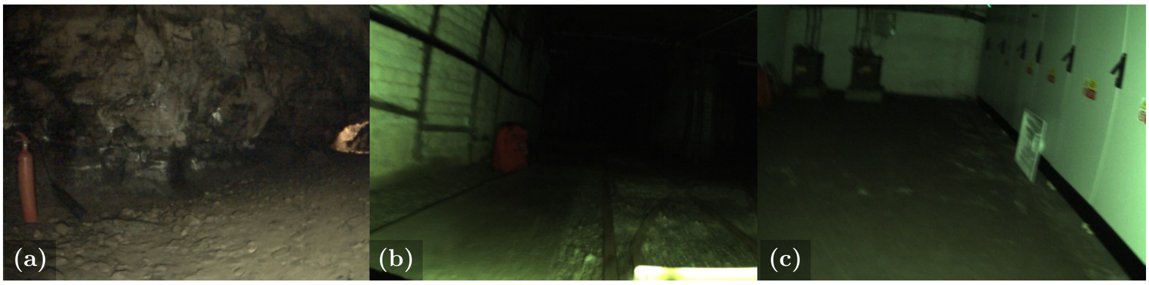


Figure 22. Training images containing artifacts captured by the onboard cameras in cave (a), tunnel (b), and urban (c) environments.

we have chosen the lightweight MobileNetV2 CNN (Sandler et al., 2018), in order to achieve a high detection rate and keep the load on the onboard computer as low as possible.

The CNN is running on the Intel UHD GPU that is integrated within the onboard CPU of the UAV. The integrated Intel GPU interfaces with our pipeline using the OpenVino[5] framework. The OpenVino framework together with the Intel GPU achieves more than 5 Hz detection rate on 4 cameras in parallel but due to fixed resource allocation, we are locking the camera rates to 5 Hz. This artificial throttling of the detection rate avoids issues when the integrated GPU locks the memory resources for the CPU, which might lead to lag in the control pipeline.

The MobileNetV2 base model is modified for training using the OpenVino open-source tools. The evaluation of the model is based on the mean average precision metric (mAP) and recall. The mAP metric is a standard metric for object detection models since it provides information about how accurate the prediction is. Recall provides an understanding what is the ratio between true positive predictions and the total number of positive samples in the dataset.

The main challenge for the model is to adapt to different domains—mine, urban, and cave environments have different lighting and backgrounds (see Figure 22), which affect the detection performance. Moreover, the angle from which the images were taken is different as part of the images in the dataset were taken by ground vehicles and the rest by UAVs.

As the whole dataset was initially not available, we had to train the model incrementally whenever we gathered data from a new type of environment or camera angle to ensure we represented all cases uniformly in the training data. The incremental training was more time-efficient than retraining on the whole dataset each time new data was added. Training from scratch or checkpoints took us 2–3 days using our GPU capabilities, while incremental training produced good results in only 4–6 hours

[5] docs.openvino.ai/latest/index.html

of training. Having the possibility to relatively quickly update the model was critical when we were doing practical experiments or hot-fixing some false-positive detections during competition days or our experimental campaign.

For training the model on a growing dataset, we used a variety of learning schedulers from the MMdetection toolbox (Chen et al., 2019). The Cosine scheduler designed by (Loshchilov and Hutter, 2016) is used for warm-restarts of the training pipeline to overcome the loss of learned features. The main challenge of transfer learning is to overcome the loss of learned distribution on the previous dataset when training the model on the new dataset (in this case the new dataset is a combination of the previous dataset and newly collected data).

In our experience, different learning rate schedulers should be used depending on the size of newly added data:

- *Cosine scheduler* (Loshchilov and Hutter, 2016) is used during clean model training on the initial dataset.
- *Cyclic scheduler* (Smith, 2015) is used when the size of new data is more than 15 % of the size of the initial dataset.
- *Step decay scheduler* is used when less than 15 % of the initial dataset size is added.

The model was trained using NVIDIA GeForce RTX 3090 video card with 24 GB of RAM, with 64 images per batch. The training size initially contained around 13 000 images and incrementally increased to 37 820 as new backgrounds and false negative samples were gradually added. Out of the 37 820 images 31 000 were labeled artifacts and 6820 were background images without any artifact to reduce the false positive rate. The train and validation split was 70 % to 30 % per training size. We open-sourced our training pipeline to facilitate replicating the achieved results by the community: `github.com/ctu-mrs/darpa_subt_cnn_training`. This method resulted in a score of 49.1 % mAP on the whole dataset. Such a value is acceptable on the onboard computation unit with limited resources, due to which a trade-off between accuracy and detection was necessary.

The dataset was collected using the off-the-shelf objects that were specified by the organizers, see Figure 1. The data have been recorded from the onboard cameras on the UAVs and UGVs, in particular:

- Intel RealSense D435,
- Basler Dart daA1600,
- Bluefox MLC200w.

The Basler cameras do not have an IR filter installed to maximize the amount of captured captured light. Altogether the dataset has 37820 images, sometimes with multiple objects in one frame. An example of images from the dataset is shown in Figure 22.

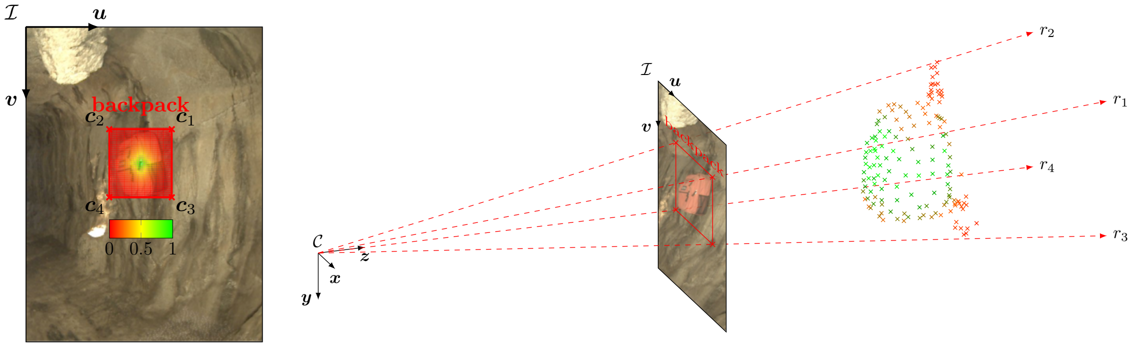
We publish the labeled detection datasets that were used for training of the neural network at `github.com/ctu-mrs/vision_datasets`. In addition, we also publish the tools to convert it into PASCAL VOC or COCO formats for immediate usage on most of the open-source models.

10.2. Estimation of 3D position of detections

Positions of the detected objects are estimated using data from the onboard LiDAR sensor and the mapping algorithm. Each detection is represented by four corner points $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4$ of its bounding rectangle in the image plane of the corresponding camera, as estimated by the detector (see Figure 23a). These points are expressed as undistorted pixel coordinates in the image frame $\mathcal{I}$.

The mathematical projection model of the camera $f_{\text{proj}} : \mathbb{R}^3 \to \mathbb{R}^2$ is assumed to be known. In our case, we have used the standard pinhole camera model formulated as

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{13}$$



(a) Example of an artifact detection with an overlay visualization of the sample weighting function f_w.

(b) Model of the camera and the point cloud-based sampling method. Rays r_1, r_2, r_3, r_4 are projections of c_1, c_2, c_3, c_4, respectively. Only the points within the area defined by these rays are selected. The selected points are colored based on their weight. Nonselected points are not drawn for clarity.

Figure 23. Illustration of the point sampling for 3D position estimation of detected artifacts with an example detection of a backpack.

where f_u, f_v, u_0, v_0 are parameters of the model (focal length and image center), $\left[x, y, z\right]^\intercal$ is a 3D point in the camera coordinate frame $\mathcal{C}$, and u, v are distortion-free pixel coordinates in the image frame $\mathcal{I}$, corresponding to the 3D point (see Figure 23b for illustration). To model the distortion of the real-world camera, we have used a standard radial-tangential polynomial distortion model. It is worth noting that the output of f_proj^{-1} is a 3D ray and not a single point, which is represented in the model by the free scaling factor $k \in \mathbb{R}$.

The input LiDAR scan is represented as a set of 3D points $\mathcal{S} = \{\mathbf{p}_i\}$ expressed in the camera coordinate frame $\mathcal{C}$. The occupancy map is represented using the DenseMap data structure that is described in Section 7, and which provides a raycasting function $f_\mathrm{raycast} : \mathcal{R} \to \mathbb{R}^3$ where $\mathcal{R}$ is the set of all 3D rays. The function f_raycast returns the point, corresponding to the first intersection of the specified ray with an obstacle in the environment (or nothing if there is no such intersection).

The position of each detected object is estimated from a number of points that are sampled using two methods: a primary one that utilizes the latest available point cloud from the LiDAR and a secondary backup method using the latest DenseMap estimated by the mapping algorithm. The primary method is more accurate and less computationally intensive, but for artifacts lying outside of the FOV of the LiDAR scan, it may not provide enough samples for accurate 3D position estimation, which is when the secondary method is employed. For each sampled point $\mathbf{s}_i \in \mathcal{S}$, its weight w_i is calculated. The position estimate $\mathbf{d}$ and its corresponding uncertainty covariance matrix $\mathbf{Q}_\mathbf{d}$ are obtained as a weighted mean of the sampled points:

$$\mathbf{d} = \sum_{i=1}^{|\mathcal{S}|} \mathbf{s}_i w_i, \qquad \mathbf{Q}_\mathbf{d} = \frac{1}{1 - \sum_{i=1}^{|\mathcal{S}|} w_i^2} \sum_{i=1}^{|\mathcal{S}|} w_i \left(\mathbf{s}_i - \mathbf{d}\right) \left(\mathbf{s}_i - \mathbf{d}\right)^\intercal, \tag{14}$$

where $\mathcal{S}$ is the set of sampled points and the weights w_i are normalized so that $\sum_{i=1}^{\mathcal{S}} w_i = 1$.

The weight of a point $\mathbf{s}$ is obtained based on the distance of its reprojection to the image coordinates $\mathbf{s}' = \left[s_u, s_v\right]^\intercal = f_\mathrm{proj}\left(\mathbf{s}\right)$ from the center of the detection's bounding box $\mathbf{c}_0 = \left[c_u, c_v\right]^\intercal$ using the function

$$f_\mathrm{w}\left(\mathbf{s}', \mathbf{c}_0\right) = \left(1 - \frac{2\left|s_u - c_u\right|}{w_\mathrm{bb}}\right)^2 \left(1 - \frac{2\left|s_v - c_v\right|}{h_\mathrm{bb}}\right)^2, \tag{15}$$

where $w_\mathrm{bb}, h_\mathrm{bb}$ are the width and height of the bounding box, respectively. The weighting function serves to suppress points further from the center of the bounding box. This is based on our empirical observation that the center provides the most reliable estimate of the detected object's position,

Algorithm 2. Algorithm for the estimation of a detection's position and covariance.

```
Input:
    $\mathcal{D} = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}, \ \mathbf{c}_i \in \mathbb{R}^2$    ▷ undistorted coordinates of the detection's bounding box
    $f_{\text{proj}} : \mathbb{R}^2 \to \mathcal{R}$    ▷ the projection model of the camera
    $\mathcal{P} = \left\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{|\mathcal{P}|}\right\}, \ \mathbf{p}_i \in \mathbb{R}^3$    ▷ the latest point cloud from the LiDAR
    $f_{\text{raycast}} : \mathcal{R} \to \mathbb{R}^3$    ▷ the raycasting function of the occupancy map
    $n_{\text{desired}} \in \mathbb{N}$    ▷ the desired number of sampled points
Output:
    $\mathbf{d} \in \mathbb{R}^3$    ▷ estimated position of the detection
    $\mathbf{Q}_{\mathbf{d}} \in \mathbb{R}^{3\times 3}$    ▷ covariance matrix of the position estimate
Begin:
    ▷ First, the desired number of points is sampled using the primary and secondary methods.
    $r_1 := f_{\text{proj}}^{-1}(\mathbf{c}_1), \ r_2 := f_{\text{proj}}^{-1}(\mathbf{c}_2), \ r_3 := f_{\text{proj}}^{-1}(\mathbf{c}_3), \ r_4 := f_{\text{proj}}^{-1}(\mathbf{c}_4)$    ▷ project the corners of the bounding box to 3D rays
    $\mathcal{S}_1 := \{\mathbf{p}_i \in \mathcal{P} \mid \mathbf{p}_i$ within the area defined by edges $r_1, r_2, r_3, r_4\}$ ▷ try to sample $n_{\text{desired}}$ points using the primary method
    $n_{\text{remaining}} := \max(n_{\text{desired}} - |\mathcal{S}_1|, 0)$    ▷ calculate the remaining number of points to be sampled
    $\mathcal{S}_2 :=$ sampleRectangle$(\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}, n_{\text{remaining}}, f_{\text{proj}}, f_{\text{raycast}})$    ▷ sample any remaining points from the occupancy map
    $\mathcal{S} := \mathcal{S}_1 \cup \mathcal{S}_2$    ▷ complement $\mathcal{S}_1$ with the remaining points from $\mathcal{S}_2$
    ▷ Then, the weight of each sampled point is calculated using the weighting function $f_{\text{w}}$.
    $\mathbf{c}_0 :=$ mean$(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$    ▷ calculate the center of the bounding box
    for each $\mathbf{s}_i \in \mathcal{S}$ do
        $\mathbf{s}'_i := f_{\text{proj}}(\mathbf{s}_i)$    ▷ project the point back to the image frame $\mathcal{I}$
        $w_i := f_{\text{w}}(\mathbf{s}'_i, \mathbf{c}_0)$    ▷ calculate its weight
    ▷ Finally, the position and its uncertainty are calculated as a weighted mean and covariance and returned.
    $\mathbf{d} := \sum_{i=1}^{|\mathcal{S}|} \mathbf{s}_i w_i$
    $\mathbf{Q}_{\mathbf{d}} = \frac{1}{1 - \sum_{i=1}^{|\mathcal{S}|} w_i^2} \sum_{i=1}^{|\mathcal{S}|} w_i (\mathbf{s}_i - \mathbf{d})(\mathbf{s}_i - \mathbf{d})^{\intercal}$
    return $\mathbf{d}, \ \mathbf{Q}_{\mathbf{d}}$
```

while the bounding box's corners typically correspond to the background and not the object, as illustrated in Figure 23a. The whole 3D position estimation algorithm is presented in Algorithm 2. The `sampleRectangle` routine used in Algorithm 2 is described in Algorithm 3.

The estimated positions and the corresponding covariance matrices serve as an input to the *artifact localization filter* described in the next section (refer to Figure 21). To avoid bias and numerical singularities in the filter, some special cases of the covariance calculation have to be handled. Namely, these are the following.

1. *All extracted points lie on a plane.* This happens, e.g. when all the cast rays of the secondary position estimation method intersect the same voxel of the DenseMap. The covariance matrix is then singular, which causes numerical problems with integrating the measurement.
2. *All extracted points are too close to each other.* This typically happens when the detected object is too far or too small. The covariance matrix's eigenvalues are then too small, biasing the fused position estimate of the artifact.

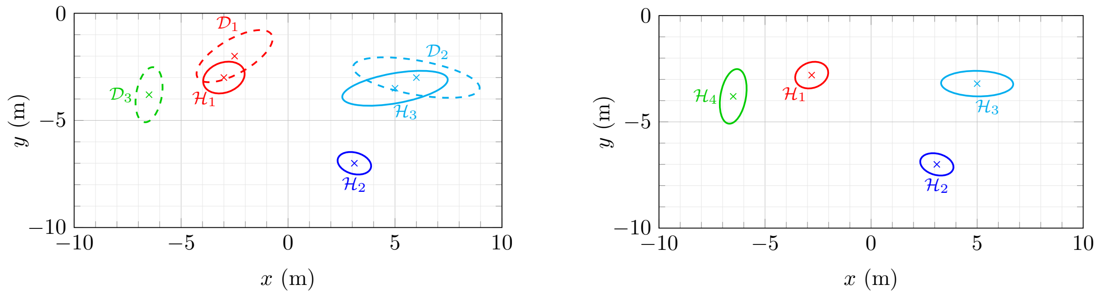
To avoid these problems, the estimated covariance matrix is rescaled, so that all eigenvalues conform to a specified minimal threshold before being processed by the artifact localization filter.

10.3. Artifact localization filter

Artifact detections are filtered using an approach based on our previous work, where a multi-target tracking algorithm was employed for detection, localization, and tracking of micro aerial vehicles (Vrba et al., 2019). The filtering serves to improve the precision of the artifacts' estimated positions and to reject false positives. Only artifacts that are consistently detected multiple times with sufficient confidence are confirmed, and only the confirmed artifacts are then reported to the operator to save the limited communication bandwidth. A single step of the algorithm is illustrated in Figure 24.

Algorithm 3. The `sampleRectangle` routine for sampling a number of 3D points from the occupancy map.

```
▷ This routine samples points within a rectangle in the image plane $\mathcal{I}$ by raycasting pixels on inscribed ellipses with an increasing radius.
Routine sampleRectangle:
    Input:
        $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$, $\mathbf{c}_i \in \mathbb{R}^2$      ▷ corners of the rectangle to be sampled in the image frame $\mathcal{I}$
        $n_{\text{remaining}} \in \mathbb{N}$      ▷ the desired number of samples
        $f_{\text{proj}} : \mathbb{R}^2 \to \mathcal{R}$      ▷ the projection model of the camera
        $f_{\text{raycast}} : \mathcal{R} \to \mathbb{R}^3$      ▷ the raycasting function of the occupancy map
    Output:
        $\mathcal{S} = \{\mathbf{s}_i\}$      ▷ a set of sampled points in the image frame $\mathcal{I}$ such that $|\mathcal{S}| \leq n_{\text{remaining}}$
    Parameters:
        $n_r \in \mathbb{N}$, $n_\alpha \in \mathbb{N}$      ▷ number of radial sampling steps and number of circumferential steps per unit circumference
    Begin:
        $w := c_{1,u} - c_{3,u}$,  $h := c_{1,v} - c_{3,v}$      ▷ calculate the width and height of the rectangle
        $r_{\text{step}} := 1/n_r$
        for $r \in \{0, r_{\text{step}}, 2r_{\text{step}}, \dots, 1\}$ do
            $\alpha_{\text{step}} := r/n_\alpha$
            ${}_\Delta\alpha := u$,  $u \sim \mathcal{U}(-\pi, \pi)$      ▷ generate a random angular offset to avoid biasing some directions
            for $\alpha \in \{0, \alpha_{\text{step}}, 2\alpha_{\text{step}}, \dots, 2\pi\}$ do
                $\mathbf{s}' := [wr\cos(\alpha + {}_\Delta\alpha)/2,\ hr\sin(\alpha + {}_\Delta\alpha)/2]^\intercal$      ▷ calculate a sample point on an ellipse
                $r := f_{\text{proj}}(\mathbf{s}')$      ▷ project the point to a 3D ray
                $\mathcal{S} := \mathcal{S} \cup f_{\text{raycast}}(r)$      ▷ find an intersection of the ray with an obstacle and add it to $\mathcal{S}$
                if $|\mathcal{S}| = n_{\text{remaining}}$ then
                    return $\mathcal{S}$
        return $\mathcal{S}$
```



(a) Situation before the update step. The detections $\mathcal{D}_1$ and $\mathcal{D}_2$ are associated to the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_3$, respectively. The detection $\mathcal{D}_3$ is not associated to any hypothesis. The hypothesis $\mathcal{H}_2$ has no detection associated.

(b) Situation after the update step. The detections $\mathcal{D}_1$ and $\mathcal{D}_2$ updated the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_3$, respectively. The detection $\mathcal{D}_3$ initialized a new hypothesis $\mathcal{H}_4$ and the hypothesis $\mathcal{H}_2$ remained unchanged.

Figure 24. Illustration of one step of the artifact localization filter (a top-down view). Hypotheses $\mathcal{H}_i$ are shown as covariance ellipsoids with the mean $\hat{\mathbf{x}}_i$ marked by an "×" symbol. Detections $\mathcal{D}_i$ are represented in the same way using dashed lines. Associations between hypotheses and detections are highlighted using color.

The filter keeps a set of hypotheses about objects in the environment. Each hypothesis $\mathcal{H}$ is represented by an estimate of the object's position $\hat{\mathbf{x}}$, its corresponding covariance matrix $\mathbf{P}$, and a probability distribution of the object's class $p_{\mathcal{H}} : \mathcal{C} \to [0, 1]$, where $\mathcal{C}$ is the set of considered classes. For every hypothesis $\mathcal{H}$, up to one detection $\mathcal{D}_{\mathcal{H}}$ is associated according to the rule

$$\mathcal{D}_{\mathcal{H}} = \begin{cases} \operatorname{argmax}_{\mathcal{D}} l(\mathcal{D} \mid \mathcal{H}), & \text{if } \max_{\mathcal{D}} l(\mathcal{D} \mid \mathcal{H}) > l_{\text{thr}}, \\ \emptyset, & \text{else}, \end{cases} \tag{16}$$

where $l\left(\mathcal{D} \mid \mathcal{H}\right)$ is the likelihood of observing $\mathcal{D}$ given that it corresponds to $\mathcal{H}$, and l_{thr} is a likelihood threshold. The associated detections are used to update the corresponding hypotheses. The detections that are not associated initialize new hypotheses.

The position estimate $\hat{\mathbf{x}}$ of a hypothesis $\mathcal{H}$ and its covariance $\mathbf{P}$ are updated using the Kalman filter's update equation and an associated detection $\mathcal{D}_{\mathcal{H}}$ at time step t as

$$\mathbf{K}_{[t]} = \mathbf{P}_{[t]}\mathbf{H}^{\intercal}\left(\mathbf{H}\mathbf{P}_{[t]}\mathbf{H}^{\intercal} + \mathbf{Q}_{\mathbf{d}[t]}\right)^{-1}, \tag{17}$$

$$\hat{\mathbf{x}}_{[t+1]} = \hat{\mathbf{x}}_{[t]} + \mathbf{K}_{[t]}\left(\mathbf{d}_{[t]} - \mathbf{H}\hat{\mathbf{x}}_{[t]}\right), \tag{18}$$

$$\mathbf{P}_{[t+1]} = \left(\mathbf{I} - \mathbf{K}_{[t]}\mathbf{H}\right)\mathbf{P}_{[t]}, \tag{19}$$

where $\mathbf{K}_{[t]}$ is a Kalman gain, $\mathbf{I}$ is an identity matrix, $\mathbf{H}$ is an observation matrix (in our case, equal to $\mathbf{I}$), $\mathbf{d}_{[t]}$ and $\mathbf{Q}_{\mathbf{d}[t]}$ are the estimated position of $\mathcal{D}_{\mathcal{H}[t]}$ and its corresponding covariance matrix, respectively. The class probability distribution $p_{\mathcal{H}}$ is updated as

$$p_{\mathcal{H}[t+1]}\left(c\right) = \frac{n_{\text{dets}[t]}p_{\mathcal{H}[t]}(c) + p_{\mathcal{D}_{\mathcal{H}}[t]}(c)}{n_{\text{dets}[t]} + 1}, \tag{20}$$

where $c \in \mathcal{C}$ is an object's class and $n_{\text{dets}[t]}$ is the number of detections, associated to $\mathcal{H}$ thus far.

Because the artifacts are assumed to be immobile, the Kalman filter's prediction step is not performed, which has the effect that the uncertainty of a hypothesis (represented by $\mathbf{P}$) can decrease without bounds. This can cause the likelihood $l\left(\mathcal{D} \mid \mathcal{H}\right)$ of new measurements corresponding to the same object to be below the association threshold, breaking the association algorithm. To avoid this, the covariance matrix $\mathbf{P}$ is rescaled after each update so that its eigenvalues are larger than a specified minimal value, which enforces a lower bound on the position uncertainty of the hypotheses.

10.3.1. Association likelihood

To calculate the likelihood $l\left(\mathcal{D}_{[t]} \mid \mathcal{H}_{[t]}\right)$ of observing a detection $\mathcal{D} \equiv \{\mathbf{d},\ \mathbf{Q}_{\mathbf{d}}\}$ given that it corresponds to a hypothesis $\mathcal{H} = \{\hat{\mathbf{x}},\ \mathbf{P}\}$ at time step t, we use a measurement model

$$\mathbf{d}_{[t]} = \mathbf{H}\mathbf{x} + \xi_{[t]}, \qquad \xi_{[t]} \sim \mathcal{N}\left(\mathbf{0},\ \mathbf{Q}_{\mathbf{d}[t]}\right), \tag{21}$$

where $\mathbf{H}$ is the observation matrix, $\mathbf{x}$ is a hidden state (the real position of the artifact), $\xi_{[t]}$ is measurement noise, and $\mathcal{N}\left(\mathbf{0},\ \mathbf{Q}_{\mathbf{d}[t]}\right)$ denotes the Gaussian probability distribution with zero mean and covariance matrix $\mathbf{Q}_{\mathbf{d}[t]}$. Using this model, the probability density function of the expected measurement given $\mathbf{x}$ is

$$p\left(\mathbf{d}_{[t]} \mid \mathbf{x}\right) = f\left(\mathbf{d}_{[t]} \mid \mathbf{H}\mathbf{x}, \mathbf{Q}_{\mathbf{d}[t]}\right), \tag{22}$$

where $f\left(\,\cdot \mid \mu, \mathbf{\Sigma}\right)$ denotes the density function of the Gaussian distribution with mean μ and covariance matrix $\mathbf{\Sigma}$.

The Kalman filter described by equations (17) to (19) can be interpreted as an estimator of the probability density of the hidden state given previous measurements. This probability density is represented as a random variable with a Gaussian distribution:

$$p\left(\mathbf{x} \mid \mathbf{d}_{[1]}, \ldots, \mathbf{d}_{[t]}\right) = f\left(\mathbf{x} \mid \hat{\mathbf{x}}_{[t]}, \mathbf{P}_{[t]}\right). \tag{23}$$

The likelihood $l\left(\mathbf{d}_{[t]}\right)$ of observing a new measurement $\mathbf{d}_{[t]}$ given previous measurements $\mathbf{d}_{[1]}, \ldots, \mathbf{d}_{[t-1]}$ is the value of a probability density function $p\left(\mathbf{d} \mid \mathbf{d}_{[1]}, \ldots, \mathbf{d}_{[t-1]}\right)$ at $\mathbf{d}_{[t]}$. By combining equations (21) and (23), the likelihood may be expressed as

$$\begin{aligned} l\left(\mathbf{d}_{[t]}\right) = p\left(\mathbf{d}_{[t]} \mid \mathbf{d}_{[1]}, \ldots, \mathbf{d}_{[t-1]}\right) &= \int p\left(\mathbf{d}_{[t]} \mid \mathbf{x}\right) p\left(\mathbf{x} \mid \mathbf{d}_{[1]}, \ldots, \mathbf{d}_{[t-1]}\right) d\mathbf{x} \\ &= \int f\left(\mathbf{d}_{[t]} \mid \mathbf{H}\mathbf{x}, \mathbf{Q}_{\mathbf{d}[t]}\right) f\left(\mathbf{x} \mid \hat{\mathbf{x}}_{[t-1]}, \mathbf{P}_{[t-1]}\right) d\mathbf{x} \\ &= f\left(\mathbf{d}_{[t]} \mid \mathbf{H}\hat{\mathbf{x}}_{[t-1]}, \mathbf{Q}_{\mathbf{d}[t]} + \mathbf{H}\mathbf{P}_{[t-1]}\mathbf{H}^{\intercal}\right), \end{aligned} \tag{24}$$

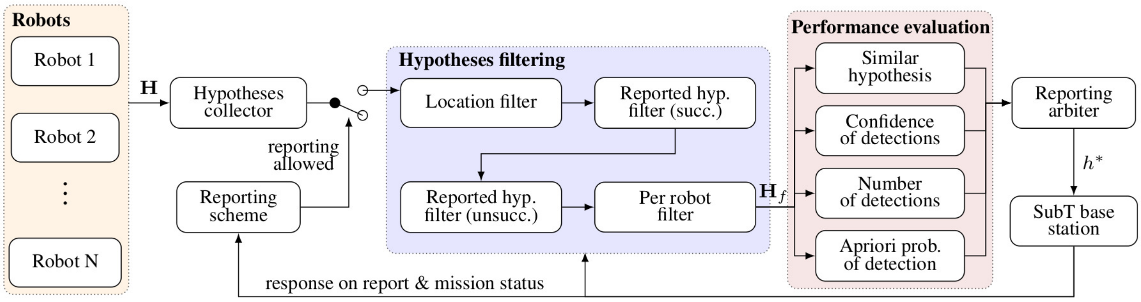


Figure 25. Illustration of the automatic reporting process from the Virtual Track.

which is the value of the probability density function of a Gaussian distribution with mean $\mathbf{H}\hat{\mathbf{x}}_{[t-1]}$ and covariance $\mathbf{Q}_{\mathbf{d}[t]} + \mathbf{H}\mathbf{P}_{[t-1]}\mathbf{H}^\intercal$ at $\mathbf{d}_{[t]}$. This expression is used to determine the detection-to-hypothesis association at each step according to equation (16), as described in the previous section.

10.4. Arbiter for artifact reporting

In contrast to the system part of the competition, the Virtual Track requires substituting the human operator with an autonomous arbiter for artifact reporting. The main functionality of the autonomous base station resides in collecting the hypotheses from the robots and reporting the location of artifacts. The number of reports in each run is limited and usually lower than the number of hypotheses collected from all robots. Therefore, a subset of hypotheses needs to be chosen so that the expected score is maximized. The implemented reporting strategy is based on filtering the collected hypotheses by considering their location and artifact type, followed by evaluating the performance index of particular hypotheses. The entire workflow is illustrated in Figure 25.

The autonomous base station collects the hypotheses from individual robots throughout the entire run. The predefined reporting scheme specifies the maximum allowed number of reports at particular time instants of the mission. Most of the reports are saved to the last minutes of the mission when the base station holds most of the information collected from the robots. However, some reports are allowed sooner during the mission to tackle the problem of unreliable communication and prevent a failure to report all hypotheses before the time limit exceeds. When the reporting scheme allows for submitting a report, the collected hypotheses are processed to obtain the best available hypothesis h^* in a set of all collected hypotheses $\mathbf{H}$. First, the hypotheses are filtered using information about previous reports, their validity, location, and per robot limits on the number of reports and minimum success rate. The final set of filtered hypotheses is obtained as

$$\mathbf{H}_f = \mathbf{H} \setminus \{\mathbf{H}_{\text{area}} \cup \mathbf{H}_{\text{succ}} \cup \mathbf{H}_{\text{unsucc}} \cup \mathbf{H}_r\}, \tag{25}$$

where $\mathbf{H}_{\text{area}}$ stands for the hypotheses located outside of the competition course, $\mathbf{H}_{\text{succ}}$ stands for hypotheses in the vicinity of the successful reports of the same artifact class, $\mathbf{H}_{\text{unsucc}}$ contain hypotheses in the vicinity of the unsuccessful reports of the same artifact class, and $\mathbf{H}_r$ represents the hypotheses of robots that have exceeded their own limit on reports and concurrently have a low success rate of their submitted hypotheses. The performance index for a hypothesis h_i is computed as

$$P(h_i) = \alpha p_r + \beta p_c + \gamma p_n + \delta p_a, \tag{26}$$

where the values p_r, p_c, p_n, p_a represent the percentile of particular performance indices of hypothesis h_i among all hypotheses in $\mathbf{H}_f$, and α, β, γ, δ are the weight coefficients. The particular performance indices are related to the number of robots with a similar hypothesis (p_r), the overall confidence of the detections assigned to the hypothesis (p_c), the number of detections assigned to the hypothesis (p_n), and the apriori probability of detection of a particular object (p_a). The next hypothesis to be

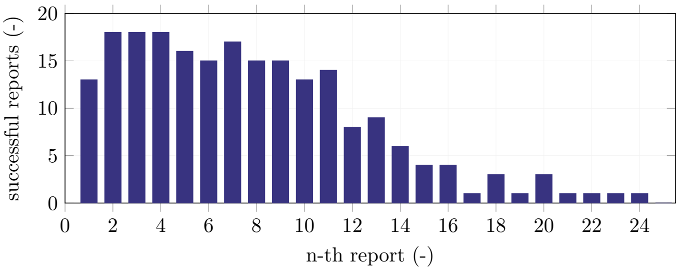


Figure 26. The distribution of successful reports over particular reporting attempts during all runs of the SubT Virtual Track Prize Round. The lower success rate of the first attempt in comparison to later attempts is caused by the early time of the first report, which was allowed 100 s after the start of the mission. By this time, only a single UAV had already entered the course, and thus the number of available hypotheses to choose from was low.

reported h^* is chosen based on the following equation:

$$h^* = \arg\max_{h_i \in \mathbf{H}_f} P(h_i). \tag{27}$$

The distribution of successful reports over particular reporting attempts during all runs of the SubT Virtual Track Prize Round is shown in Figure 26. In the Systems Track, the autonomous arbiter was not used as the decision-making of the human operator regarding which hypotheses to report was superior to the autonomous arbiter, which operated based on a fixed set of rules.

11. Mission control

The proposed system is designed for fully autonomous operation, so that the rescue team can benefit from the autonomous reconnaissance of the UAV without the need for any additional personnel operating the UAV. The DARPA SubT competition reflects this requirement on autonomy by allowing only robots without human operators to enter the course. In theory, the robots could be teleoperated (Moniruzzaman et al., 2022). However, this is not scalable with the number of robots. Moreover, for teleoperation, a reliable communication link between the robot and the operator is required, but is often not available, especially deeper in the subterranean environment where impenetrable walls diminish signal propagation. Thus the correct execution of an autonomous mission relies on a state machine that governs the high-level actions of the UAV.

11.1. State machine

The state machine applied in the SubT System Finals consists of 12 fundamental states. In the first state, the status of components that are vital to the mission is checked to ensure that the mission will be accomplished. Both the software components (*localization, mapping, planning, artifact detection, artifact localization, database*) and hardware components (*LiDAR, RGB cameras, depth cameras, mobilicom unit*) are checked prior to the mission. This component health check is crucial as, while still in the staging area, any potential component failures can be addressed, but it is not possible when the UAV is already flying.

When all components are running correctly, the UAV enables the output of the reference controller, transits to *WAITING FOR TAKEOFF* state, and waits for approval from the safety operator to start the mission. The approval required to guarantee the safety of the personnel moving in the vicinity of the UAV is given by arming the UAV and transferring the control of the UAV fully to the onboard computer by toggling the Radio Controller (RC) switch. After the approval to start, the UAV waits for a specified safety timeout in the *READY FOR TAKEOFF* state while signaling the imminent takeoff by flashing LEDs. In this state, the approval can be taken back by the safety operator. After the timeout elapsed, the *PERFORMING TAKEOFF* state is entered, during which the UAV ascends until reaching the desired takeoff height.

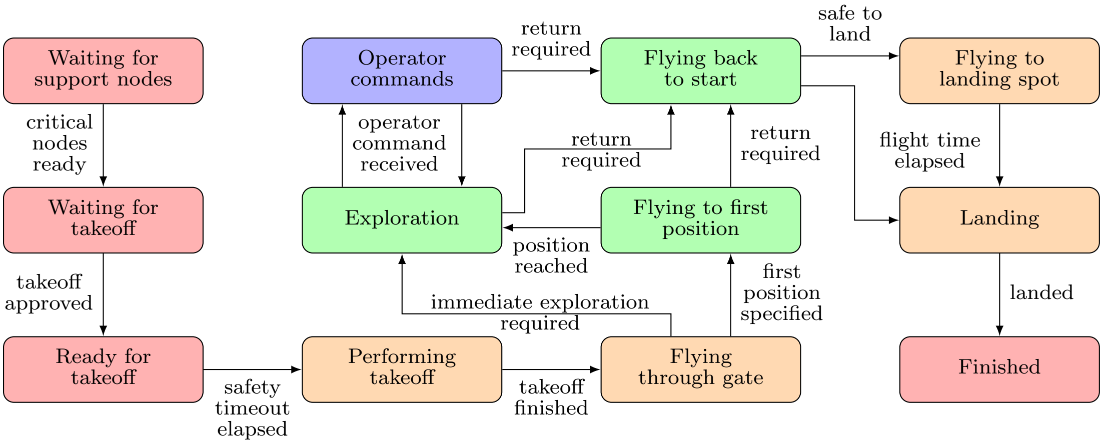


Figure 27. Simplified version of the state machine governing the autonomous mission in SubT Systems Track.

In the next state (*FLYING THROUGH GATE*), the UAV is navigated to a position inside the area to be explored. Once this position is reached, the space behind the UAV is virtually closed to prevent flight back towards the rescue personnel. If the rescuers have some prior knowledge about the environment, e.g., they see a door to which they want to send the UAV, they can optionally specify this first position to steer the UAV in that direction. After reaching this first position or if the flight to the first position is not requested, the UAV enters the *EXPLORATION* state. In this state, the UAV fulfills the primary mission goals until the upper bound of the estimated time to return is equal to the remaining flight time. Then the UAV initiates returning to the takeoff position in the state *FLYING BACK TO START*.

The return position is the takeoff position by default, but the operator can request any other position (e.g., to serve as a communication retranslation node) to which the UAV tries to return. After the position is reached, the UAV flies to the nearest safe landing spot as described in Section 5.3, and the *LANDING* state is entered. The landing is also triggered when the flight time is elapsed during the *FLYING BACK TO START* or *FLYING TO LANDING SPOT* states. When the UAV lands, it enters the *FINISHED* state, in which it turns off the motors, Light-Emitting Diodes (LEDs), LiDAR, and other components except the communication modules to conserve battery power for retranslating communications.

The required communication between the UAV and its operator during the start of the mission is limited to signals provided by the RC and visual signals provided by flashing LEDs. This enables very fast deployment of the UAV that automatically starts all necessary software components once the onboard computer is powered on and provides the information about being prepared to start by a single long flash of LEDs. After that, the operator can approve the mission by the remote controller without the need for any additional communication or commanding of the UAV. Following this automated procedure, the UAVs are prepared to start one minute after the battery is plugged in.

A *FAULT* state (not shown in the simplified diagram in Figure 27) can be entered from all states in which the UAV is in the air (all states except the ones colored red in Figure 27). The *FAULT* state is entered only when it is detected that the mission cannot continue safely. In such a case, a controlled emergency landing is initiated if a position estimate is available. When a position estimate cannot be provided the emergency landing escalates into the failsafe landing, during which the UAV gradually lowers its thrust, while maintaining zero tilt. After contact with the ground is detected, the motors are turned off and the UAV is disarmed. The *FAULT* state is final, i.e., the mission cannot continue due to the failures, which triggered the transition into this state. The conditions for entering the *FAULT* state are the following.

- Data from a sensor critical for localization are not available for 1 s. This situation can happen in case of a hardware failure, detached cable, power supply failure, sensor driver bug, etc.

- The control error exceeds 2 m, which can occur with a diverging state estimate, overloaded CPU, or insufficient thrust.
- The state estimate is not available for 0.1 s, which can be caused by a bug in the state estimation module or an overloaded CPU.
- The innovation of the state estimation exceeds 2 m. Innovation is the difference between current state and a correction coming from a localization algorithm. Large innovation indicates a discrete step in the localization algorithm.
- A maximum thrust threshold of 80 % is exceeded for 1 s. This condition is triggered when a discharged battery cannot provide enough current to perform the desired motion. A faulty or older battery with many discharge cycles might struggle to provide sufficient current sooner than is the expected flight time.
- A tilt over 75° is detected, which can happen if a discrete step appears in the state estimate or when the UAV collides with an obstacle.

The state machine applied in the Virtual Track of the SubT Challenge differs only in a few states given by the specifics of the simulation environment. First, it does not contain the operator commands states that are not available in a virtual environment. Second, it contains two additional states, *BACKTRACKING* and *AVOIDING COLLISIONS*. The *BACKTRACKING* state is entered when the UAV is stuck in a fog and tries to escape from it by backtracking to the most recent collision-free poses, ignoring the occupied cells in the current map (see Section 5.2.1 for details). In the *AVOIDING COLLISIONS* state, the UAV is avoiding collision with the UAVs of higher priority by stopping the lateral motion and decreasing its altitude. We have decided against using collision avoidance in the Systems Track due to the low probability of collision, and high probability of deadlocks in narrow corridors.

11.2. Operator commands

While the UAV is capable of carrying out the mission on its own in the fully autonomous mode, the operator can intervene by issuing an operator command to influence the behavior of the UAV. All operator commands can be activated only in the *EXPLORATION* state and in the operator command states, in which the UAV performs its primary goal. Allowing operator commands in other states would interfere with the takeoff, returning, and landing processes. The commands are transmitted from the operator's base station to the UAV through the available communication modalities described in Section 11.4. The following commands are available for the operator:

- **Explore to position**. The operator can bias the automatic goal selection process by issuing the *Explore to position* command. After the command is received by the UAV, the currently used reward function for evaluating viewpoints is extended by a term that penalizes the Euclidean distance of the viewpoint from the desired position $\mathbf{p}_D$. The term added to the reward function for a viewpoint ξ is simply

  $$\Delta R(\xi_{\text{UAV}}, \xi, \mathbf{p}_D) = -c_{oc} \left|\mathbf{p}_\xi - \mathbf{p}_D\right| . \tag{28}$$

  Such modification of the reward function causes the viewpoints closer to the desired positions to be preferred over farther viewpoints. The assertiveness of reaching the desired position can be controlled by the coefficient c_{oc}. If this is set too high, it might force the viewpoints with a minimal distance from obstacles and low information value to be selected.
- **Plan to position**. The *Plan to position* command bypasses the viewpoint selection process and requests the planner to find a path directly to the specified position. When the requested position is not reachable, i.e., it is in an occupied or unknown space, the planner will find the path to the closest point using the Euclidean distance heuristic function. Thus this command should be used primarily for reaching an already visited position, e.g., to land there and retranslate communication from robots that are already further in the environment, or to approach a stuck robot to retrieve its data.

- **Set return position**. Under normal operation, the UAV returns to the staging area when its battery is depleted. The operator can change the return position by issuing the *Set return position* command. This can save valuable flight time of the UAV when a communication chain is already established.
- **Stop**. The operator can also halt the movement of the UAV by issuing the *Stop* command. This command is useful when the operator wants to inspect an interesting area in more detail, prevent the UAV from going into a noninformative or dangerous area, or temporarily retranslate communications. Moreover, this command is a prerequisite for calling the *Land* command.
- **Land**. It is possible to land the UAV prematurely before the end of the mission by issuing the *Land* command. The expected use case involves landing the UAV at a position advantageous for extending the communication network. Before calling the *Land* command, the *Stop* command must be called to prevent an accidental landing at an incorrect location, due to the arbitrary delay of the command sent through an unreliable network. The system does not guarantee landing at the exact specified position, as a safe landing spot is found in the vicinity of the requested position.
- **Return home**. The *Return home* command switches the UAV to the *returning* state, as defined in Section 8.2. In this state, the UAV uses the navigation module to get as close as possible to the specified return position.
- **Resume autonomy**. The last operator command cancels the behavior that was forced by previous operator commands (except *Land* and *Set return position*). This causes the UAV to resume autonomous exploration, start its return, or land (depending on the flight time left).

11.3. Operator interface

Only a single human (operator) could view the mission-specific data sent by the robots to the base station. His main task was to analyze the artifact hypotheses and report ones that seemed correct to the DARPA server to score points. He could also influence the behavior of the robots by issuing high-level operator commands (Section 11.2).

To facilitate his responsibility, each of the two tasks has a dedicated interface. Commands are issued from the RViz-based interface with each command mapped to a unique keyboard shortcut. The operator also often used live camera streams from the robots to get contextual information about the environment. This information was essential for deciding where each robot should be sent (e.g., quadrupeds to urban sections) and also for quick assessment of why a robot could be stuck.

The second interface for artifact hypotheses management is also based on RViz with a custom rqt plugin for viewing the details of each hypothesis including the image, number of detections, class probabilities, and position. These properties help the operator decide whether to send the hypothesis to the DARPA scoring server or decline it. Manual refinement of hypotheses poses is also possible by dragging them on the map.

The GUI was displayed on a semi-mobile workstation with 3 integrated displays and one external monitor standing on top of the workstation. The arrangement of the 4-displays is shown in Figure 28. From our experience, the more the operator sees without keyboard and mouse interaction, the better for his performance.

Apart from the human operator who could view all mission data, the rules also allowed the other staging area personnel to view status data. We have thus set up a diagnostics console on a computer outside the staging area. This console showed useful diagnostics information that could be relayed via voice to the human operator.

11.4. Communication

The developed system assumes an unreliable bidirectional low-bandwidth communication network with intermittent dropouts. It should be mentioned that two meshing-capable wireless technologies are used on the hardware level—2.3 GHz Mobilicom and 868 or 915 MHz motes, with details of

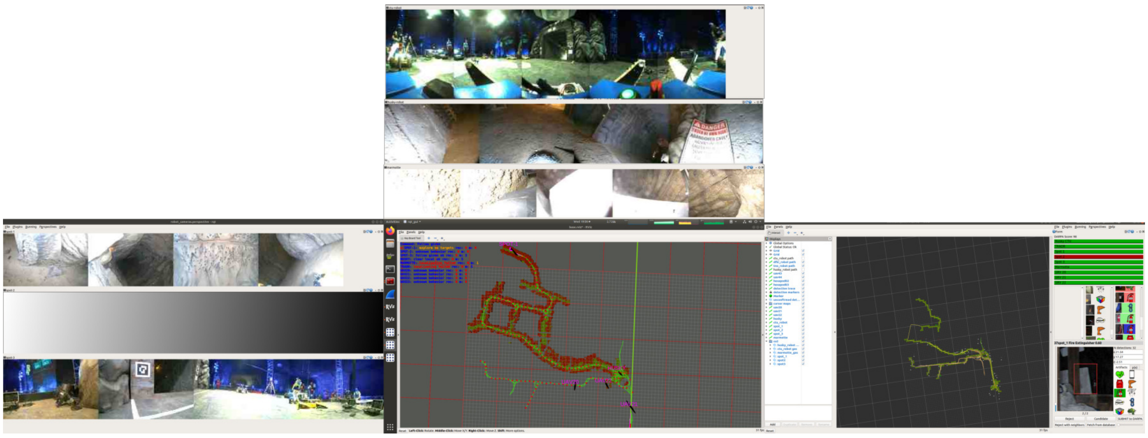


Figure 28. The operator interface display arrangement. Top screen shows live views of cameras from 3 UGVs. In bottom row, left to right, are screens with another 3 live UGV streams, the control GUI and artifact reporting GUI. The bottom left screen could also show a terminal window that was used for direct execution of scripts on the robots (as a fallback for a failure case that was not handled by the control GUI). Figure 2 illustrates the physical look of the setup of the workstation with 4 displays.

both available in (Roucek et al., 2020). The motes are also dropped by UGVs as deployable range-extending battery-powered modules (Bayer and Faigl, 2020) similarly to (Ginting et al., 2021; Saboia et al., 2022) to build a communication mesh network. Our custom-made motes have lower bandwidth ($100\,\mathrm{B\,s^{-1}}$) than (Ginting et al., 2021; Saboia et al., 2022), which is compensated by sending only necessary compressed data. Moreover, bandwidth-intensive data are sent through a $1\,\mathrm{MB\,s^{-1}}$ Mobilicom network. This multimodal communication approach is robust to the failure of either Mobilicom or motes as both are able to transfer mission-critical data.

This paper focuses on high-level usage of the communication network, which is used as a black box, and as such the low-level layers of the communication protocol are not discussed.

The developed system benefits from available connections to other agents and the base station in multiple ways. First, when a robot detects an artifact, the detection with its estimated position is shared over the network instead of returning physically to the base station, thus saving time valuable for the success of the mission. Second, the individual agents can share the information about the already explored volume in the form of a topological-volumetric map (LTVMap) introduced in Section 7.4. The knowledge of other agents' topological-volumetric maps penalized regions already explored by other robots, which encourages splitting of the robot team and covering a larger volume over the same time period as shown in Figure 29. Third, each robot shares its position with the base station, so that the operator has an overview of where all robots are located. The operator can then influence the future behavior of any robot in the communication range by sending an operator command (Section 11.2). Last, positions of the communication nodes (breadcrumbs or landed UAVs), which form the communication network shown in Figure 30, are sent to be used for returning to the communication range when the remaining flight time is low.

11.5. Calibrating global reference frame

The entire navigation system of heterogeneous robots within the CTU-CRAS-NORLAB team is decentralized under the assumption of a shared coordinate frame—the world coordinate frame O_W. To obtain the transformation of a robot's local origin within the world frame, the staging area of the competition environment provides a set of visual tags and a set of reflective markers, both with precisely known poses within the world (see the markers mounted on the entrance to the environment in Figure 31). The reflective markers are used within our 6-DOF calibration procedure in which a Leica TS16 total station is employed to measure 3D points with sub-millimeter accuracy. The origin

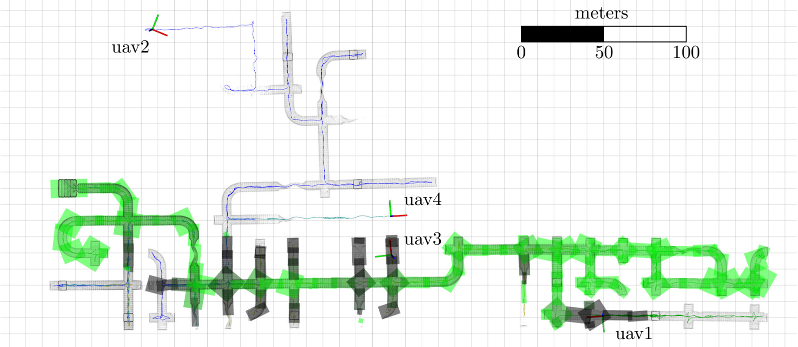


Figure 29. Example of the dispersed exploration of a tunnel system during the first run in world 1 of the virtual track. Only LTVMap from UAV1 is shown for clarity, other UAVs received this map and maps from the other UAVs. Instead of exploring again the same places as UAV1, both UAV2 and UAV4 explore previously unvisited corridors. Dark parts of LTVMap in this figure are not yet fully explored, so UAV3 flies to inspect these areas to not miss any potentially hidden artifacts.

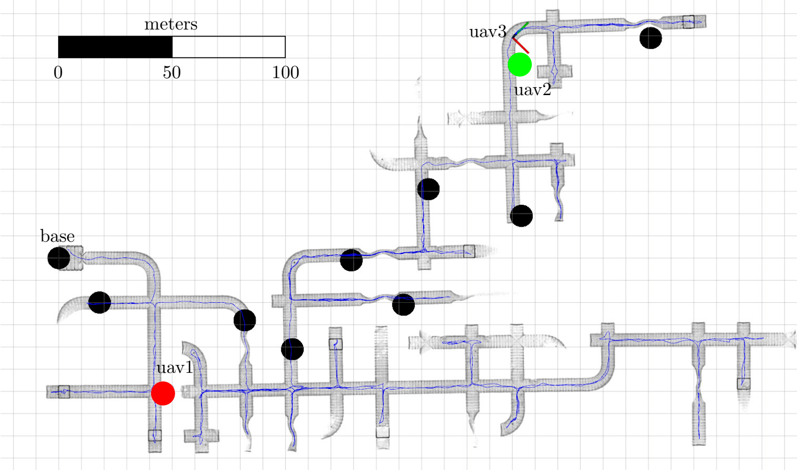


Figure 30. A communication network consisting of a base station and 8 breadcrumbs (black) deployed by the UGVs and 2 UAVs from the 3rd run in world 1 of the virtual track. UAV3 with its trajectory shown in blue could explore further thanks to the deployed communication nodes. Without the communication network, the UAV would have to return to the staging area, thus traveling additional 500 m from its final landing position.

$\mathbf{T}_{TS}^{W}$ of the total station in the world is derived from measuring known in-world marker poses and used in deriving $\mathbf{T}_{B}^{W}$ of a robot B.

To calibrate the pose of a single robot B after $\mathbf{T}_{TS}^{W}$ is known, 4 known points on the robot's frame need to be measured, used in estimating $\mathbf{T}_{B}^{W}$, and sent to the information database (see Section 11.4) or directly to the robot. As the number of robots in the CTU-CRAS-NORLAB team deployments reached up to 9 robots per run (see Figure 31), the overhead for robots-to-world calibration decelerated the rate of robot deployments as well as limited the possibilities for quick in-situ decision-making. To speed up the calibration pipeline for UAVs with limited flight distance (and hence with greater room for calibration errors), just a single UAV A needs to be calibrated with the total station wherein the initial pose of the remaining UAVs B is estimated from on-board LiDAR data. The known transformation $\mathbf{T}_{A}^{W}$ and pre-takeoff LiDAR data $\mathbf{D}_{A}$ of a robot A are shared throughout the robots and used to estimate $\mathbf{T}_{B}^{W}$. The transformation $\mathbf{T}_{B}^{A}$ is estimated by registering source LiDAR data $\mathbf{D}_{B}$ onto target data $\mathbf{D}_{A}$ using Iterative Closest Point (ICP) with extremely tight constraints in matching the rotation component of the transformation. The tight rotation

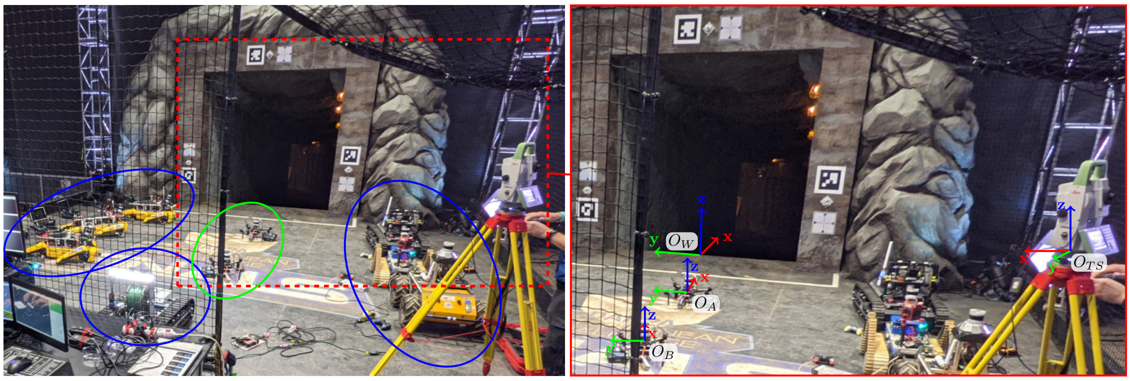


Figure 31. Example robot distribution (7 UGV robots in blue, 2 UAV robots in green) of team CTU-CRAS-NORLAB within the staging area of Systems Track environment of DARPA SubT Challenge, 2021. The Right figure highlights the reference frames of interest—the world origin O_W together with the origin of the Leica total station O_{TS} used for calibrating local robot origins O_A and O_B within the world.

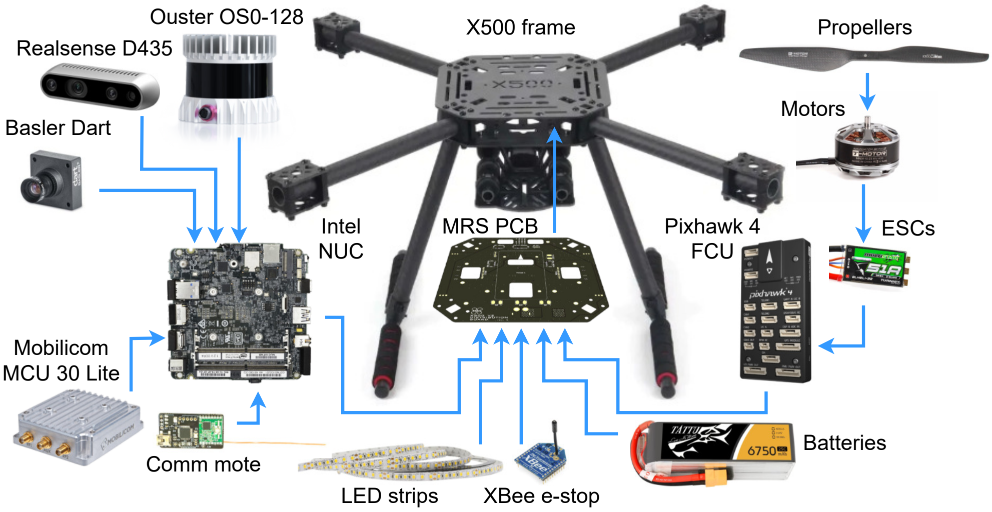


Figure 32. The interconnection of hardware components that were used is the Systems Track.

constraints are important as frame-orientation misalignments are the largest source of absolute error during deep deployments. The pose of robot B in the world is then given by $\mathbf{T}_B^W = \mathbf{T}_B^A \mathbf{T}_A^W$.

12. Hardware platform

The components of our S&R UAV were carefully selected to optimize the flight time and perception capabilities based on years of experience with building aerial robots for research (Ahmad et al., 2021), competitions (Walter et al., 2022), inspection (Silano et al., 2021), documentation (Kratky et al., 2021b) and aerial filming (Kratky et al., 2021). All platforms we have designed for diverse tasks and purposes including DARPA SubT are presented in (Hert et al., 2022).

Our platform is built upon the Holybro X500 quadrotor frame. The 500 mm frame is made entirely of carbon fiber, therefore it is stiff and light. Moreover, the arm length can be changed to accommodate different propellers. A description of all components that are mounted on the UAV frame follows. The connections of the components are depicted in Figure 32. Our team designed and manufactured a custom Printed Circuit Board (PCB) that replaced the top board of the X500 frame. This PCB (see Figure 34) supplies battery power to individual Electronic Speed Controllers (ESCs), integrates several independent power supplies and provides a communication interface among the main computer, the Pixhawk flight controller, and MRS Modules. MRS Modules are small expansion boards that provide additional functionality and have a standardized electrical and mechanical interface. The UAV can be configured with different MRS Modules depending on the required capabilities. The PCB is connected to the main flight 4S lithium-polymer battery, which provides 14.0 V–16.8 V depending on the state of charge. The battery voltage is used to directly power the ESCs and the Intel NUC main computer. The board then integrates three independent 5 V/3 A buck converters, two to provide redundant power for the Pixhawk flight controller and one to power the MRS Modules. One 24 V/2 A boost converter is used to power the Ouster OS0-128 3D LiDAR scanner. The board has two slots for MRS Modules, one is used to control 12 V LED strips which provide illumination for the onboard RGB cameras. The second module is an interface for an XBee radio module, used as an e-stop receiver. Communication among the main computer, Pixhawk, and MRS Modules is provided by FT4232 Quad USB-UART bridge, which is integrated into the PCB. We selected MN3510 KV700 motors from T-motor and paired them with 13-inch carbon fiber propellers for large payload capacity and propulsion efficiency. The motors are driven by Turnigy Bl-Heli32 51A ESCs, as they are lightweight and easily configurable.

The 3D LiDAR was upgraded to the OS0-128 model, which features 128 scanning lines and wide 90° vertical field of view, which allows for perceiving the surroundings of the UAV in the challenging underground environments. Despite the wide coverage of the LiDAR sensor, there are still blind spots above and below the UAV when mounted horizontally. To cover these spots, we use two Intel Realsense D435 RGBD cameras, facing up and down. This enables the UAV to fly directly upwards, even in cluttered vertical shafts, without risking collision. Both of the RGBD cameras are also used for mapping and artifact detection. Additionally, the bottom facing RGBD camera is used for landing site detection. The platform is equipped with two (left and right) dedicated artifact detection cameras, the Basler Dart daA1600 with 97° horizontal FOV lens, and sufficient lighting provided by LED strips. All algorithms run on the onboard Intel NUC i7-10710U CPU with 6 physical cores and the detection CNN utilizes the integrated Intel UHD GPU.

The high-power Mobilicom MCU-30 Lite wireless communication module provides long-range connection between robots and the base station while keeping low weight of 168 g. In some topologically complex areas, even the high-power Mobilicom cannot assure reliable connection between the units, so it is supported by smaller communication motes, which are also dropped as breadcrumbs by the UGVs to improve the signal range. These motes are compact communication modules based on the RFM69HCW transceiver working at 868 MHz or 915 MHz with 100 mW transmission power and 100 B s^{-1} data bandwidth. The performance of the motes was analyzed in the Bull Rock cave (Bayer and Faigl, 2020) and first deployed at the Urban Circuit (Roucek et al., 2020). The WiFi unit of the onboard Intel NUC computer was not used for any communication.

Finally, the large payload capacity of the UAV allowed us to extend the flight time by using a larger battery. We used two 4S 6750 mA h Li-Po batteries in parallel. Instead of a larger battery, two smaller batteries were used due to the 100 W h limit for aircraft transportation. This gave the UAV a flight time of 25 min with a total mass of 3.3 kg.

The X500 platform (Figure 33) is capable of flying in dense indoor environments, even in tight vertical shafts, while being able to localize itself with the required accuracy. It has four different cameras for artifact detection, is able to communicate and form mesh networks with other robots, and possesses a long flight time.

Furthermore, this platform was also replicated in the virtual competition with the same parameters as the physical counterpart. All of the teams except for two used the X500 platforms in the Virtual Track due to its long flight time, substantial sensor suit, and agile dynamics.

13. Technical details of hardware deployment

With a few exceptions, the components of the UAV software stack deployed in the Virtual and Systems tracks are equal, yet the available processing powers are not. The Virtual Track yields a

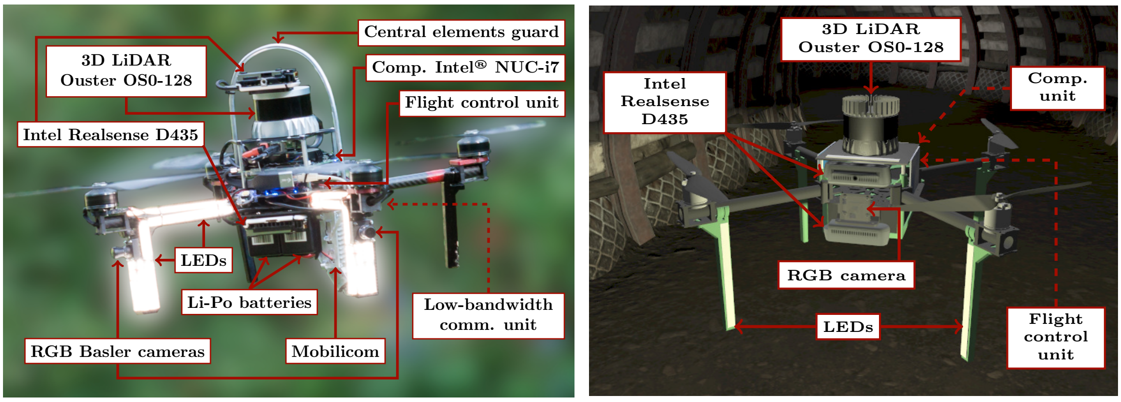


Figure 33. X500 platform used in the Systems Track (left) and Virtual Track model counterpart (right).

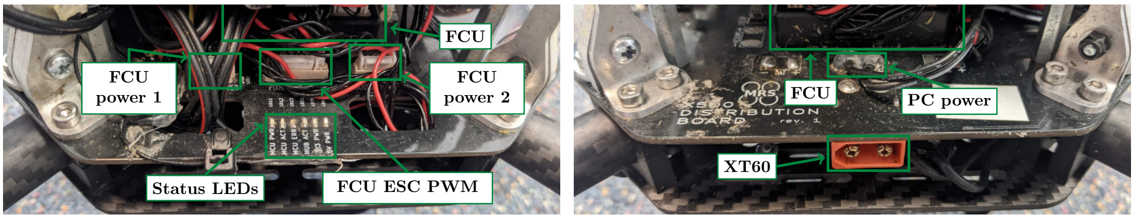


Figure 34. Custom PCB replacing the top board of the X500 frame from the front (left) and back (right). The FCU is powered by a dual redundant 5 V power supply, while the Intel NUC computer is powered directly from the 4-cell battery at 14.0 V–16.8 V. The battery is connected using the XT60 connector and the status of individual modules of the PCB are signalized by 6 status LEDs. The ESCs are also connected to our PCB.

low real-time simulation factor. Together with the computational capacities of each simulated robot, it provides almost unlimited computational resources for running all algorithms with any desired resolution or maximal settings. On the other hand, the simulation-to-world transition requires the algorithms to run on the onboard processing units. This imposes hard requirements on the algorithms' optimization, as well as on minimization of the amount of data transfers and their latency. These requirements force us to

- compromise between accuracy and real-time performance in the system design (i.e., cutting out global optimization in on-board running SLAM),
- ensure real-time properties for systems handling critical factors of the mission (i.e., UAV control),
- optimize the data flow and the priorities of processing order within the software stack, and
- prevent any possible deadlocks from arising from outages of both synchronous, and asynchronous data.

Ensuring real-time settings for all systems of a robotic deployment is implausible, particularly in complex robotic-research projects where the stack design must allow for the system to function as a whole under limited real-world conditions. We summarize the specific aspects of the proposed ROS-based software stack, allowing us to transfer all components to on-board processing capacities. Thus providing full decentralization within a UAV team.

Software based on ROS 1 allows for connecting components under a *nodelet manager* in order to group *nodelet* plugins. In contrast to *node* configuration, the *nodelets* under a *manager* have shared memory and do not require copying data, a tool useful particularly in the case of passing large maps within the navigation stack. Our deployment stack consists of several *managers*, each of which

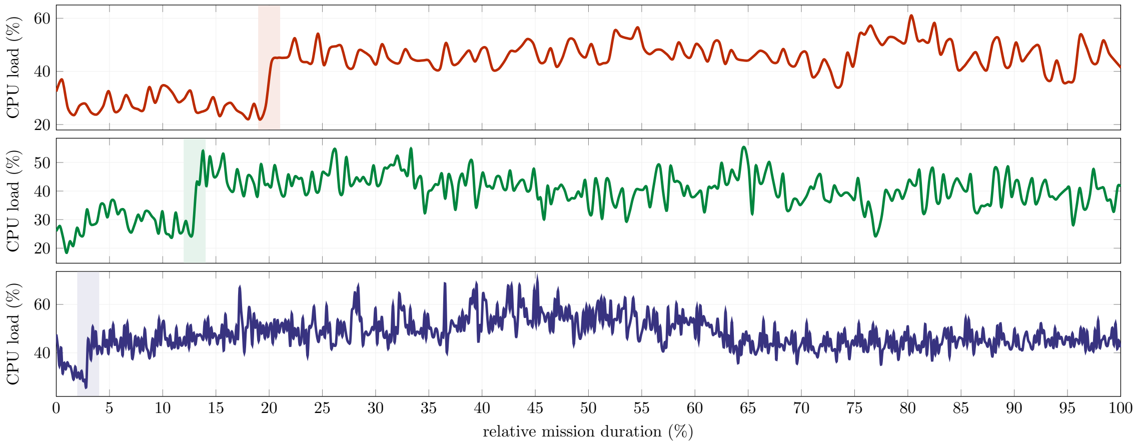


Figure 35. The CPU load of onboard computers of individual UAVs (*red*, *green*, *blue*) during the prize round of SubT Systems Track. The highlighted parts of the graph correspond to the start of processing onboard images by the object detection pipeline.

handles a distinctive part of the system. These include UAV control, preprocessing of LiDAR data and SLAM, preprocessing of RGBD data and dense mapping, navigation and path planning, and perception. The data flowing between these *managers* are copied, and thus the rate of sharing is subject to maximal reduction. To decrease the temporal and memory demands of algorithms, the resolution of input data and the output maps is decreased as much as possible within the scope and requirements of the desired application. The rate of saving data for after-mission analyses is also limited as much as possible, with no postreconstructable data being recorded at all.

In contrast to the system designs for UGV platforms, the delays in state estimation and control inputs are a critical subject for reduction. This is because excessive delays lead to destabilization of a multirotor aerial platform (see analysis on delay feasibility in Figure 11) as it is a dynamically unstable system requiring frequent feedback, even for simple hovering. The *nodelet managers* handling such critical parts of the system are prioritized at the CPU level, utilizing the negative `nice` values that prioritize the related processes during CPU scheduling. To decrease asynchronous demands on the CPU, nonprioritized components are penalized with positive `nice`. Furthermore, their scheduling is restricted on a predetermined set of threads in a multithreaded CPU. The primary subject of scheduling restriction is the perception pipeline containing a computationally heavy CNN, where static allocation reduces its asynchronous influence on the rest of the system at the cost of a limited processing rate. The effect of switching on the perception pipeline is visible in Figure 35, showing the CPU load of the three deployed UAVs during the DARPA SubT Systems Track. In other validation tests, the CPU load reached up to 90 % in 1500 s long missions within vast underground environments. Such an overloaded CPU results in frequent asynchronous delays, culminating to unpredictable and destructive behavior.

To limit the power consumption and hence, increase the maximum flight time, unsolicited hardware and software components can be temporarily powered off. These include switching off on-board lights in meaningless settings, disabling CNN processing when not needed, or powering off the LiDAR in the after-landing phase when the UAV is serving solely as a retranslation unit for communication.

14. System deployment

Throughout the development of the system presented in this paper, the individual components were extensively tested before integration. Deployments of the whole system were less frequent, but allowed testing the interaction of individual modules and verifying the ability to fulfill the primary objective of finding objects of interest in subterranean environments.

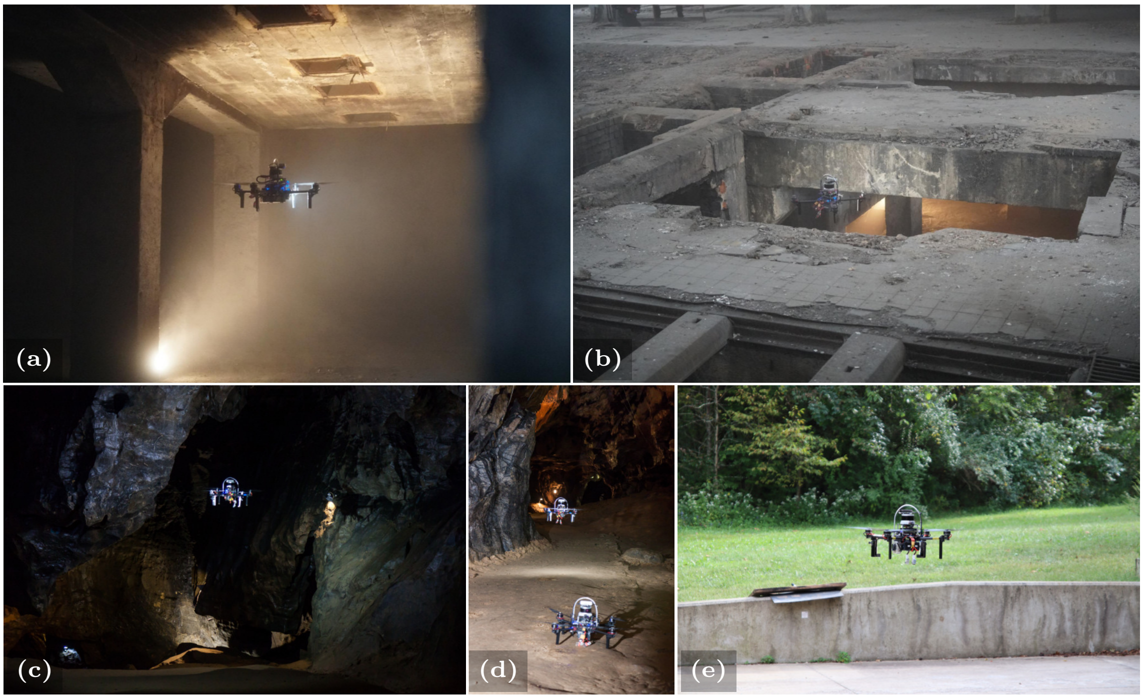


Figure 36. The verification of localization and perception in the following scenarios: data degraded by insufficient lighting and whirling dust (a), traversal of vertical narrow passage (b), performance in humid caves (c), multirobot exploration (d), and scalability with the environment size (e).

14.1. Continuous field verification

The S&R UAV system was continuously tested to empirically verify the correctness and reliability of the developed algorithms, strategies, and hardware. The UAVs were deployed into diverse types of environments, including historical and industrial buildings of varied levels of disintegration, in humid unstructured caves, a decommissioned underground military fortress, and vast outdoor rural areas. Some of these environments are shown in Figure 36. Such tests are critical for evaluating the performance under the stochastic influence of real-world conditions, which are typically not modeled in simulations. In particular, each perception mode is more or less degraded by ambient lighting or the lack of it, the fog with microscopic condensed droplets of water, smoke or dust particles, reflections on water or smooth surfaces, etc. The filtration of LiDAR and depth data from Section 5.2 therefore had to be tuned correctly to prevent the integration of false positives into the map, while keeping the actual obstacles. Moreover, the artifact detection system needed to work under a wide range of visibility conditions and chromatic shifts, for which it was necessary to collect artifact datasets from the mentioned environments.

14.2. DARPA SubT Final Event Systems Track

The Final Event, which was the culmination of the DARPA SubT competition, was organized in the Louisville Mega Cavern in Kentucky on September 23, 2021. The course consisted of all three environments from the previous circuits and contained all artifacts from previous events plus *the cube*, which was a new artifact for the Final Event. This section reports on the results achieved by the aerial part of the CTU-CRAS-NORLAB team. A total of 40 artifacts were distributed over 880 m long course, which was divided into 28 smaller sectors to track the team's progress. Every robot starts in the staging area, from which a single corridor leads to an intersection that branches into three ways. Each of the branches leads to one of the three specific environment types (tunnel, urban, and cave).

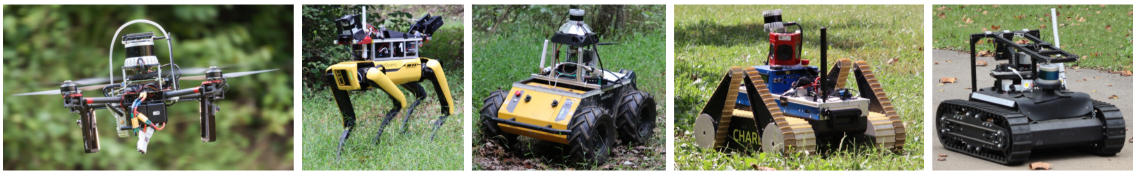


Figure 37. All robotic platforms used in the Prize Round. (From left to right) X500, Spot, Husky, TRADR, Marmotte.

Table 3. The summary of deployed robots in the Final Event sorted by deployment times. *Operable time* means how long the robot was operable, i.e., its computers were running and it could move. *Motion time* is the time the robot was moving faster than $0.1\,\mathrm{m\,s^{-1}}$. The row *Artifacts* shows the number of confirmed hypotheses as defined in Table 5.

Robot	**Spot 1**	**Red**	**Spot 2**	**Marmotte**	**Husky**	**Spot 3**	**TRADR**	**Blue**	**Green**
Locomotion	Legged	Aerial	Legged	Tracked	Wheeled	Legged	Tracked	Aerial	Aerial
Deploy time	0:20	2:00	4:00	7:20	12:40	17:32	28:20	36:00	46:30
Operable time	6:00	3:00	7:00	44:00	20:00	11:00	32:00	22:25	6:10
Motion time	3:00	2:32	2:00	6:00	5:00	9:00	4:00	15:22	4:33
Traveled	111 m	69 m	47 m	181 m	131 m	195 m	97 m	304 m	119 m
Artifacts	4	1	2	1	2	0	3	3	3
Sectors explored	4	2	2	2	2	5	4	4	4

Our team deployed a heterogeneous lineup of robots. A total of 3 legged robots (Spots), 2 tracked robots (TRADR, Marmotte), 3 aerial robots (X500), and 1 wheeled robot (Husky) robot were deployed in the Final Event Prize Round (see Figure 37). The Husky robot is a fast wheeled platform ($3.6\,\mathrm{km\,h^{-1}}$ max. speed) for exploration of easy terrain. Tracked Marmotte was also fast ($4\,\mathrm{km\,h^{-1}}$ max. speed) but could overcome obstacles larger than Husky could. Spots were the universal ground platform thanks to the ability to pass most terrain except slippery surfaces (max speed $5\,\mathrm{km\,h^{-1}}$). The highest traversability among the ground platforms was offered by the tracked TRADR robot thanks to its controllable flippers (however, maximum speed is approximately $2\,\mathrm{km\,h^{-1}}$). The primary role of the aerial robots was to explore areas unreachable by ground robots such as vertical shafts or paths blocked by obstacles. The detailed composition of the team is summarized in Table 3 together with deployment times and mission statistics. The payload of all UGVs consisted of Ouster OS0-128, 5–6× Basler Ace 2 or 1× PointGrey Ladybug 2 cameras, Xsens MTI-30 IMU, Mobilicom MCU-30 Lite, Nvidia Jetson Xavier AGX, Intel NUC 10i7FNK, LED illumination, SDC30 gas sensor, and communication motes.

14.2.1. UAV deployment summary

Three UAVs in total (*red*, *green*, and *blue*) were deployed in the 60 min long run. All UAVs used the Greedy strategy (Section 8.2.2) without VPE for the simplicity of its reward function, which made it easier to fine-tune the reward function coefficients for the competition environment and debug the UAVs' behavior. The UAV performance is summarized in Table 4 and the flight trajectories are plotted in Figure 44. The first UAV (*red*) took off just after the first UGV, arrived to the first intersection, explored 10 m of the tunnel section, returned to the intersection, flew to the cave branch where it collided with the Spot UGV (Figure 42a). The chronologically second deployed UAV was *blue*, which went into the urban branch where it traveled to a vertical alcove with a phone artifact. Then it returned to the start of the urban section, where it hovered until exhausting the battery (Figure 42c), because all viewpoints were blocked in its map corrupted by drift in the featureless urban corridor. The last deployed UAV was *green* that explored the tunnel section, where it was blocked by a dynamically added artificial wall (Figure 41). After flying through a cluttered tunnel corridor, the UAV collided with a metal rod protruding from the wall (Figure 42b).

Table 4. The mission statistics from the prize round of the Final Event. The localization accuracy was not evaluated for UAV blue. Obtaining the ground truth position using scan matching would have been extremely strenuous due to the degenerate geometry of LiDAR scans in the urban tunnel. This degeneracy also caused the onboard localization to drift several meters.

UAV	Red	Green	Blue
Localization accuracy:			
avg|max error in translation (m)	0.38 | 0.63	0.97 | 2.66	-
avg|max error in heading (°)	0.64 | 4.06	1.48 | 5.37	-
Safety clearance	0.4 m	0.11 m	0.21 m
Landing cause	Collision with UGV	Collision with a metal rod protruding from the wall	Depleted battery after being trapped in degraded map

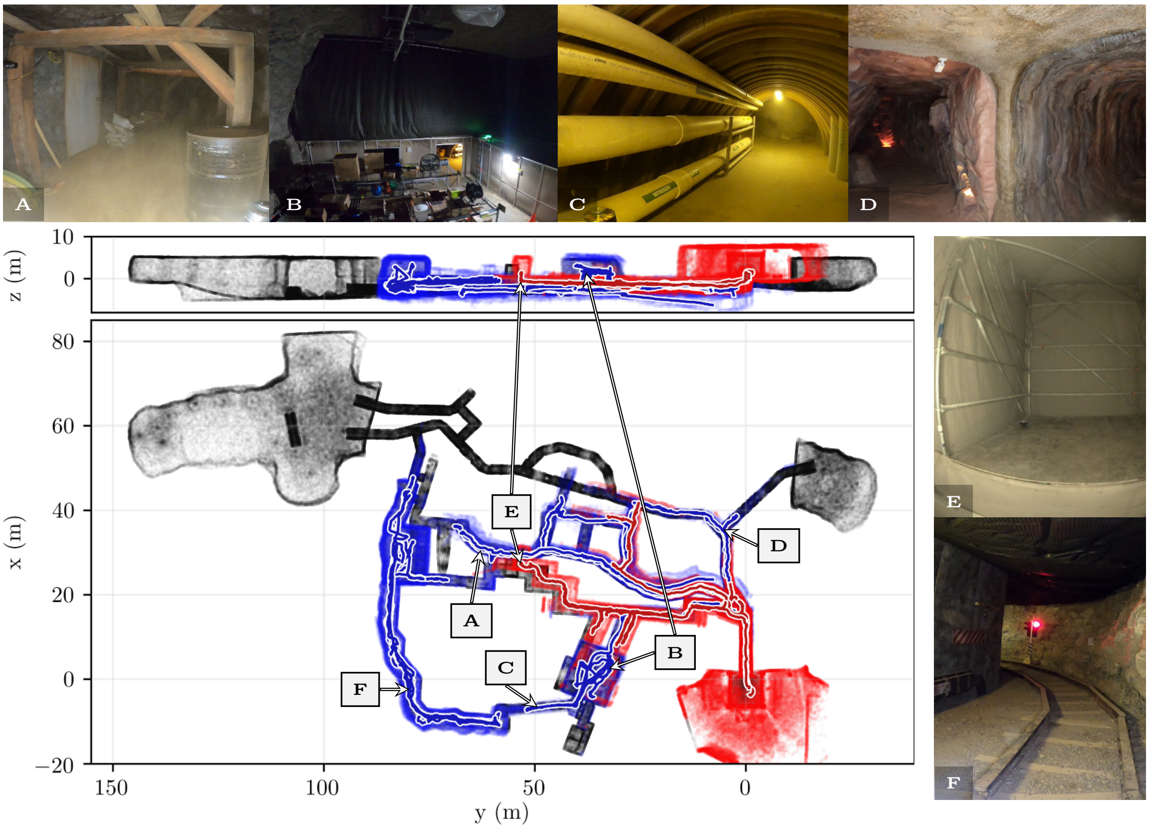


Figure 38. UAV trajectories and on-board-built maps of the environment from all flights during the prize round (colored in red) and the postevent testing (colored in blue) overlaid over the ground truth map (colored in black). The photos from on-board camera highlight the diversity and the narrow confines of the environment.

The maps and the trajectories of all our UAV flights during the prize round and the postevent testing are shown in Figure 38, together with summary of the mapping errors from these flights in Figure 39. The distance of the UAVs from the nearest obstacle during all flights in the prize round are shown in Figure 40.

14.2.2. Artifact detection discussion

The performance of the artifact detection and localization system is summarized in Table 6, and the number of artifacts detected by each UAV in Table 5. A total of seven artifacts appeared in

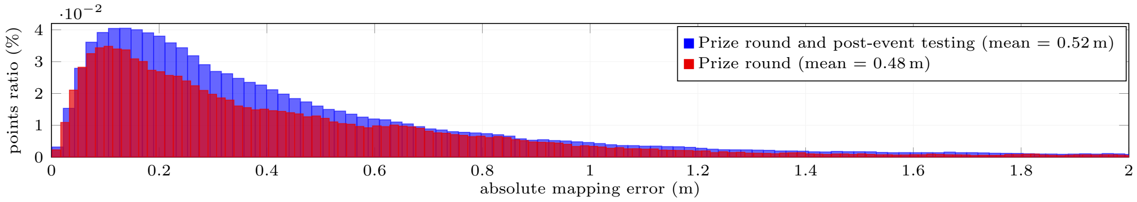


Figure 39. Distribution of mapping errors throughout the prize round and the postevent testing flights (colored in red and in blue in Figure 38) of DARPA SubT. The absolute mapping error denotes the distance between the ground truth map and concatenation of DenseMaps built with resolution of 20 cm on-board during particular UAV flights. The error metric is the Euclidean distance between a point from the on-board maps to its closest point in the ground truth map.

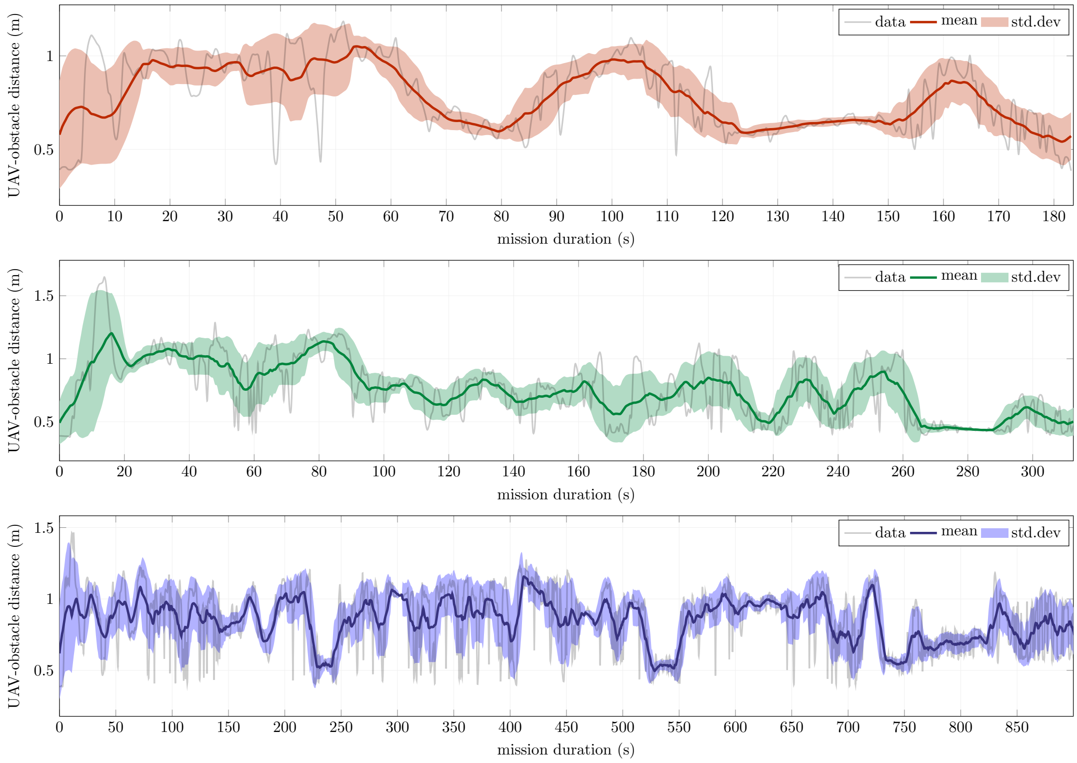


Figure 40. Distance between the center of the UAVs and the measured nearest obstacle during the prize round of the SubT Systems Track. The moving mean and standard deviation are computed over a 10 s long time window.

the camera images, and six artifacts were detected by the CNN. The detections with estimated bounding boxes from all UAVs are shown in Figure 45. The survivor *s2* was seen in three frames of the bottom camera. However, only a small part of the survivor sleeve was visible and the images were further degraded by motion blur, as can be seen in Figure 43. Thus the CNN did not manage to detect the artifact. From the six detections, the cellphone artifact *p1* was detected only on one image frame when the UAV *blue* peeked into the vertical shaft in the urban part. However, as explained in Section 10, a total of four detections are necessary to create a hypothesis and to confirm the position, and thus this single detection was discarded. Another missed point was the survivor *s1*, which was detected and localized within the 5 m limit, but the artifact was labeled as a cube instead of a survivor. The hypothesis was merged with a high number of false positives and, consequently, the correct image was not sent to the operator, who could not determine the correct class to report.

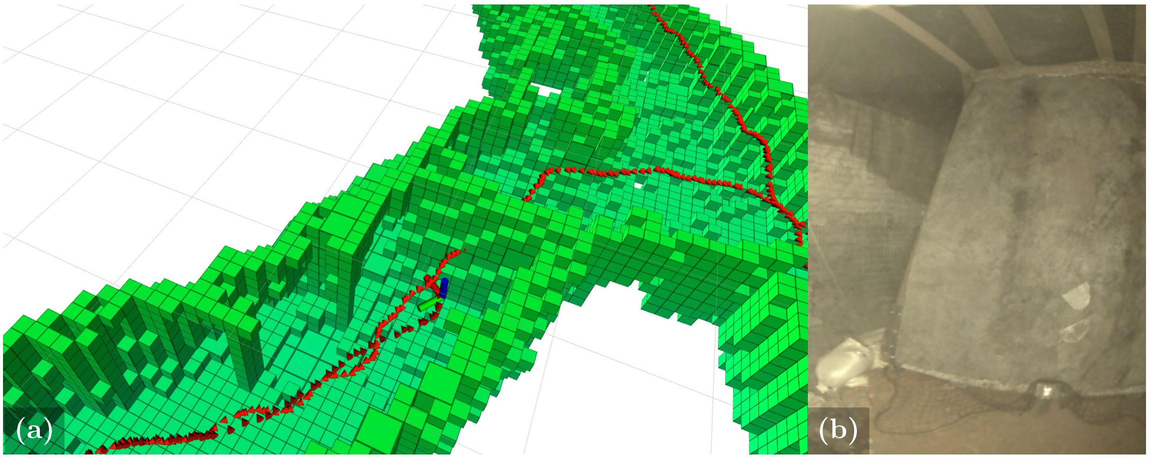


Figure 41. The artificial wall that blocked the way back for UAV *green* in the map (a) and in the camera image (b).

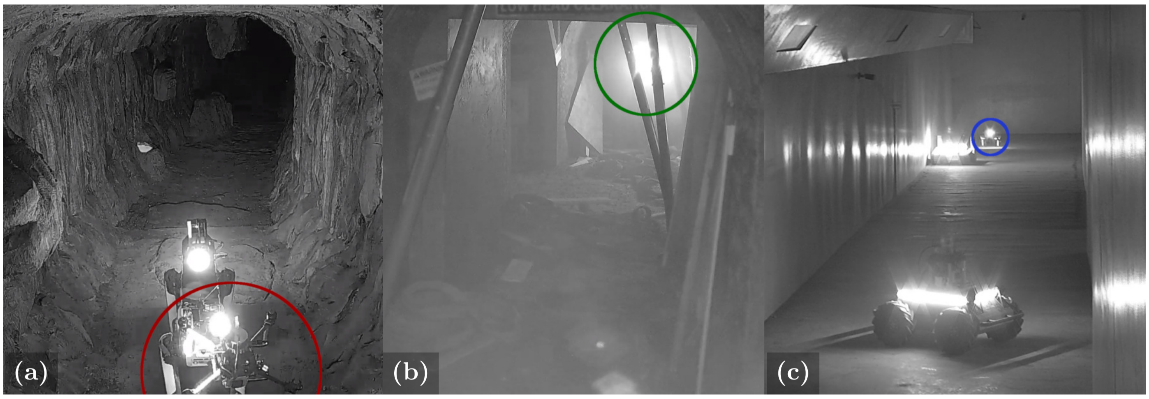


Figure 42. The landing events of all three UAVs. The UAV *red* (a) collided with the Spot UGV, UAV *green* (b) hit a metal rod protruding from the wall, and UAV *blue* (c) landed after its battery was exhausted by hovering while being trapped in a map corrupted by drift in the featureless corridor.

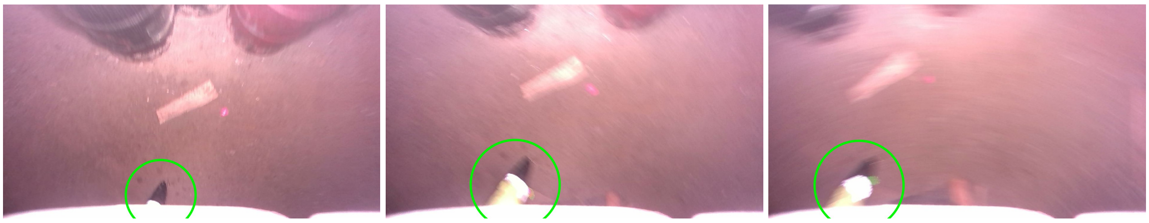


Figure 43. The only three image frames of the survivor *s2* captured by the downward-facing camera. The artifact was not detected as there is only a small part of the survivor's sleeve visible in the image, which is also degraded by motion blur.

Both vent *v1* and drill *d1* were detected, localized, and correctly labeled. The drill *d4* was incorrectly classified as a backpack, nevertheless, the operator reported the correct class based on the detection image. All three UAVs detected the *d4* drill, but UAV *green* provided the highest accuracy, which is reported in Table 6. In total, four artifact hypotheses arrived to the base station with sufficient information for obtaining a point for the report.

Table 5. Statistics of artifact detection for each deployed UAV from the prize round of the Final Event. The *seen* column yields the number of artifacts that appeared in the image of one of the on-board cameras. If the artifact was detected by the CNN, it is listed in the *detected* column and the detection is shown in Figure 45. Artifacts that were *confirmed* had enough consistent detections to establish a hypothesis. *Confirmed unique* artifacts were not detected by another robot, including UGVs.

	Artifacts			
UAV	**seen**	**detected**	**confirmed**	**confirmed unique**
Red	1	1	1	0
Green	4	3	3	1
Blue	4	4	3	1

Table 6. Unique artifacts detected by lightweight CNN running on-board UAVs in real time. The total error e_{tot} of the artifact position is the sum of the UAV localization drift error e_{loc} and the error of estimating the artifact position e_{est} from the detected bounding box. Artifacts detected by more UAVs are listed only once with values from the most accurate hypothesis among the UAVs. The hypothesis was *Confirmed* when more than four images were associated with it. Some artifacts were correctly detected and localized, but the wrong label was assigned to them. This is documented in the *Correct class* column. Even with a wrong label, the operator could still deduce the correct class by looking at the image sent with the hypothesis. Only one image was sent with each hypothesis, and if it was possible to deduce the correct class, then the image was listed as *Correct image*.

Artifact	**Frames detected**	**Confirmed**	**Correct class**	**Correct image**	e_{loc} **(m)**	e_{est} **(m)**	e_{tot} **(m)**
v1	27	✓	✓	✓	1.94	4.61	3.08
s1	60	✓	×	×	2.93	4.57	2.89
p1	1	×	×	×	-	-	-
d4	11	✓	×	✓	0.77	1.61	1.30
f1	13	✓	✓	✓	0.85	1.33	1.31
d1	9	✓	✓	✓	1.46	2.30	1.55

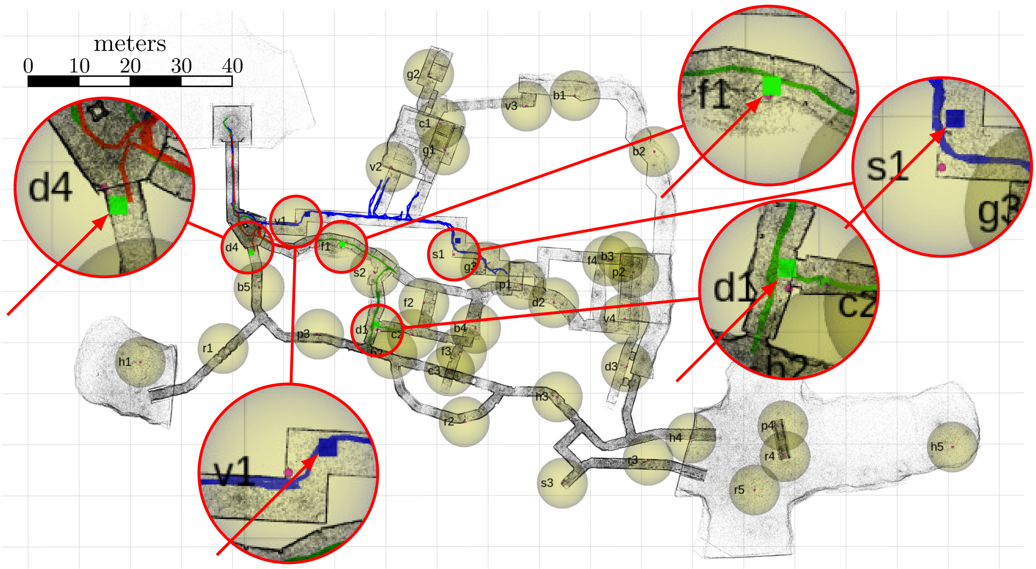


Figure 44. The map of the Final Event course was obtained by the organizers by scanning the course with a laser scanner station. The paths traveled by all three UAVs (*red*, *green*, and *blue*) during the Final Event are depicted by their respective colors. The ground truth positions of artifacts are surrounded by a yellow sphere in order to visualize the 5 m limit for the reported artifact to be counted as a point in the competition. The five artifacts that were detected and localized within this 5 m limit are shown as squares colored by the detecting UAV and highlighted in the magnified sections with red arrows.

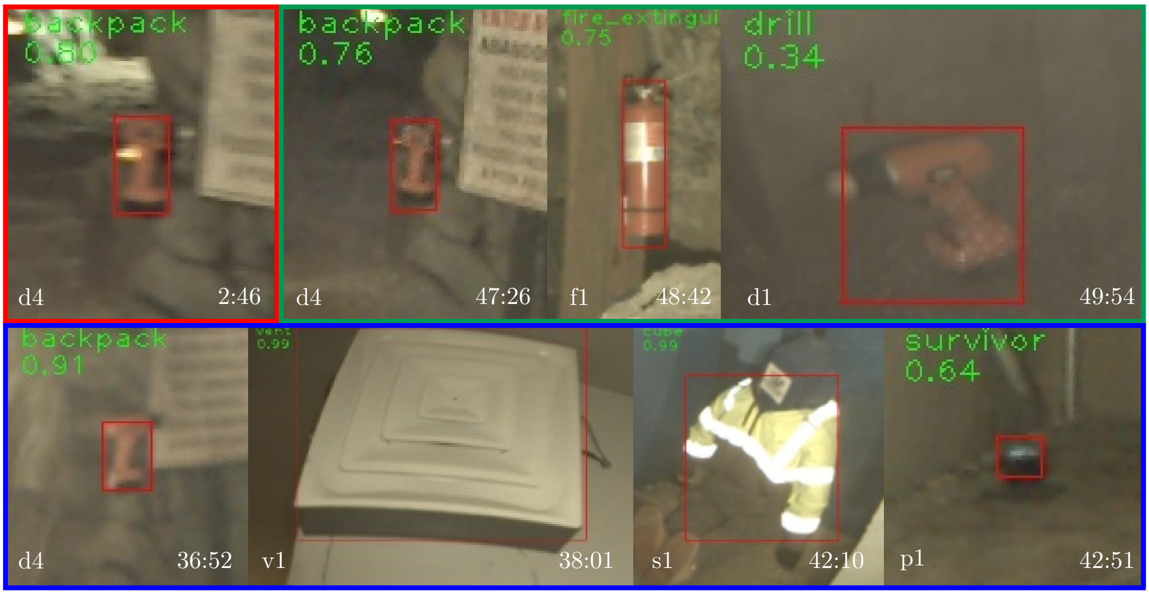


Figure 45. Images of artifacts detected by the UAVs in the Final Event. The color of the rectangle shows which UAV detected the artifact and at what mission time as shown in the bottom right corner.

14.3. DARPA SubT Final Event Virtual Track

In parallel to the Systems Track, the competition was also running in the simulated form of the Virtual Track. The teams had to submit a solution consisting of docker images of a robotic team put together within a limited budget to buy the robots and their sensory packages.

The Systems Track included a single run (with two preliminary rounds) conducted in a single world and was therefore focused on the reliability of the robots, which had to overcome challenging terrain with narrow passages and adverse conditions for perception. On the other hand, the virtual teams were deployed three times in each of the eight worlds, ranging from vast models of artificially created environments to scanned courses from the previous events, including the Final Event course. Moreover, in the Virtual Track, the whole mission must be fully autonomous and no human interventions are possible. The purpose of the virtual event was to evaluate the high-level planning, cooperation, decision-making, and efficient coverage of the large worlds. As the cooperative searching strategy is one of the core contributions of this work, we have presented the results from the virtual course here as most of the worlds allowed for efficient deployment and cooperation of the multirobot teams.

14.3.1. Differences from the Systems Track

The simulation model of the IMU provides much better data compared to the real sensor with the same parameters. Thus is due to the measurements in the simulation not being corrupted by propeller-induced vibrations, wind gusts, or saturation, as well as having the IMU rigidly attached to the UAV body with known extrinsic parameters. The higher quality of the simulated data allows for the use of LiDAR-inertial odometry. In addition to the LiDAR, it also relies on the IMU preintegration in its optimization process, thus providing a smooth and drift-free position estimate, even when there are few geometrically rich features present. Specifically, the LIO-SAM (Shan et al., 2020) algorithm was chosen for its low drift and high precision over the A-LOAM deployed in the Systems Track. Both algorithms are detailed in Section 6.

The computation power available for artifact detection in the Virtual Track was not constrained by the weight of the onboard computation hardware as was the case in the Systems Track. As a result, compromises of the performance/weight ratio had to be made on the detector in the Systems Track, as reported in Section 10.

Reporting of the found artifacts is handled by the operator in the Systems Track, which is not possible in the fully autonomous Virtual Track. A virtual artifact reporter algorithm was developed to gather artifact hypotheses from all robots and decide which hypotheses are the most likely to score a point (described in detail in Section 10.4).

The control interface of the simulated UAV was also different from the real one. While the FCU of the real UAV accepted attitude rate commands generated by the Special Euclidean group of dimension 3 (SE(3)) controller, the simulated UAV was controlled on a higher level by velocity commands. This did not allow for precise control of the UAV motion, as was the case for the low-level attitude rate control.

The deployment sequence of individual robots in the Systems Track could be decided by the operator based on the requirements of locomotion modality, dynamics, and sensory payload during the progress of the mission. In contrast, the sequence in the Virtual Track was fixed before the start of the run.

LandMap introduced in Section 7.5 was not used in the Virtual Track where the UAV was not destroyed even after a rough landing. As long as the UAV landed in the communication range of the network it could send its hypotheses to the base station and further retranslate messages from/to other robots.

14.3.2. Virtual Track results

In the virtual deployment, our team consisted of five UAVs and two UGVs. The UAVs were the superior platform in the Virtual Track due to their greater movement speed, smaller form-factor, and better mobility to fly over terrain untraversable by the UGVs. We deployed two UGVs to build a communication network consisting of breadcrumbs dropped at the edges of the wireless signal range. This allowed for the UAVs to maximize the time for searching for artifacts as they could return to the nearest breadcrumb instead of to the base station back at the staging area. Both UGVs were deployed at the start of the run. The deployment times and exploration strategies of individual UAVs are listed in Table 7. Our solution achieved 2$^{\text{nd}}$ place with a total of 215 scored points. Table 8 summarizes the points scored by the top three teams on each world of the Virtual Track (Figure 46). The lower number of points on worlds 4, 5, 6, and 8 can be explained by the fact that these worlds were not made of the tiles that were used in the qualification and practice worlds. The details on traveled distance and collected hypotheses by particular UAVs during all runs of the SubT Virtual Finals are provided in Figure 47 and Figure 48, respectively.

Table 7. The times of deployment and assigned strategies from Section 8.2 in the Virtual Track. The second UAV was scheduled to take off after the first UAV returned to communication range so that it can take advantage of the LTVMap of the first UAV. Both DEI and Greedy strategies were used as DEI guarantees covering dead-end corridors at the cost of lower average velocity and lower total surface covered. The first UAV used DEI so that the rest of the team did not need to return to where the first UAV already had been. The next two UAVs maximize the searched volume with the Greedy strategy and the last two UAVs cover any missed surfaces with DEI.

UAV	1	2	3	4	5
Start (s)	60	1560	1680	1800	1920
Strategy	DEI	Greedy	Greedy	DEI	DEI

Table 8. The score achieved by the top three teams on each world of the Virtual Track. The reported values are the sums of three runs on each world.

World	1	2	3	4	5	6	7	8	total
Dynamo	21	52	48	18	15	11	44	14	223
CTU-CRAS-NORLAB	31	39	45	16	18	13	36	17	215
Coordinated Robotics	44	41	27	23	17	14	26	20	212

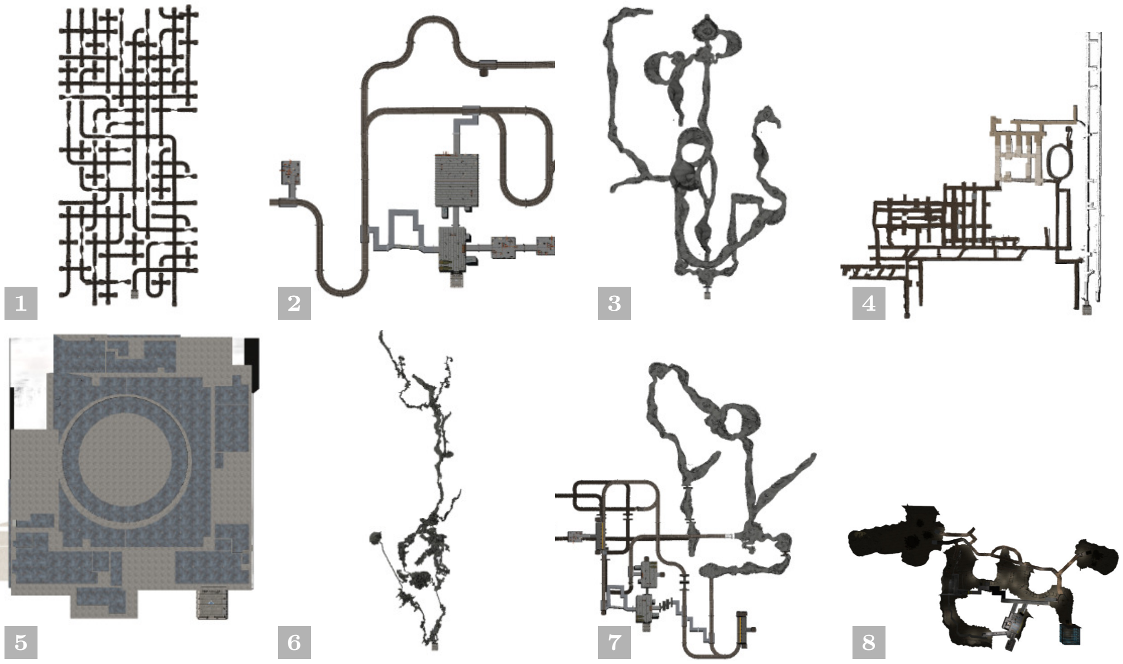


Figure 46. All eight worlds used in the Virtual Track of the DARPA SubT Finals. The worlds 1, 2, 3, and 7 are built from tiles that were used in the preliminary and practice rounds. World 4 is the model of the NIOSH research mine, where the tunnel circuit was held. Similarly, world 5 corresponds to the model of the location of the urban circuit—the unfinished Satsop nuclear power plant. World 6 is a model of a narrow cave system. World 8 is modeled based on the Systems Track Finals.

15. Lessons learned and future work

In this section, we present our view on the state of the S&R UAVs, the lessons learned, which problems are solved, and what areas require more research to achieve reliable performance suitable for deployment as a tool for assisting rescue workers. These findings were collected throughout the preparation for as well as during the DARPA SubT Competition, which aimed to push the state of the art of S&R robotics. Furthermore, this discussion should be of some interest to the community as we highlight aspects that could be explored in future research and development. In general, most of the individual subproblems, such as localization, mapping, detection, and communication, are solved to the point of being capable of performing an autonomous mission in extremely challenging conditions. The developed algorithms are now used in actual field deployment instead of just laboratories and simulations, which introduces disturbances, noise, dust, and other detrimental effects that negatively impact the algorithms' performance and reliability. It is essential to focus on the reliability of the employed methods to make the UAVs a valuable asset to the S&R team.

The role of the aerial robot in a heterogeneous S&R robotic team is a quickly deployable agent that can provide swift situation awareness, environment type, and topology information that allows for informed decision-making about the rest of the mission. Furthermore, areas such as caves, collapsed buildings or high openings can often be reached only by UAVs. On the other hand, ground robots have the advantage of higher payload capacity, which results in improved perception capabilities compared to UAVs.

The localization method based on 3D LiDAR provides precise position estimates, even under severe degradation by dust. However, as proved by the UAV *blue*, the estimate can begin to drift when the solved optimization is ill-conditioned due to low-variance geometry, typically in long corridors with straight walls. The unpredictable nature of subterranean environments requires a localization method that is reliable and drift-free under arbitrary conditions. Solutions based

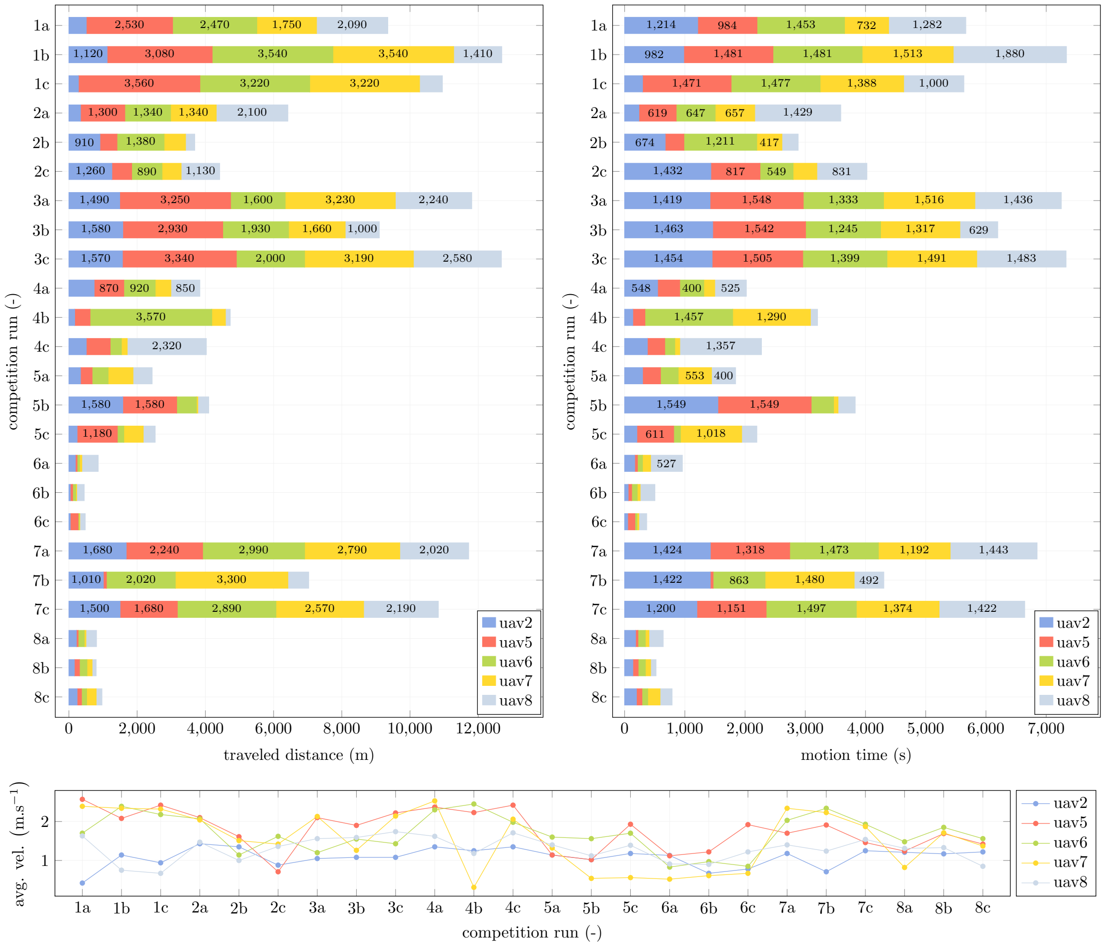


Figure 47. Overall traveled distance, time of active motion, and average velocity of particular UAVs in all runs of the SubT Virtual Finals. The maximum traveled distance throughout all runs was achieved by UAV5 in run 1c (3560 m). The maximum active time was achieved by UAV2 in run 5b (1539 s). The presented average velocity incorporates the entire flight, including hovering states.

on detecting geometrical degeneracy, and multimodal fusion of LiDAR and visual methods were described in Section 2.2. The results seem promising but due to high unpredictability and challenges of subterranean environments more research in localization algorithms is still required for truly robust pose estimation in arbitrary conditions.

In addition to map drift caused by errors in the localization, the volumetric occupancy grid did not contain the smaller obstacles like ropes, cables, and thin poles, which led to the collision of UAV *green* as seen in Figure 42b. Although some LiDAR rays hit these thin obstacles, the occupied cells generated by these rays were often changed to free when multiple rays that passed through these cells hit the wall behind them. As a result, the navigation pipeline planned a path through these cells that appeared free, but contained a thin metal pole, causing a collision. The ability to traverse narrow passages is also impaired since the passages appear narrower than they really are due to grid discretization. We propose to locally increase the resolution of the grid of DenseMap on demand to approximate the free space more accurately, while keeping the scan integration times bounded. This approach is however only a partial solution as the need for a more granular resolution might not always be reliably detected. Consequently, the need arises for a flexible map that is not bound by fixed cell size, similarly to the SphereMap, possibly based on surfel mapping as seen in (Behley and Stachniss, 2018), or based on GMM (O'Meadhra et al., 2018).

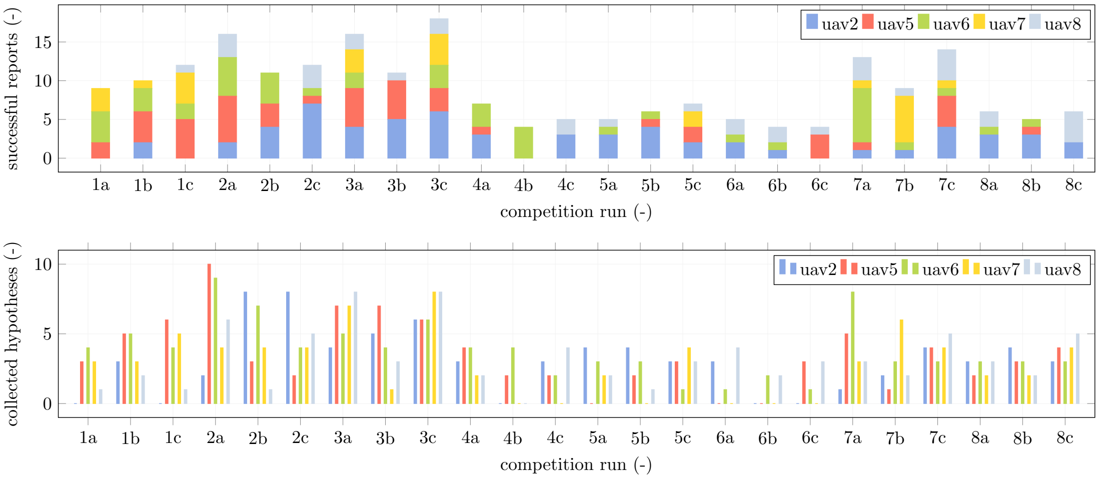


Figure 48. Distribution of successful reports among the UAVs in particular runs of SubT Virtual Finals (top) and the number of valid hypotheses collected throughout particular runs by individual robots (bottom). The number of successful reports of individual robots is mostly influenced by the ordering of robots and their delayed starts in the mission.

Related to the narrow passage traversal was also the decision to not include prop guards in the platform design. With the experience from Urban Circuit (Kratky et al., 2021a) where our platform featured prop guards, we decided against prop guards in the Final Event as they further increase the size and mass of the UAV. The advantage of the prop guards is uncertain as they do not automatically allow the UAV to continue operation after a collision. The control and localization software needs to be designed to handle these dampened collisions and even then it is not guaranteed that the UAV will continue in flight as the collision is caused by a failure of a module (perception failure to detect an obstacle, localization error, etc.), which cannot be solved by prop guards. Nevertheless, platforms with prop guards were deployed successfully by other teams [e.g., (Scherer et al., 2022; Agha et al., 2021)] so a consensus on this design choice has not been reached yet.

We experienced a surprising issue when our UAV equipped with the Ouster OS0-128 LiDAR was passing around a UGV with LeiShen C16 LiDAR. The rays emitted by the LeiShen corrupted some of the Ouster measurements, which manifested as points in random distance within the FOV of the LiDAR. These false positives were not filtered out by the intensity filter from Section 5.2, because the intensities fall into the same range of values as true positives. As a result, the points get integrated into the map, as shown in Figure 49. Nevertheless, the performance of the UAV was not degraded as the navigation pipeline is robust to such sparse noise. This experience highlights the importance of testing the compatibility of robotic platforms deployed in heterogeneous teams.

The flight time of the UAV over 20 min was achieved as the payload was limited only to crucial components. However, the presence of only a single computational unit without CNN acceleration or dedicated GPU led to compromises in the artifact detection CNN. Large-size models such as YOLOv3 (Redmon and Farhadi, 2018) were too slow for achieving satisfactory frame rates on the CPU, so lighter models had to be used. As explained in Section 10, the lightweight MobileNetV2 CNN allowed for lightweight models (7 MB) that could fit into the cache of the CPU. Furthermore, the OpenVino framework supports accelerating the CNN on the GPU integrated with the CPU, which helped to achieve sufficient frame rates. Although the lightweight model successfully detected all artifact types, the labeling was not very reliable and many false positives were detected. This impacted the artifact localization process, as the false positives were fused into the artifact hypotheses, which shifted the estimate further from the true position. Also, the images of these false positives were sometimes sent as the representative image of the hypothesis. Thus the operator could not correctly decide the artifact class when the label produced by the CNN was incorrect. When

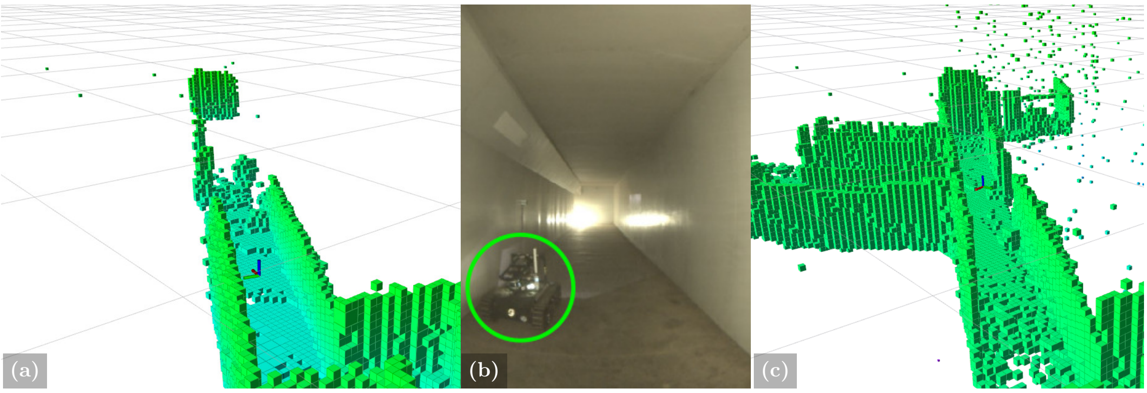


Figure 49. DenseMap before (a) approaching the UGV with LeiShen C16 LiDAR (b) and (c) when it gets corrupted by random points in the FOV of the LiDAR mounted on the UAV after flying in close vicinity (≈ 1 m) to the UGV. Notice, a few false positives were integrated into the map even when the UAV was 8 m away from the UGV (a).

payload capacity prevents the use of more capable hardware, the issue must be compensated by the sensing strategy. In contrast to UGVs the mobility of UAVs allows reaching closer to the artifact to verify the detection. Approaches of perception-driven navigation can improve the performance of the lightweight detector by planning a trajectory to inspect the artifact from a closer distance and other angles after the initial detection.

Although our platform is quite compact (500 mm without propellers), it could not pass through all of the narrow passages, even during the postevent testing. Apart from the discrete map and conservatively set distance from obstacles (see Table 4), the size of the UAV prevented flying through some of the narrower passages of the circuit. As even smaller passages are to be expected during deployment of robots in real S&R scenarios, the UAV platforms should be further miniaturized to allow safe traversal of narrow passages. Deployment of visually localized UAVs could decrease the size significantly but the capabilities of visual navigation pipelines still underperform compared to the LiDAR solutions, which was the preferred approach of most teams. A possible workaround that compensates for the lower flight time of smaller platforms is the marsupial deployment (Lindqvist et al., 2022; De Petris et al., 2022). When such miniaturization is not possible due to, e.g., insufficient payload of smaller platforms, a heterogeneous aerial team consisting of both large and small platforms can be deployed, In such case, the large platform carrying a LiDAR can command and send position corrections to smaller visually localized UAV that can inspect tight narrow passages that are unreachable by the large UAV (Pritzl et al., 2022b).

A mutual collision avoidance module is a necessity for any application where multiple robots share the same workspace. The developed priority-based module uses the already shared information about the robots' positions when communication is available, as it should since the risk of collision arises when robots are in close proximity. This module prevented collisions in the Virtual Track, where despite the vastness of most of the worlds, the collisions happened often in the practice runs before implementing the collision avoidance. We decided against using the collision avoidance module in the Systems Track. This was done as the robots could easily become deadlocked in tight corridors and also due to the collision probability being reasonably low because of the delay between each UAVs launch. Additionally, the operator could override the full autonomy to prevent collision, if necessary. Nevertheless, the UAV *red* collided with a Spot UGV shortly after the start of the run, which could have been prevented if collision avoidance was enabled. A deadlock-free solution based on agent theory approaches can be devised for situations when communication is available, and behavior prediction methods can provide a backup when communication is not possible.

Even though the organizers did a great job at providing a realistic simulation environment for the Virtual Track, many phenomena, unexpected situations, and issues from the real world are not present in the simulation. Moreover, the rules of the competition are different for the two tracks. For example, the absence of a human operator in the virtual team changes the approach drastically as all decision-making needs to be automated. For details about the differences between the Systems and Virtual tracks see Section 14.3.1. As a team that participated in both Systems and Virtual tracks, we want to list the greatest hurdles encountered in the simulation to real-world transfer:

- **Degraded sensor output.** The output of real-world sensors is corrupted by various negative effects. On the other hand, the imperfections in simulated sensors are typically modeled only by an additive noise, which most algorithms can cope with by smoothing or filtering. The performance of such algorithms severely deteriorates with input data degraded by the harsh conditions of underground environments. In the case of laser-based sensors, rays get reflected by small airborne particles such as dust, smoke, and fog to produce false measurements. Although the fog was modeled in the simulation, the distribution of fog points in the point cloud did not coincide with the distribution in the real world, and thus we implemented different approaches for Virtual and Systems tracks, which are detailed in Section 5.2. Cameras in addition to the airborne particles suffer from insufficient illumination, high-contrast scenes, motion blur, and rolling shutter aliasing. Using neural networks for object detection proved to be robust to these effects when trained on datasets with similar data.
- **Environment scale.** The scale of the environment was much smaller in the Systems Track than in the Virtual Track. Most of the simulation worlds spanned several-kilometers-long corridors with vast caves to challenge the cooperative exploration abilities of the teams. The vastness and openness of the worlds favored fast flights to cover as much space as possible during flight time. In contrast, the Systems Track was narrow from the beginning of the course (see the cross-section distribution in Table 2) and the UAV was closer than 1 m from obstacles for most of the flight (see the distance to obstacle plot in Figure 40). To minimize the chance of collision, the velocity of the UAV was constrained to $1\,\mathrm{m\,s^{-1}}$ instead of $5\,\mathrm{m\,s^{-1}}$ in simulation and the control was tuned for low error by commanding the UAV in attitude rates instead of linear velocities.
- **Safety.** In simulation, the robots cannot harm anyone and to perform another run after a collision it is sufficient to restart the simulation. Contrary to that, in the real world, special care must be taken to make the robots, especially fast aerial robots with quickly spinning propellers, safe for the environment, operators, and any other humans in the vicinity. To assure maximum safety during takeoff, status checks are performed automatically but then the safety operator has to approve the takeoff by toggling a switch on the RC as described in Section 11.1. During the flight, health checks of the rate of crucial data streams are performed, control errors are monitored, and innovation of state estimation corrections are analyzed. When any of the monitored values exceeds a critical threshold, an emergency landing is initiated to minimize the damage to the platform. A technique based on data from the downward-facing depth camera was developed to assure safe regular (not emergency) landing on planar low-slope surfaces stored in LandMap (Section 7.5).
- **Decision-making in artifact reporting.** Due to the limited payload of the UAV, a lightweight CNN was used for artifact detection, which forced us to choose a compromise between precision and recall. Having a human operator to verify the artifact hypotheses allowed us to maximize recall to not miss any artifact at the cost of a higher false positive count. A correctly detected but misclassified artifact could still score a point if the operator correctly deduced the class from the image (see Table 6). This would not have been possible using the autonomous arbiter and thus the flexibility of a human operator overperforms the autonomous arbiter, which optimizes a fixed criterion when reporting an artifact hypothesis.
- **Computational resources** The simulated run in Virtual Track was running only at a fraction of real time and thus the employed algorithms had more computation time available. In the Systems Track, the developed algorithms had to run on the onboard processing units in real

time. Thus, for the simulation to real-world transfer, the algorithms need to be optimized, critical systems prioritized, and often a compromise between accuracy and computation time has to be found. We discuss all modifications and optimizations in Section 13.

16. Conclusion

This paper has presented the complex UAV system deployed in the final round of the DARPA SubT Challenge after 3 years of development and testing in numerous real world demanding environments (including gold mine, coal mine, abandoned nuclear power plant, caverns, military fortress, natural caves, old factory hall, subway station, etc.). Based on these unique opportunities and experience, we have designed both the hardware UAV platform and the multi-UAV software with a focus on the exploration of such vast, complicated, and varying environments.

In the Systems Track of DARPA SubT Challenge, three UAVs were deployed alongside ground robots into the competition course consisting of a heterogeneous environment of the tunnel, urban, and cave sections, where the aerial team detected and localized four artifacts and traveled 492 m in total. The austere conditions of the circuit, such as narrow passages, dust, featureless corridors, and dynamic obstacles, tested the reliability of the system as a whole, including the hardware design of a compact platform with a considerable flight time of 25 min. Most of the testing was realized in environments where it performed exceptionally well, including a former brewery where the UAV had to explore an abandoned building with partially collapsed floor and ceiling, or during the exploration of Byci Skala (Bull Rock Cave) in the Moravian Karst cavern system. Compared to ground robots the UAVs could search a larger volume of space because they could easily fly over any encountered problematic terrain such as mud, water, and rubble and thus had an advantage in the exploration of unknown terrains with unexpected obstacles. Furthermore, the worlds of the Virtual Track of the competition were also very large; even with our UAV possessing a 25 min flight time and fast dynamics, they were not able to reach the furthest parts of some worlds. Although our system was designed primarily for these large-scale environments, its performance in the challengingly tight corridors of the prize round was also impressive. The difficulty of UAV deployment in such adverse environments motivated numerous achievements beyond the state of the art that are summarized in this paper. Many lessons were learned in the process that could facilitate and support designing complex robotic systems in similar applications in the future.

A larger team of five aerial robots was deployed in the Virtual Track, alongside two UGVs. By employing the proposed cooperative exploration strategies based on topological map sharing, the exploration effort of our team was spread out over a wider area. Together with dynamic flight and reliable artifact detection/localization, this helped to achieve the 2$^{\text{nd}}$ place with 215 scored points. Moreover, seven out of the nine participating teams used our X500 UAV, which was modeled according to the specification of the physical platform thanks to its long flight time, a wide array of sensors, modest size, and reasonable price.

Based on the successful deployment in the DARPA SubT, which focused on providing challenging conditions typically encountered during rescue missions in underground environments, we conclude that the presented UAV system is a valuable addition to teams of first responders, as it can provide situational awareness and even find survivors after a catastrophe without risking the lives of rescuers in dangerous environments.

Acknowledgments

We would like to thank the members of the CTU-CRAS-NORLAB team who participated in the design and development of the UAV and UGV hardware platforms, software development, simulations, testing and general support. Namely: Ruslan Agishev, Afzal Ahmad, Teymur Azayev, Jan Bayer, Tommy Bouchard-Lebrun, Petr Čížek, Simon-Pierre Deschênes, Jan Faigl, Olivier Gamache, Alexandre Guénette, Bedřich Himmel, Jakub Janoušek, Tomáš Krajník, Vladimír Kubelka, Denis Ouellet, Tomáš Petříček, François Pomerleau, Miloš Prágr, Tomáš Rouček, Vojtěch Šalanský, Martin

Škarytka, Vojtěch Spurný, Pavel Stoudek, Arsen Tkachev, Maxime Vaidis, Volodymyr Zabulskyi, Karel Zimmermann, and Martin Zoula.

This work was partially funded by the Defense Advanced Research Projects Agency (DARPA), by the CTU grant no. SGS20/174/OHK3/3T/13, by the Czech Science Foundation (GAČR) under research project no. 20-29531S, by TAČR project no. FW03010020, by the OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics," by the European Union's Horizon 2020 research and innovation programme AERIAL-CORE under grant agreement no. 871479, and by the NAKI II project no. DG18P02OVV069.

ORCID

Matěj Petrlík https://orcid.org/0000-0002-5337-9558
Pavel Petráček https://orcid.org/0000-0002-0887-9430
Vít Krátký https://orcid.org/0000-0002-1914-742X
Tomáš Musil https://orcid.org/0000-0002-9421-6544
Yurii Stasinchuk https://orcid.org/0000-0002-2197-1442
Matouš Vrba https://orcid.org/0000-0002-4823-8291
Tomáš Báča https://orcid.org/0000-0001-9649-8277
Daniel Heřt https://orcid.org/0000-0003-1637-6806
Martin Pecka https://orcid.org/0000-0002-0815-304X
Tomáš Svoboda https://orcid.org/0000-0002-7184-1785
Martin Saska https://orcid.org/0000-0001-7106-3816

References

Agha, A., Otsu, K., Morrell, B., Fan, D. D., Thakker, R., Santamaria-Navarro, A., Kim, S., Bouman, A., Lei, X., Edlund, J. A., Ginting, M. F., Ebadi, K., Anderson, M., Pailevanian, T., Terry, E., Wolf, M. T., Tagliabue, A., Vaquero, T. S., Palieri, M., et al. (2021). Nebula: Quest for robotic autonomy in challenging environments; TEAM costar at the DARPA subterranean challenge. *CoRR*, abs/2103.11470.

Ahmad, A., Walter, V., Petracek, P., Petrlik, M., Baca, T., Zaitlik, D., and Saska, M. (2021). Autonomous aerial swarming in gnss-denied environments with high obstacle density. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 570–576. IEEE.

Ahmad, S., Sunberg, Z. N., and Humbert, J. S. (2021). End-to-end probabilistic depth perception and 3d obstacle avoidance using pomdp. *Journal of Intelligent & Robotic Systems*, 103(2):1–18.

Alismail, H., Kaess, M., Browning, B., and Lucey, S. (2016). Direct visual odometry in low light using binary descriptors. *IEEE Robotics and Automation Letters*, 2(2):444–451.

Alotaibi, E. T., Alqefari, S. S., and Koubaa, A. (2019). Lsar: Multi-uav collaboration for search and rescue missions. *IEEE Access*, 7:55817–55832.

Baca, T., Hert, D., Loianno, G., Saska, M., and Kumar, V. (2018). Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–8. IEEE.

Baca, T., Loianno, G., and Saska, M. (2016). Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles. In *2016 IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 992–997.

Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M. (2021). The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 102(26):1–28.

Bayer, J. and Faigl, J. (2020). Handheld localization device for indoor environments. In *2020 4th International Conference on Automation, Control and Robots (ICACR)*, pages 60–64.

Behley, J. and Stachniss, C. (2018). Efficient surfel-based slam using 3d laser range data in urban environments. In *Robotics: Science and Systems*, volume 2018, page 59.

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016). Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE.

Blochliger, F., Fehr, M., Dymczyk, M., Schneider, T., and Siegwart, R. Y. (2018). Topomap: Topological mapping and navigation based on visual slam maps. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9.

Bosse, M., Zlot, R., and Flick, P. (2012). Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21:376 – 386.

Burri, M., Oleynikova, H., , Achtelik, M. W., and Siegwart, R. (2015). Real-Time Visual-Inertial Mapping, Re-localization and Planning Onboard MAVs in Unknown Environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332.

Chang, Y., Ebadi, K., Denniston, C. E., Ginting, M. F., Rosinol, A., Reinke, A., Palieri, M., Shi, J., Chatterjee, A., Morrell, B., Agha-mohammadi, A.-a., and Carlone, L. (2022). Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments. *IEEE Robotics and Automation Letters*, 7(4):9175–9182.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. (2019). MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Chen-Lung, L., Jui-Te, H., Huang, C.-I., Zi-Yan, L., Chao-Chun, H., Yu-Yen, H., Siao-Cing, H., Po-Kai, C., Zu Lin, E., Po-Jui, H., Po-Lin, L., Bo-Hui, W., Lai-Sum, Y., Sheng-Wei, H., MingSian R., B., and Hsueh-Cheng, W. (2022). A heterogeneous unmanned ground vehicle and blimp robot team for search and rescue using data-driven autonomy and communication-aware navigation. *Field Robotics*, 2:557–594.

Dang, T., Khattak, S., Mascarich, F., and Alexis, K. (2019a). Explore locally, plan globally: A path planning framework for autonomous robotic exploration in subterranean environments. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 9–16. IEEE.

Dang, T., Mascarich, F., Khattak, S., Nguyen, H., Nguyen, H., Hirsh, S., Reinhart, R., Papachristos, C., and Alexis, K. (2020a). Autonomous search for underground mine rescue using aerial robots. In *2020 IEEE Aerospace Conference*, pages 1–8. IEEE.

Dang, T., Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2019b). Graph-based path planning for autonomous robotic exploration in subterranean environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3105–3112.

Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., and Hutter, M. (2020b). Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388.

De Petris, P., Khattak, S., Dharmadhikari, M., Waibel, G., Nguyen, H., Montenegro, M., Khedekar, N., Alexis, K., and Hutter, M. (2022). Marsupial walking-and-flying robotic deployment for collaborative exploration of unknown environments. *arXiv preprint arXiv:2205.05477*.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., et al. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191.

Denniston, C. E., Chang, Y., Reinke, A., Ebadi, K., Sukhatme, G. S., Carlone, L., Morrell, B., and Agha-mohammadi, A.-a. (2022). Loop closure prioritization for efficient and scalable multi-robot slam. *IEEE Robotics and Automation Letters*, 7(4):9651–9658.

Ebadi, K., Bernreiter, L., Biggie, H., Catt, G., Chang, Y., Chatterjee, A., Denniston, C. E., Deschenes, S.-P., Harlow, K., Khattak, S., et al. (2022). Present and future of slam in extreme underground environments. *arXiv preprint arXiv:2208.01787*.

Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., et al. (2020). Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually degraded subterranean environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 80–86. IEEE.

Ebadi, K., Palieri, M., Wood, S., Padgett, C., and Agha-mohammadi, A.-a. (2021). Dare-slam: Degeneracy-aware and resilient loop closing in perceptually degraded environments. *Journal of Intelligent & Robotic Systems*, 102(1):1–25.

Fabris, A., Kirchgeorg, S., and Mintchev, S. (2021). A soft drone with multi-modal mobility for the exploration of confined spaces. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 48–54.

Fan, D. D., Otsu, K., Kubo, Y., Dixit, A., Burdick, J., and Agha-Mohammadi, A.-A. (2021). Step: Stochastic traversability evaluation and planning for safe off-road navigation. *arXiv preprint arXiv:2103.02828*.

Ginting, M. F., Otsu, K., Edlund, J. A., Gao, J., and Agha-Mohammadi, A.-A. (2021). Chord: Distributed data-sharing via hybrid ros 1 and 2 for multi-robot exploration of large-scale complex environments. *IEEE Robotics and Automation Letters*, 6(3):5064–5071.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Harabor, D. and Grastien, A. (2011). Online graph pruning for pathfinding on grid maps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1114–1119.

Hert, D., Baca, T., Petracek, P., Kratky, V., Spurny, V., Petrlik, M., Matous, V., Zaitlik, D., Stoudek, P., Walter, V., Stepan, P., Horyna, J., Pritzl, V., Silano, G., Bonilla Licea, D., Stibinger, P., Penicka, R., Nascimento, T., and Saska, M. (2022). MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE.

Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1271–1278. IEEE.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34:189–206.

Huang, Y.-W., Lu, C.-L., Chen, K.-L., Ser, P.-S., Huang, J.-T., Shen, Y.-C., Chen, P.-W., Chang, P.-K., Lee, S.-C., and Wang, H.-C. (2019). Duckiefloat: a collision-tolerant resource-constrained blimp for long-term autonomy in subterranean environments. *arXiv preprint arXiv:1910.14275*.

Hudson, N., Talbot, F., Cox, M., Williams, J. L., Hines, T., Pitt, A., Wood, B., Frousheger, D., Surdo, K. L., Molnar, T., Steindl, R., Wildie, M., Sa, I., Kottege, N., Stepanas, K., Hernandez, E., Catt, G., Docherty, W., Tidd, B., Tam, B., Murrell, S., Bessell, M., Hanson, L., , et al. (2022). Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO data61's approach to the DARPA subterranean challenge. *Field Robotics*, 2:595–636.

Kasper, M., McGuire, S., and Heckman, C. (2019). A benchmark for visual-inertial odometry systems employing onboard illumination. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5256–5263. IEEE.

Khattak, S., Mascarich, F., Dang, T., Papachristos, C., and Alexis, K. (2019). Robust thermal-inertial localization for aerial robots: A case for direct methods. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1061–1068. IEEE.

Khattak, S., Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020). Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1024–1029.

Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE.

Koval, A., Karlsson, S., Mansouri, S. S., Kanellakis, C., Tevetzidis, I., Haluska, J., Agha-mohammadi, A.-a., and Nikolakopoulos, G. (2022). Dataset collection from a subt environment. *Robotics and Autonomous Systems*, 155:104168.

Kramer, A., Kasper, M., and Heckman, C. (2021). Vi-slam for subterranean environments. In *Field and Service Robotics*, pages 159–172. Springer.

Kratky, V., Alcantara, A., Capitan, J., Stepan, P., Saska, M., and Ollero, A. (2021). Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles. *IEEE Robotics and Automation Letters*, 6(4):7580–7587.

Kratky, V., Petracek, P., Baca, T., and Saska, M. (2021a). An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments. *Journal of Field Robotics*, 38(8):1036–1058.

Kratky, V., Petracek, P., Nascimento, T., Cadilova, M., Skobrtal, M., Stoudek, P., and Saska, M. (2021b). Safe documentation of historical monuments by an autonomous unmanned aerial vehicle. *ISPRS International Journal of Geo-Information*, 10(11):738/1–16.

Kulkarni, M., Dharmadhikari, M., Tranzatto, M., Zimmermann, S., Reijgwart, V., De Petris, P., Nguyen, H., Khedekar, N., Papachristos, C., Ott, L., et al. (2021). Autonomous teamed exploration of subterranean environments using legged and aerial robots. *arXiv preprint arXiv:2111.06482*.

Lajoie, P.-Y., Ramtoula, B., Chang, Y., Carlone, L., and Beltrame, G. (2020). Door-slam: Distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, 5(2):1656–1663.

Lee, T. et al. (2010). Geometric tracking control of a quadrotor UAV on SE(3). In *2010 IEEE Conference on Decision and Control*, pages 5420–5425. IEEE.

Lei, X., Kim, T., Marchal, N., Pastor, D., Ridge, B., Scholler, F., Terry, E., Chavez, F., Touma, T., Otsu, K., et al. (2022). Early recall, late precision: Multi-robot semantic object mapping under operational constraints in perceptually degraded environments. *arXiv preprint arXiv:2206.10062*.

Lindqvist, B., Kanellakis, C., Mansouri, S. S., Agha-mohammadi, A.-a., and Nikolakopoulos, G. (2021). Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations. *arXiv preprint arXiv:2108.13105*.

Lindqvist, B., Karlsson, S., Koval, A., Tevetzidis, I., Haluska, J., Kanellakis, C., Agha-mohammadi, A.-a., and Nikolakopoulos, G. (2022). Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue. *Robotics and Autonomous Systems*, 154:104134.

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts.

Lu, C. X., Rosa, S., Zhao, P., Wang, B., Chen, C., Stankovic, J. A., Trigoni, N., and Markham, A. (2020). See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 14–27.

Martinez-Rozas, S., Rey, R., Alejo, D., Acedo, D., Cobano, J. A., Rodriguez-Ramos, A., Campoy, P., Merino, L., and Caballero, F. (2022). An aerial/ground robot team for autonomous firefighting in urban gnss-denied scenarios. *Field Robotics*, 2:241–273.

Moniruzzaman, M., Rassau, A., Chai, D., and Islam, S. M. S. (2022). Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey. *Robotics and Autonomous Systems*, 150:103973.

Murphy, R. R., Kravitz, J., Stover, S. L., and Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *IEEE Robotics & Automation Magazine*, 16(2):91–103.

Museth, K. (2013). Vdb: High-resolution sparse volumes with dynamic topology. *ACM transactions on graphics (TOG)*, 32(3):1–22.

Musil, T., Petrlik, M., and Saska, M. (2022). Spheremap: Dynamic multi-layer graph structure for rapid safety-aware uav planning. *IEEE Robotics and Automation Letters*, 7(4):11007–11014.

Ohradzansky, M. T., Rush, E. R., Riley, D. G., Mills, A. B., Ahmad, S., McGuire, S., Biggie, H., Harlow, K., Miles, M. J., Frew, E. W., et al. (2021). Multi-agent autonomy: Advancements and challenges in subterranean exploration. *arXiv preprint arXiv:2110.04390*.

Orekhov, V. and Chung, T. (2022). The darpa subterranean challenge: A synopsis of the circuits stage. *Field Robotics*, 2:735–747.

O'Meadhra, C., Tabib, W., and Michael, N. (2018). Variable resolution occupancy mapping using gaussian mixture models. *IEEE Robotics and Automation Letters*, 4(2):2015–2022.

Palieri, M., Morrell, B., Thakur, A., Ebadi, K., Nash, J., Chatterjee, A., Kanellakis, C., Carlone, L., Guaragnella, C., and Agha-mohammadi, A.-a. (2020). Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robotics and Automation Letters*, 6(2):421–428.

Papachristos, C., Khattak, S., and Alexis, K. (2017). Uncertainty-aware receding horizon exploration and mapping using aerial robots. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4568–4575. IEEE.

Papachristos, C., Khattak, S., Mascarich, F., and Alexis, K. (2019a). Autonomous navigation and mapping in underground mines using aerial robots. In *2019 IEEE Aerospace Conference*, pages 1–8. IEEE.

Papachristos, C., Mascarich, F., Khattak, S., Dang, T., and Alexis, K. (2019b). Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning. *Autonomous Robots*, 43(8):2131–2161.

Petracek, P., Kratky, V., Petrlik, M., Baca, T., Kratochvil, R., and Saska, M. (2021). Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles. *IEEE Robotics and Automation Letters*, 6(4):7596–7603.

Petrlik, M., Baca, T., Hert, D., Vrba, M., Krajnik, T., and Saska, M. (2020). A Robust UAV System for Operations in a Constrained Environment. *IEEE Robotics and Automation Letters*, 5(2):2169–2176.

Petrlik, M., Krajnik, T., and Saska, M. (2021). Lidar-based stabilization, navigation and localization for uavs operating in dark indoor environments. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 243–251. IEEE.

Pritzl, V., Stepan, P., and Saska, M. (2021). Autonomous flying into buildings in a firefighting scenario. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 239–245. IEEE.

Pritzl, V., Vrba, M., Stepan, P., and Saska, M. (2022a). Cooperative navigation and guidance of a micro-scale aerial vehicle by an accompanying uav using 3d lidar relative localization. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 526–535. IEEE.

Pritzl, V., Vrba, M., Stepan, P., and Saska, M. (2022b). Cooperative navigation and guidance of a micro-scale aerial vehicle by an accompanying uav using 3d lidar relative localization. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 526–535. IEEE.

Queralta, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., and Westerlund, T. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8:191617–191643.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Reinke, A., Palieri, M., Morrell, B., Chang, Y., Ebadi, K., Carlone, L., and Agha-Mohammadi, A.-A. (2022). Locus 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping. *IEEE Robotics and Automation Letters*, 7(4):9043–9050.

Reynolds, D. A. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Richter, C., Bry, A., and Roy, N. (2016). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer.

Rogers, J. G., Gregory, J. M., Fink, J., and Stump, E. (2020a). Test your slam! the subt-tunnel dataset and metric for mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 955–961.

Rogers, J. G., Schang, A., Nieto-Granda, C., Ware, J., Carter, J., Fink, J., and Stump, E. (2020b). The darpa subt urban circuit mapping dataset and evaluation metric. In *International Symposium on Experimental Robotics*, pages 391–401. Springer.

Roucek, T., Pecka, M., Cizek, P., Petricek, T., Bayer, J., Salansky, V., Hert, D., Petrlik, M., Baca, T., Spurny, V., et al. (2019). Darpa subterranean challenge: Multi-robotic exploration of underground environments. In *International Conference on Modelling and Simulation for Autonomous Systems*, pages 274–290. Springer.

Roucek, T., Pecka, M., Cizek, P., Petricek, T., Bayer, J., Salansky, V., Hert, D., Petrlik, M., Baca, T., Spurny, V., Pomerleau, F., Kubelka, V., Faigl, J., Zimmermann, K., et al. (2020). Darpa subterranean challenge: Multi-robotic exploration of underground environments. In Mazal, J., Fagiolini, A., and Vasik, P., editors, *Modelling and Simulation for Autonomous Systems*, pages 274–290, Cham. Springer International Publishing.

Saboia, M., Clark, L., Thangavelu, V., Edlund, J. A., Otsu, K., Correa, G. J., Varadharajan, V. S., Santamaria-Navarro, A., Touma, T., Bouman, A., et al. (2022). Achord: Communication-aware multi-robot coordination with intermittent connectivity. *arXiv preprint arXiv:2206.02245*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Santamaria-Navarro, A., Thakker, R., Fan, D. D., Morrell, B., and Agha-mohammadi, A.-a. (2019). Towards resilient autonomous navigation of drones. In *The International Symposium of Robotics Research*, pages 922–937. Springer.

Scherer, S., Agrawal, V., Best, G., Cao, C., Cujic, K., Darnley, R., DeBortoli, R., Dexheimer, E., Drozd, B., Garg, R., et al. (2022). Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robotics*, 2:678–734.

Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., and Guizani, M. (2019). Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *Ieee Access*, 7:48572–48634.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Daniela, R. (2020). Lio-sam: Tightly coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE.

Shin, Y.-S., Park, Y. S., and Kim, A. (2020). Dvl-slam: sparse depth enhanced direct visual-lidar slam. *Autonomous Robots*, 44(2):115–130.

Silano, G., Baca, T., Penicka, R., Liuzza, D., and Saska, M. (2021). Power line inspection tasks with multi-aerial robot systems via signal temporal logic specifications. *IEEE Robotics and Automation Letters*, 6(2):4169–4176.

Smith, L. N. (2015). Cyclical learning rates for training neural networks.

Spurny, V., Pritzl, V., Walter, V., Petrlik, M., Baca, T., Stepan, P., Zaitlik, D., and Saska, M. (2021). Autonomous firefighting inside buildings by an unmanned aerial vehicle. *IEEE Access*, 9:15872–15890.

Tardioli, D., Riazuelo, L., Sicignano, D., Rizzo, C., Lera, F., Villarroel, J. L., and Montano, L. (2019). Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments. *Journal of Field Robotics*, 36(6):1074–1101.

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19(3):46–56.

Tordesillas, J., Lopez, B. T., Everett, M., and How, J. P. (2022). Faster: Fast and safe trajectory planner for navigation in unknown environments. *IEEE Transactions on Robotics*, 38(2):922–938.

Tranzatto, M., Dharmadhikari, M., Bernreiter, L., Camurri, M., Khattak, S., Mascarich, F., Pfreundschuh, P., Wisth, D., Zimmermann, S., Kulkarni, M., et al. (2022a). Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned. *arXiv preprint arXiv:2207.04914*.

Tranzatto, M., Mascarich, F., Bernreiter, L., Godinho, C., Camurri, M., Khattak, S., Dang, T., Reijgwart, V., Loeje, J., Wisth, D., Zimmermann, S., Nguyen, H., Fehr, M., Solanka, L., Buchanan, R., Bjelonic, M., Khedekar, N., Valceschini, M., Jenelten, F., and Alexis, K. (2022b). Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *Field Robotics*, 2:274–324.

Vrba, M., Hert, D., and Saska, M. (2019). Onboard marker-less detection and localization of noncooperating drones for their safe interception by an autonomous aerial system. *IEEE Robotics and Automation Letters*, 4(4):3402–3409.

Walter, V., Novak, T., and Saska, M. (2018). Self-localization of unmanned aerial vehicles based on optical flow in onboard camera images. In *Lecture Notes in Computer Science, vol 10756.*, Cham. Springer International Publishing.

Walter, V., Spurny, V., Petrlik, M., Baca, T., Zaitlik, D., Demkiv, L., , and Saska, M. (2022). Extinguishing real fires by fully autonomous multirotor uavs in the mbzirc 2020 competition. *Field Robotics*, 2:406–436.

Williams, J., Jiang, S., O'Brien, M., Wagner, G., Hernandez, E., Cox, M., Pitt, A., Arkin, R., and Hudson, N. (2020). Online 3d frontier-based ugv and uav exploration using direct point cloud visibility. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 263–270. IEEE.

Xu, W., Cai, Y., He, D., Lin, J., and Zhang, F. (2022). Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics.*

Zhang, J., Kaess, M., and Singh, S. (2016). On degeneracy of optimization-based state estimation problems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–816. IEEE.

Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2.

Zhang, J. and Singh, S. (2015). Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE.

Zhao, S., Zhang, H., Wang, P., Nogueira, L., and Scherer, S. (2021). Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736.

Zhou, B., Pan, J., Gao, F., and Shen, S. (2021a). Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Transactions on Robotics*, 37(6):1992–2009.

Zhou, B., Zhang, Y., Chen, X., and Shen, S. (2021b). Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786.

How to cite this article: Petrlík, M., Petráček, P., Krátký, V., Musil, T., Stasinchuk, Y., Vrba, M., Báča, T., Heřt, D., Pecka, M., Svoboda, T., & Saska, M. (2023). UAVs beneath the surface: Cooperative autonomy for subterranean search and rescue in DARPA SubT. *Field Robotics, 3*, 1–68.

Chapter 4

Cooperative sensing by robotic teams

The first core publication [3c] related to the field of cooperative sensing presents a UAV-based autonomous data collection approach for realization of RTI. The RTI documentation technique [43] involves capturing a set of images of an object taken from a stationary camera position with each image under illumination from a different, but known direction. This method is often used in contexts like art conservation and restoration, where detailed surface texture documentation is of high interest. The author carried out the design of the novel approach that allows for obtaining the data required for the realization of the RTI method in areas that are not suitable for conventional approaches [129]–[131], which need the scanned objects to be directly accessible for humans. The proposed framework utilizes a team of cooperating UAVs and builds upon the author's previous work on formation flight in cooperative sensing scenarios [16a]. The motion planning efficiency of this system was enhanced in the author's subsequent research [12a] by formulating the sequence of visits to particular locations as the Traveling Salesman Problem with Neighborhoods on a Sphere (TSPNS). The introduced framework for the realization of RTI is an essential part of the DRONUMENT[1] system, which was developed for the documentation of historical buildings with the use of multi-rotor UAVs. Although the proposed method is primarily designed for the realization of the RTI method in high-above-ground locations, it is applicable in general cooperative data collection scenarios with restrictions on occlusions in the field of view of the documentation sensor. This adaptability makes it a valuable tool for a wide range of applications beyond its initial focus on cultural heritage documentation.

The contribution of the author of this thesis on this work is 60%, with co-authors providing feedback on the designed method and the manuscript, and help with the real-world experiments.

[3c] **V. Krátký**, P. Petráček, V. Spurný, and M. Saska, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020

The task of documenting historical structures also motivated the second core publication [4c], which describes a complete framework for documentation and inspection of the interiors of historical monuments using UAVs. The proposed framework addresses the entire documentation process from the initial survey through planning of the missions, autonomous control and navigation software stack deployed in actual documentation process, up to processing of collected data for the end-users. The automation of the documentation process allows for repeatable, frequent monitoring of the objects' states, and thus helps to prevent irreversible damage, while not restricting the regular use of the objects (such as with building scaffolding). The manuscript presents an approach to realization of various advanced documentation

[1] https://mrs.felk.cvut.cz/dronument

techniques and achieved results, including application of the framework in the documentation of historical structures of various size and characteristics. Furthermore, the manuscript outlines a methodology that has been officially recognized and approved by the Czech Ministry of Culture as the standard approach for using multi-rotor UAVs in the documentation of historical structures. Apart from contributing to the field of UAV-based documentation, this publication also represents a significant advancement in the preservation and study of cultural heritage, demonstrating the practical and impactful application of automated UAV technology in sensitive and historically significant environments.

The contribution of the author of this thesis on this work is 41%, with the author being one of two co-first authors with an equal contribution to this work. The particular author's contributions reside in the methodology, writing the manuscript, designing and implementing the cooperative documentation techniques and planning documentation missions, including supporting software, realization of the experiments, and data processing and analysis.

> [4c] P. Petráček, **V. Krátký**, T. Báča, M. Petrlík, and M. Saska, "New era in cultural heritage preservation: Cooperative aerial autonomy for fast digitalization of difficult-to-access interiors of historical monuments," *IEEE Robotics & Automation Magazine*, pp. 2–19, 2023

The author's last core publication oriented towards cooperative sensing is motivated by autonomous aerial filming with additional lighting provided by a team of UAVs. Generating trajectories for aerial cinematography that are optimal and satisfy all constraints being imposed by a task and the environment is a complex, generally non-convex problem, for which finding the solution in real-time onboard UAVs is impractical. In the proposed work, the problem of generating trajectories is decoupled to a sequence of less complex subproblems. As a result, the proposed method is able to generate trajectories on a longer horizon, enabling utilization of this approach in applications requiring higher velocities, such as dynamic target following. A concept of convex decomposition of free space is applied to identify the space of feasible solutions to make the use of a longer horizon possible without introducing additional assumptions on the environment. This technique helps in avoiding unnecessary approximations in the environment representation and also makes the computational requirements for the whole method independent of the number of UAVs in the team. The method is generally applicable in coordinated navigation of a formation of UAVs through complex environments, with a possible application in cooperative monitoring of a static or dynamic target. In the author's subsequent work [18a], [20a], the proposed approach was further extended to allow for simultaneous monitoring of a human worker from various directions and interaction with a human using gestures.

The author of this thesis contributed 38% to this publication, including the initial idea, methodology, implementation of the system, simulation and experimental validation of the method, and writing the manuscript.

> [5c] **V. Krátký**, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, "Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2020. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2020.2970646

Autonomous Reflectance Transformation Imaging by a Team of Unmanned Aerial Vehicles

Vít Krátký, Pavel Petráček, Vojtěch Spurný, and Martin Saska

***Abstract*—A Reflectance Transformation Imaging technique (RTI) realized by multi-rotor Unmanned Aerial Vehicles (UAVs) with a focus on deployment in difficult to access buildings is presented in this paper. RTI is a computational photographic method that captures a surface shape and color of a subject and enables its interactive re-lighting from any direction in a software viewer, revealing details that are not visible with the naked eye. The input of RTI is a set of images captured by a static camera, each one under illumination from a different known direction. We present an innovative approach applying two multi-rotor UAVs to perform this scanning procedure in locations that are hardly accessible or even inaccessible for people. The proposed system is designed for its safe deployment within real-world scenarios in historical buildings with priceless historical value.**

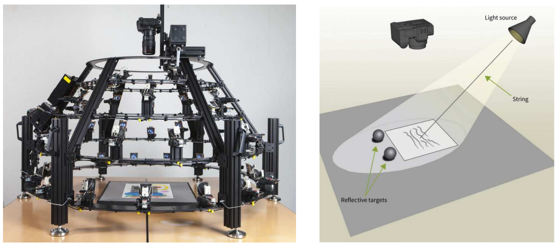
***Index Terms*—Aerial Systems: Applications, Cooperating Robots, Multi-Robot Systems**

I. Introduction

REFLECTANCE Transformation Imaging (RTI) is an image-based rendering method widely used by experts in the field of archaeology, restoration and historical science [1]–[7]. Based on the set of images with varying known lighting, a representation of an image is produced by RTI, that enables to view a captured object lit from an arbitrary direction and therefore to easily inspect the three-dimensional character of the object without the need to capture thousands of photographs with lighting from all possible directions.

The most traditional approaches for gathering the desired set of images is an RTI dome (see Fig. 1a) and the Highlight RTI method [8]. The RTI dome includes tens of light-emitting diodes (LEDs) placed on the inner surface of a hemisphere and a camera placed on its top. During the image capturing phase, the RTI dome is placed above the scanned object, and the LEDs are sequentially lit up while the camera is capturing images. Each image is then labelled with the corresponding lighting vector computed from the known position of particular LEDs.

Using the Highlight RTI method (H-RTI), a source of light is manually placed at unknown positions in a constant distance from the scanned object, while a camera mounted on the tripod is capturing images. Respective lighting vectors are then computed from the reflection detected on the high reflective object (metal ball) placed next to the scanned artifact. An example of a setup for the H-RTI method is illustrated in Fig. 1b.



(a) RTI dome (b) highlight RTI

Fig. 1: Illustration of traditional approaches to the realization of RTI method. Image sources: https://www.idigbio.org/wiki/images/7/70/Graham_0429.pdf and https://historicengland.org.uk/images-books/publications/multi-light-imaging-heritage-applications/heag069-multi-light-imaging/

The drawback of both methods is that the scanned object has to be directly accessible to humans, which is difficult to achieve in large historical and sacred buildings. Thus it significantly limits the usage of this very powerful technique. We propose to solve this problem by applying two cooperating multi-rotor UAVs equipped with a camera and light source. The UAV team is able to gather the set of images with corresponding lighting vectors of objects located at places hardly accessible or even inaccessible for people, and much faster than using the H-RTI approach. During the scanning process, the UAV carrying a camera is hovering steadily in the air, while the UAV equipped with a light source is flying around to provide the lighting from different directions.

Although the scanning process could be performed manually, regardless of the experience of operators, the manual navigation of UAVs to desired positions is typically less precise than in the case of autonomous control. Moreover, the desired scanning locations are assumed to be located high above the ground and hence far from the operators, which increases the difficulty and danger of manual control of two UAVs flying close to each other. Therefore, the proposed method relies on using two fully autonomous and self-localized UAVs. Although we present the system as autonomous, each UAV is supposed to be monitored by an operator, who is prepared to take over the control in case of an unexpected behavior. This requirement is given by the aviation authority for flying outdoors (e.g., for scanning statues, mosaics and plasters on the exteriors of churches and castles - see video https://youtu.be/lTRqd1gQOAI) and by the

Manuscript received: September 10, 2019; Revised December 19, 2019; Accepted January 15, 2020.

This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by project no. DG18P02OVV069 in program NAKI II, by CTU grant no. SGS17/187/OHK3/3T/13, and by the Grant Agency of the Czech Republic under grant no. 17-16900Y.

Authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, Prague 6, {kratkvit|petrapa6|spurnvoj|saskam1}@fel.cvut.cz.

Digital Object Identifier (DOI): see top of this page.

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2020. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2020.2970646

heritage institute for flying indoors (see Fig. 2 and videos from Saint Nicholas Church at Old Town Square in Prague (https://youtu.be/g1NuPnLCFTg), Grotto Gorzanow, Poland (https://youtu.be/6mRYxciDLCM), and St. Anne's and St. Jacob's Church in Stará Voda (https://youtu.be/yNc1WfebIag), where autonomous UAVs have been applied).

The presented application is specific due to cooperation with experts from the Czech National Heritage Institute (https://www.npu.cz/en), who have introduced requirements and constraints, which are untraditional from the robotic point of view. Therefore, two different approaches to the generation of lighting positions and determination of an ideal sequence of these positions to achieve sufficiently good coverage of lighting for the RTI technique are presented in this paper. One of them is based on the method using Fibonacci lattice [9] for achieving approximately equal distribution of points on the sphere, and it applies the approaches for the solution of the Traveling Salesman Problem (TSP) to find a path connecting these positions. The second proposed approach is aimed to find a compromise among the optimality, robotic constraints, and requirements of the aviation authorities and the heritage institute that require paths producing predictable and easy to follow movement of UAVs, which is optimal regarding the UAV deployment in historical buildings.

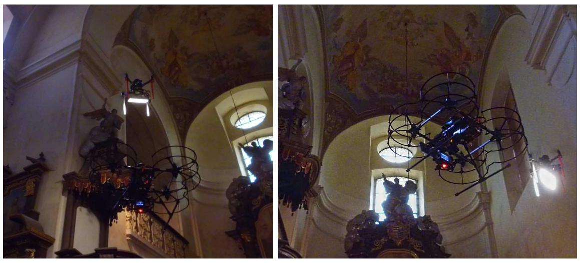


Fig. 2: Deployment of the proposed system for autonomous RTI by a team of UAVs in Church of St. Mary Magdalene in Chlumín. Multimedia material of the experiment is available at http://mrs.felk.cvut.cz/papers/rti2020ral.

A. State-of-the-art and contribution

Single manually controlled UAVs are being commercially used in numerous scenarios, both outdoor and indoor. Nevertheless, the number of possible applications can be significantly increased by introducing autonomous cooperative teams of UAVs, which is the aim of this paper. One of such applications is the documentation of interiors of historical buildings with distributed lighting, which is motivated by the preservation of cultural heritage in the form of digital documentation [10]. It provides the ability to perform later reconstructions of already destroyed historical buildings or art pieces, and also provides the ability to analyze this data and plan the future restoration work without repetitive direct access to particular artifacts.

The documentation of buildings is problematic due to its time complexity and limited accessibility by humans, which naturally leads to the introduction of semi-autonomous or autonomous systems developed for this purpose. Works related to the scanning of buildings are mostly interested in the planning of the best sensing locations [11]–[13] and only a few of them aim to exploit autonomous vehicles. In [14], an unmanned ground vehicle (UGV) equipped with a laser scanner capable of autonomous planning of scanning locations and moving through a large scale outdoor environment is introduced. In [15], authors exploit advantage of UAV systems to operate in larger space by applying them to autonomous inspection of industrial chimneys. Regarding the documentation of particular artifacts in interiors of historical buildings, the only work we have found is [16], where technology assisting an operator of a single UAV explicitly developed for this application is described.

Regarding the multi-robot systems works presenting systems deploying UGVs or UAVs for the documentation or mapping of buildings, the robotic groups are focused on reducing the overall mission time or on expanding the scanned area [17], [18], but the direct cooperation of robots is not exploited. The cooperative lighting by a UAV team introduced in [10], and extended for use of the RTI technique in this paper, is unique in comparison to all the aforementioned works since it employs a team of cooperating UAVs in tasks that cannot be solved by a single UAV only in principle. The proposed method in [10] is exceptional in its approach to actively influence its surrounding environment in order to increase quality and variety of gathered digital material.

In this work, we introduce the first system for autonomous realization of the RTI technique independent on the location of scanned objects, which takes the advantage of our previous works on formation control [10], [19]–[21].

II. PROBLEM DESCRIPTION

The problem of autonomous realization of Reflectance Transformation Imaging technique consists of 1) determining a set of desired positions of a light source, 2) finding a feasible trajectory so that a UAV can provide illumination from these positions, 3) precise mutual localization of UAVs, and 4) processing the captured images for computation of the desired representation of an image. The team of UAVs consists of one UAV equipped with a high-resolution camera and one UAV carrying a light source. Both UAVs are assumed to be capable of steady hovering in the air and controlling the orientation of the camera and the light independently of their motion.

We suppose that the UAVs operate in a known environment represented by a map, obtained from a three-dimensional scan of the historical building, and they are equipped with necessary sensors and software for their precise localization and state estimation [22]. The map of the environment is obtained from a three-dimensional terrestrial laser scanner, providing an incomplete map with missing data in occluded out-of-view locations. Such map is sufficient for localization, however does not provide sufficiently precise and complete models of particular artifacts. Requirements on the scanning process of an object are given by specification of the RTI technique [23] and a position of the object selected for scanning is known prior the mission. Both UAVs are able to accurately follow the trajectory given by the sequence of configurations in an available map of the environment [24].

The output of the system is the requested representation of the image computed from the set of images taken with the camera carried by the UAV. Corresponding lighting vectors, which are needed for computation of this representation, are obtained from a known position of the scanned object and positions of the UAV carrying the light.

III. PRELIMINARIES

A. Reflectance Transformation Imaging

Reflectance Transformation Imaging (RTI) is an image-based rendering method used for obtaining a representation of an image that enables it to be displayed under arbitrary lighting conditions. One type of such representation is the Polynomial Texture Map (PTM), which was proposed by T. Malzbender [23]. In contrast to the common representation of an image, where each pixel has assigned three static values for red, green and blue color (RGB), the simplified version of PTM represents the intensity of each color channel $I_{c,x,y}$, $c \in \{red,\ green,\ blue\}$ of the pixel at position (x, y) by function

$$I_{c,x,y} = f(l_u, l_v), \tag{1}$$

where l_u and l_v are elements of lighting vector and the function $f(\cdot)$ is a second-order bi-quadratic polynomial function with varying coefficients $\alpha_{i,c}$ for particular pixels (x, y). Thus the intensity $I_{c,x,y}$ of each color can be interpreted as

$$\begin{aligned} I_{c,x,y} = \alpha_{1,c} l_u^2 + \alpha_{2,c} l_v^2 + \alpha_{3,c} l_u l_v + \alpha_{4,c} l_u + \\ + \alpha_{5,c} l_v + \alpha_{6,c},\ c \in \{red,\ green,\ blue\}. \end{aligned} \tag{2}$$

The input of the RTI method is a set of images taken from the same viewpoint under varying known lighting conditions, where each image in the set has assigned corresponding lighting vector. With the use of this data, coefficients in equation (2) can be computed for all pixels and their color channels (see [23] for details).

B. Localization

Precise determination of position and orientation of UAVs is a crucial assumption for the good performance of the introduced documentation method. Since we aim at the deploying of the system mostly in indoor environments, we rely on the approach presented in [25], which is capable of working in environments without a sufficient signal from Global Navigation Satellite Systems (GNSS). The method requires UAVs to be equipped with one 360° laser scanner (such as a lightweight RP-Lidar), and two distance sensors (e.g., Garmin LIDAR-Lite v3) oriented downwards and upwards with respect to the frame of UAV. A combination of Iterative Closest Point (ICP) and particle filter algorithm is applied to find the position and orientation of the UAV relative to a three-dimensional point cloud of the environment obtained from a terrestrial laser scanner.

IV. DISTANT AUTONOMOUS RTI METHOD

Methods designed for the realization of the RTI scanning technique by a team of UAVs, which are described in the following sections, are highly influenced by the requirements of experts from the field of restoration and historical science, where key factors are safety and deployability independently to an external infrastructure.

A. Generation of the set of lighting positions

To achieve a good coverage of lighting to a general object during RTI scanning, the lighting vectors need be uniformly distributed over the range defined by the minimum and maximum lighting angles in horizontal ($\lambda_{h,min}$, $\lambda_{h,max}$) and vertical ($\lambda_{v,min}$, $\lambda_{v,max}$) direction. The intensity of lighting presented at the scanned object should be the same for all lighting directions.

Given these two requirements and assumption that the intensity of the light source is constant, we can determine that the desired positions of light sources are distributed on a cap of the sphere with its center located at the position of the scanned object. This task can be defined as the problem of uniform distribution of points on the sphere. Since this problem has an exact solution only for particular cases [26], we apply an approximate approach based on the Fibonacci lattice. Inputs of this process are the number of desired lighting positions to be uniformly distributed over the area defined by angles $\lambda_{d,m}$, $d \in \{h, v\}$, $m \in \{min, max\}$, position of an Object of Interest (OoI), and the desired lighting distance. The resulting set of points Λ_c computed within this process is constructed as

$$\Lambda_c = \Lambda \cup P_i, \tag{3}$$

where Λ is the set of desired lighting positions and $P_i \in \mathbb{R}^3$ is the initial position of the UAV carrying the light.

B. Determination of the optimal sequence

An optimal closed path connecting all the desired RTI lighting positions in the set Λ_c with respect to a certain criterion (minimum energy, shortest path, minimum time) needs to be found. This problem can be defined as TSP, which is usually solved by splitting it into two subproblems - finding paths between all possible pairs of positions from the set Λ_c and finding the optimal sequence of these paths with respect to a certain criterion. The final path is then given as a connection of paths in the optimal sequence. Using this approach, it is difficult to guarantee feasibility of a composed path with respect to constraints given by the kinematic model of a moving robot. Nevertheless, the RTI method requires a static illumination while capturing an image, and so the UAV carrying the light has to be static for taking each picture in the sequence. Therefore, the UAV should stop at every position from Λ_c and the problem of an unfeasible path in connections of curve segments does not need to be considered here.

Considering the expected application of the system, we propose to use the minimum energy as the optimization criterion for the solution of proposed alternative of TSP, which also leads to maximization of possible flight time. Based on our experiments, the energy consumption along the closed trajectory flown at constant velocity is proportional to the length of this trajectory and does not depend on the direction of flight. By combining the observations mentioned above and considering an obstacle-free environment, the problem of finding the optimal sequence of the lighting positions is completely defined as the Euclidean TSP (ETSP). For solution of this problem, we have applied the solver using

Lin-Kernighan heuristic [27] (LKH solver), which belongs to the most efficient approximate algorithms for solution of TSP. An example of path produced with the described approach (further referenced as Fib-LKH) is presented in Fig. 7c.

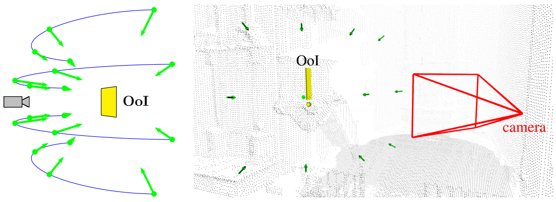
C. Safety pilot predictable approach (SPPA)

In this section, we present an alternative approach to the obtaining of scanning plan that aims at the generation of lighting positions close to uniform distribution and finding a short path connecting these positions while complying to requirements on human predictability of the resulting trajectory. Thus this method decreases the time needed for recognition of the faulty behavior by a safety pilot, who monitors the UAV during the scanning process. This approach is motivated by a technique used by restorers during the manual acquisition of images for RTI method in [28].

The proposed method for obtaining the set of desired lighting positions uses as inputs the border lighting angles $\lambda_{d,m}$, $d \in \{h, v\}$, $m \in \{min, max\}$, the position of scanned object $P_{OoI} \in \mathbb{R}^3$, the orientation of camera defined with yaw and pitch angle $(\psi_{cam}, \zeta_{cam})$, the desired distance between the light and scanned object d_l, and the desired number of samples of the lighting angles v_s in vertical direction for which holds $v_s \geq 2$. In the first step of the method, a set of samples of vertical lighting angles Λ_v from interval $\langle \lambda_{v,max},\, \lambda_{v,min} \rangle$ is obtained so that they are equally distributed over this interval, $|\Lambda_v| = v_s$, $\min(\Lambda_v) = \lambda_{v,min}$ and $\max(\Lambda_v) = \lambda_{v,max}$. Subsequently one spline on which possible positions of the light lie is constructed for each sample λ_v from Λ_v. These splines are parts of a circle and with given $P_{OoI} = [x_{OoI}, y_{OoI}, z_{OoI}]^T$ are defined as

$$\begin{aligned} x_s &= x_{OoI} - d_l \cos(\lambda_v + \zeta_{cam}) \cos(\lambda_h + \psi_{cam}), \\ &\qquad\qquad \lambda_h \in \langle \lambda_{h,min}, \lambda_{h,max} \rangle \\ y_s &= y_{OoI} - d_l \cos(\lambda_v + \zeta_{cam}) \sin(\lambda_h + \psi_{cam}), \\ &\qquad\qquad \lambda_h \in \langle \lambda_{h,min},\, \lambda_{h,max} \rangle \\ z_s &= z_{OoI} - d_l \tan(\lambda_v + \zeta_{cam}). \end{aligned} \tag{4}$$

The splines defined by equation (4) are graphically illustrated in Fig. 3a. The desired distance between the lighting positions on one spline s_d is determined by the equation



(a) illustration (b) experiment in real environment

Fig. 3: The example of the generated set of RTI goals marked with green dots and arrows. The yellow rectangle identifies the scanned object, and the blue curves indicate the horizontal splines, that represent possible positions of the RTI goals.

$$s_d = \frac{d_l(\lambda_{v,max} - \lambda_{v,min})}{v_s}, \tag{5}$$

which corresponds to the shortest distance between two neighboring splines traveled on the surface of the spherical cap. The number of lighting positions on each spline is defined as

$$n_s(\lambda_v) = 1 + \left\lfloor \frac{d_l \cos(\lambda_v)(\lambda_{h,max} - \lambda_{h,min})}{s_d} \right\rceil. \tag{6}$$

The set of sample positions $\Lambda_h(\lambda_v)$ on spline corresponding to angle λ_v is obtained as

$$\Lambda_h(\lambda_v) = \left\{ \lambda_{h,min} + \frac{\lambda_{h,max} - \lambda_{h,min}}{2} \right\}, \tag{7}$$

if $n_s(\lambda_v) = 1$ and

$$\begin{aligned} \Lambda_h(\lambda_v) = \Bigg\{ \lambda_{h,min} + k \frac{\lambda_{h,max} - \lambda_{h,min}}{n_s - 1} \Bigg| \\ k \in \{0, 1, \ldots, n_s - 1\} \Bigg\}, \end{aligned} \tag{8}$$

if $n_s = (\lambda_v) \geq 2$. The complete set of the desired lighting positions Λ_c generated with the SPPA is defined as

$$\Lambda_c = \{\Lambda_h(\lambda_v) | \lambda_v \in \Lambda_v\} \cup P_i. \tag{9}$$

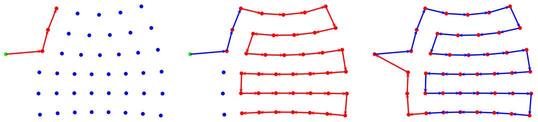
As the first step of the SPPA, the current position of the UAV carrying the light is added at the beginning of the ideal sequence of the RTI positions S_p. Then the closest pair of the RTI positions P_s, $P_e \in \mathbb{R}^3$ on the vertical boundaries needs to be found to select the higher one as the start point and the lower one as the end point among RTI positions. For P_s, P_e holds

$$\begin{aligned} P_s, P_e = \arg \min_{P_{i,j}, P_{k,l}} \; & \operatorname{dist}(P_{i,j}, P_i) + \operatorname{dist}(P_{k,l}, P_i), \\ s.t. \qquad & i = k + 1, \\ & (j, l) \in \{(1, 1), (|\lambda_{h,i}|, |\lambda_{h,k}|)\}, \end{aligned} \tag{10}$$

where $P_{i,j}$ stands for the RTI position in the i-th row and j-th column, function $\operatorname{dist}(\cdot)$ returns the Euclidean distance between two positions given as arguments, and $\lambda_{h,i}$ stands for the set of RTI positions in the i-th row. The position P_s is then added to the sequence of positions S_p. After that, all positions on the vertical boundary on the way up to the highest row are added to S_p. By these three steps, one of the corner positions in the highest row is reached. In the following stages, the procedure depends on the number of rows.

In the case of an even number of horizontal rows, the RTI positions are added line by line with switching the left-right and the right-left direction, and omitting the points that lie on the same vertical boundary as P_s. After reaching the last admissible position in the most bottom row, the remaining points are added from the bottom row up to and including the P_e into S_p. Finally, the P_i is added at the end of the S_p to ensure the return to the initial position. The graphical illustration of this process is shown in Figures 4a, 4b and 4c.

The solution for an odd number of horizontal lines is derived from the solution for even number of rows with several modifications. Firstly, the pair of consequent horizontal lines (indicated by pair of indices $(h_{o,1}, h_{o,2})$) with minimum number of RTI positions is determined. Then, the solution for an odd number of rows is the same as in the case of



(a) (b) (c)

Fig. 4: Illustration of the procedure of determining the safety pilot predictable sequence of RTI positions for even number of horizontal rows. The green dot marks the initial state P_i, the blue dots stand for the unvisited RTI positions, and the red dots for already visited RTI positions. The arrows show the transitions between particular RTI positions, where the red arrows stand for the transitions added during the last step.

even number of rows until the procedure reaches the pair $(h_{o,1}, h_{o,2})$. The particular RTI positions within this pair of rows are traversed either in an up-side-down-side manner (see Fig. 7a for example), or by following the positions in row $h_{o,1}$ to the opposite side, then flying back to the starting side of row $h_{o,2}$ and again following this row to the opposite side. From there, the situation and also the solution is again the same as in the case of an even number of rows.

D. Trajectory Generation and Tracking

The desired trajectory for the RTI (Ω) is generated by the sampling of direct straight paths between consequent RTI positions with the sampling distance d_{RTI} (computed based on the desired velocity), without considering any obstacles. To achieve precise lighting conditions, the UAV carrying the light is supposed to hover at the desired position while taking a photo. This requirement is introduced into the presented system by multiple recurrences of the desired RTI position as the transition point of Ω after each fly-over to the next RTI position. The number of these repetitions is proportional to the time required for the stabilization.

To achieve a reliable following of the desired trajectory Ω in an environment with obstacles and in the presence of disturbances, which cannot be omitted in real systems, the trajectory tracking during the RTI procedure is defined as an optimization task within the MPC framework in the proposed system. Thanks to the independence of the position and orientation control in case of multi-rotor UAVs, the optimization loop can be divided into two separate tasks.

The position control is defined as a nonlinear constrained optimization task over a sequence of control inputs $\mathcal{U}_p(t)$ starting at time t with an objective function J_p, and set of nonlinear constraints $g_p(\cdot)$ on the horizon of length N as

$$\begin{aligned} \mathcal{U}_p(t)^* = \arg\min J_p(\mathcal{U}_p(t)),& \\ s.t.\ g_p(\mathcal{U}_p(t), \mathcal{O}(t)) \quad &\leq 0, \end{aligned} \tag{11}$$

where $\mathcal{O}(t)$ is the set of all obstacles present at time t in the environment, including the UAV carrying the camera.

The objective function $J_p(\cdot)$ is defined as the weighted sum

$$J_p = \alpha J_{pos} + \beta J_c + \gamma J_{obs} + \delta J_{rti}, \tag{12}$$

where J_{pos} stands for the part penalizing the deviations from the desired trajectory, J_c is the part penalizing the changes in sequence of control inputs, and J_{obs} responds for the penalization of trajectories in the proximity of obstacles. The value of J_p is increased by adding J_{rti} for trajectories that lead to occlusions caused by the UAV carrying light or lead to shades in the image caused by the lighting from behind the UAV carrying the camera. Coefficients α, β, γ, and δ are weights used for the scaling of particular parts of the objective function.

The function J_{rti}, which was proposed specifically for this application, is defined as

$$J_{rti} = \sum_{k=1}^{N} \left(\min \left\{ 0, \frac{d_{FoV}(k) - r_{d,FoV}}{d_{FoV}(k) - r_{a,FoV}} \right\} \right)^2, \tag{13}$$

where $r_{d,FoV}$ and $r_{a,FoV}$ are detection and avoidance radii with respect to camera Field of View (FoV), and $d_{FoV}(\cdot)$ stands for the distance from the nearest border of the FoV. This distance can be computed according to equations

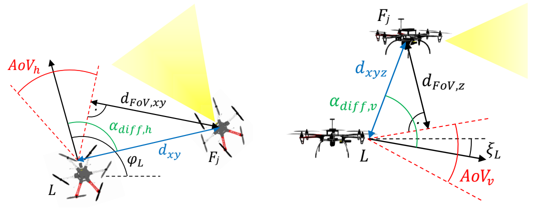
$$\begin{aligned} d_{xy}(k) &= \sqrt{(x_c(k) - x_l(k))^2 + (y_c(k) - y_l(k))^2}, \\ \beta_{diff,h}(k) &= \min\{\alpha_{diff,h}(k), \pi - \alpha_{diff,h}(k)\}, \\ \beta_{diff,v}(k) &= \min\{\alpha_{diff,v}(k), \pi - \alpha_{diff,v}(k)\}, \\ d_{FoV,xy}(k) &= d_{xy}(k) \sin\left(\beta_{diff,h}(k) - \frac{AoV_h}{2}\right), \\ d_{FoV,z}(k) &= \mathrm{dist}(P_l(k), P_c(k)) \sin\left(\beta_{diff,v}(k) - \frac{AoV_v}{2}\right), \\ d_{FoV}(k) &= \sqrt{d_{FoV,z}(k)^2 + d_{FoV,xy}(k)^2} - r_d, \end{aligned} \tag{14}$$

where $P_l(k) = [x_l, y_l, z_l]^T$ and $P_c(k) = [x_c, y_c, z_c]^T$ is the position of UAV carrying light and UAV carrying camera at the time corresponding to k-th transition point. AoV_h and AoV_v are horizontal and vertical angles of the camera FoV, $d_{FoV,xy}(k)$ is the distance to the nearest vertical border of FoV, $d_{FoV,z}(k)$ is the distance to the nearest horizontal border of FoV, and r_d marks the radius of the UAV. $\alpha_{diff,h}(k)$ and $\alpha_{diff,v}(k)$ stand for the angle between the nearest vertical respectively horizontal border of the FoV and connecting line between UAV carrying camera and UAV providing light. $\beta_{diff,h}(k)$ and $\beta_{diff,v}(k)$ are equivalent to $\alpha_{diff,h}(k)$ and $\alpha_{diff,v}(k)$, but besides the FoV of the camera, they include also the FoV of the virtual camera pointed in the exact opposite direction than the real camera. With this alteration, the J_{rti} penalizes not only the occlusion caused by the UAV carrying the light but also the shadows visible in the FoV caused by lighting from behind the UAV carrying the camera, which is important for the RTI image processing. The graphical illustration of symbols used in equation (14) is shown in Fig. 5.

The set of nonlinear constraints $g_p(\cdot) \leq 0$ can be broken down into the following constraints

$$\begin{aligned} g_c(\mathcal{U}_p(k)) \leq 0, \forall k \in \{1, ..., N\}, \\ g_{obs}(P_l(k), \mathcal{O}(t)) \leq 0, \forall k \in \{1, ..., N\}, \\ g_{rti}(P_l(k), \mathcal{O}(t), \psi_c(k)) \leq 0, \forall k \in \{1, ..., N\}, \end{aligned} \tag{15}$$

where $\psi_c(k)$ stands for the configuration of the UAV carrying camera, $g_c(\cdot)$ includes the limitations on control inputs, $g_{obs}(\cdot)$ defines the infeasibility of trajectories colliding with obstacles, and $g_{rti}(\cdot)$ complements the objective function J_{rti}



(a) top view (b) side view

Fig. 5: Graphical illustration of meaning of particular symbols used in equations (14) for computation of part of the objective function penalizing the occlusion caused by the UAV carrying a light

by defining the entire FoV as an unfeasible region.

In a similar manner, the process of finding the optimal sequence of orientation control inputs $\mathcal{U}_o(t)$ on the horizon of length N can be defined as the quadratic constrained optimization task with the objective function $J_o(\cdot)$ and set of nonlinear constraints $g_o(\cdot)$ as

$$\begin{aligned} \mathcal{U}_o(t)^* &= \arg\min J_o(\mathcal{U}_o(t)), \\ s.\ t.\ & g_o(u_o(k), O_j(k)) \leq 0, \forall k \in \{1, ..., N\}. \end{aligned} \tag{16}$$

The objective function $J_o(\cdot)$ consists of two parts

$$J_o = \zeta J_{or} + \kappa J_{co}, \tag{17}$$

where J_{or} is the part penalizing the deviation from the desired orientation of light, J_{co} stands for the part penalizing the fast changes in consequent control inputs $u_o(\cdot)$, and ζ and κ are weights used for scaling of parts of the objective function $J_o(\cdot)$. The set of nonlinear constraints $g_o(\cdot) \leq 0$, $\forall k \in \{1, ..., N\}$ can be split into the following constraints

$$\begin{aligned} g_{co}(u_o(k)) \leq 0, \forall k \in \{1, ..., N\}, \\ g_{or}(O_l(k)) \leq 0, \forall k \in \{1, ..., N\}, \end{aligned} \tag{18}$$

where $O_l(\cdot)$ is the orientation of the light carried by the UAV, $g_{co}(\cdot)$ stands for the constraints introducing the limits on control inputs, and $g_{or}(\cdot)$ introduces the limitations on angles that define the orientation of the light.

V. Experimental Results

A. Performance of generation of lighting positions sequence

The purpose of this section is to qualitatively and quantitatively compare algorithm SPPA, FIB-LKH and their combination which applies SPPA part for the generation of the desired lighting positions and LKH solver for finding a path connecting these positions (further referenced as LKH). The test was performed on the testing case of 10000 samples, each with randomly chosen parameters $\lambda_{v,min}, \lambda_{v,max}, \lambda_{h,min}, \lambda_{h,max}, d_l, v_s$, and initial position of the UAV carrying the light P_i.

The quality of solutions was compared regarding time requirements and the length of the resulting path. Concerning the CPU time, the SPPA is faster than the others. However, since the total CPU time needed by any method does not exceed 0.5 s for all considered problems (computed on the single-core CPU Intel CORE i7 8250), this aspect is not important for our application. More significant parameter is the length of the paths produced by particular methods. Considering this criterion as the comparison value, SPPA is better or equals to LKH solution in 9% of test samples and is not longer by more than 50% in 98% of test samples. Paths generated by FIB-LKH approach are mostly the shortest among all methods. However, they do not fully exploit the borders of the defined scanning area (see Fig. 7c). More detailed results of the quantitative comparison are shown in Fig. 6. Examples of generated sequences by particular methods used for qualitative comparison are shown in Fig. 7.

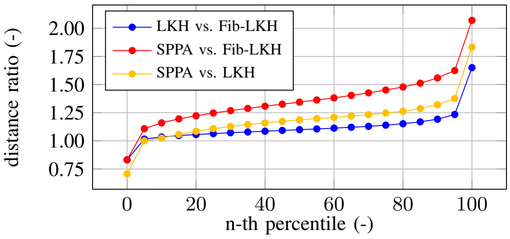
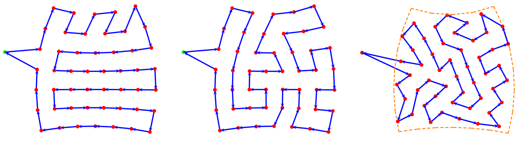


Fig. 6: Comparison of length of paths obtained by SPPA, LKH, and Fib-LKH approach



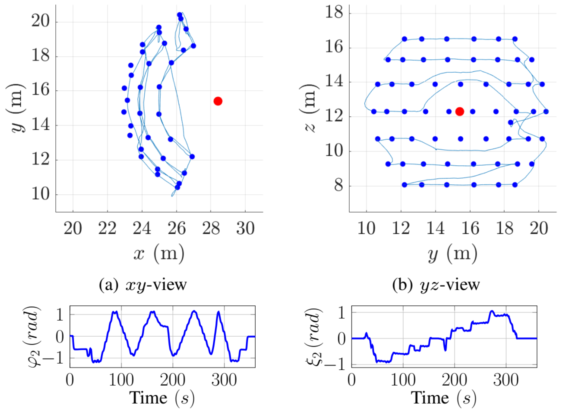
(a) (b) (c)

Fig. 7: Comparison of the solution obtained with SPPA (a), the solution generated by LKH method (b), and the solution generated by Fib-LKH approach (c). The orange dashed line marks the borders of the defined scanning area.

B. Verification of the overall RTI approach

The deployability of the SPPA method for the realization of RTI scanning, which was chosen together with experts from the Czech National Heritage Institute, who are potential end users of the proposed system, as the best variant for its real deployment, was verified through several experiments in the realistic robotic simulator Gazebo and within real-world experiments deploying two autonomous UAVs in the interior of the Church of St. Mary Magdalene in Chlumín.

The presented simulation in which the RTI scanning procedure is performed on the statue situated above the altar leads to the generation of 56 RTI positions and the resulting trajectory of the overall length 110.55 m. The set of generated points together with the trajectory flown by the UAV carrying the light are shown in Figure 8. In compliance with the theory presented in section IV, the UAV carrying light stops at each reachable RTI position and waits until an image is taken by the UAV carrying the camera. In this way, the system collects 56 images of the scanned object under various lighting conditions. The images are then registered to each other to compensate for the motion of UAV carrying a camera during the scanning process. Based on the registered images and the file containing the information about corresponding lighting directions, the PTM representation of the image is computed with the use of program PTM Fitter.



(a) xy-view

(b) yz-view

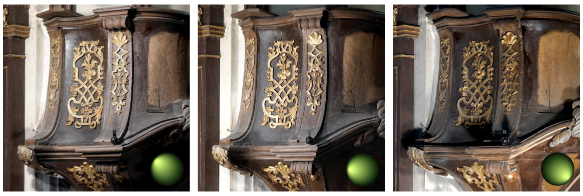
(c) yaw - light

(d) pitch - light

Fig. 8: The generated RTI positions and the trajectory flown by the UAV carrying the light during the RTI scanning procedure. The blue dots mark particular RTI positions, the red dot marks the position of the scanned object, and the blue line shows the trajectory.

The main advantage of obtaining the PTM from the set of images is that the image can be displayed under arbitrary lighting conditions. Since this result can be hardly presented within the printed work, the resulting PTM representation of the scanned object, obtained from the images taken by an onboard camera, is shown in the video available at http://mrs.felk.cvut.cz/papers/rti2020ral.

The real experiment was adapted to fit into the restricted space of the church in Chlumín. To enable the comparison of results of the proposed method and H-RTI, the object of interest (part of the pulpit) was chosen in the height accessible by people and it was illuminated from the same 12 positions (see Fig. 3b) by two different approaches - with the camera carried by an autonomous UAV (see Fig. 2) and with the camera mounted on a static tripod (see Fig. 12c). The latter approach eliminates the imprecision caused by the camera motion and hence enables the objective comparison of the results obtained from the same set of images with lighting vectors computed from the reflections on the black ball (H-RTI) and from the position of the light-carrying UAV provided by the application-tailored localization system [25]. The images generated based on the PTM representation of the scanned object are shown in Fig. 9 and in the video available at http://mrs.felk.cvut.cz/papers/rti2020ral.



(a) upper-left light

(b) left light

(c) bottom-right light

Fig. 9: Presentation of the PTM representation of the scanned object obtained from images taken by UAV during the RTI experiment performed in Church of St. Mary Magdalene in Chlumín. For video see http://mrs.felk.cvut.cz/papers/rti2020ral.

C. Dependence of PTM quality on precision of localization

To examine how the precision of localization affects the quality of the resulting PTM, a simulation-based quantitative comparison was conducted. The whole RTI procedure was performed on a lion statue with localization error sampled from the normal distribution with zero mean and multiple distinct values of standard deviation. The normal map obtained using SPPA (60 positions) and a modelled localization error is compared to the normal map obtained with SPPA (360 positions and zero localization error) used as ground truth. The results of this comparison are presented in Fig. 10. The average difference from the ground truth normals for the normal map obtained for the precision of localization presented in [25] is 0.026 rad (see Fig. 11 for details). This value is lower than the average difference caused by the misplacement of the reflective ball with respect to the center of the scanned object within the H-RTI procedure, which is unavoidable in this method.

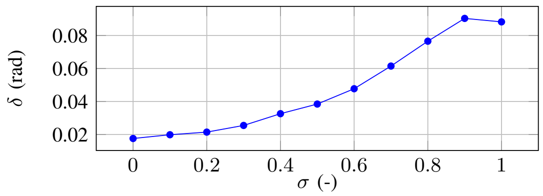
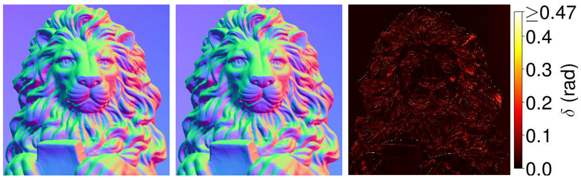


Fig. 10: The dependence of the average error in normals δ on the simulated localization error represented by $\mathcal{N}(0, \sigma^2)$ for positional error (m) and $\mathcal{N}(0, (\frac{2\pi\sigma}{36})^2)$ for orientation error (rad). Values of δ for a particular σ is computed as an average result of 20 experiments.



(a)

(b)

(c)

Fig. 11: Comparison of the normal map obtained with SPPA (360 positions and zero localization error) (a) used as a ground truth and normal map obtained with SPPA (60 positions and localization error (m) modeled by $\mathcal{N}(0,\ 0.09)$ for position and $\mathcal{N}(0,\ 0.003)$ for orientation) (b). Figure (c) shows the size of angle between normal vectors in maps (a) and (b) for particular pixels.

D. Comparison of the proposed approach with H-RTI

For comparison of the proposed approach and H-RTI, results of the RTI method in the form of normal maps are presented in Fig. 12. These two normal maps were obtained with lighting vectors computed by H-RTI method and with lighting vectors computed based on the pose of UAV obtained by the application-tailored localization system. Although the ground truth measurement is not available, we can, based on the known structure of the pulpit, claim that the results obtained with H-RTI method are more precise especially in the surroundings of the reflective ball.

However, the proposed method has an undeniable advantage in realization of the RTI method in hardly accessible places.

Moreover, under the condition of sufficiently precise localization, which is achieved by the applied localization system [25], the determination of lighting vectors is more precise than its detection from reflections on the ball, which cannot be placed directly in the center of a scanned object. The main drawbacks with respect to manually performed RTI lie in the inability to eliminate any camera motion. This issue is partially solved by the image registration process, however, on the high level of details, the imperfections of the alignment can cause unsharpness in images generated from the PTM representation. The camera motion, together with the high exposure time required in dark conditions, also causes the blur in images taken by the camera. However, this problem can be suppressed by the mechanical stabilization of the camera or by use of light source with higher power output, which enables the reduction of exposure time.

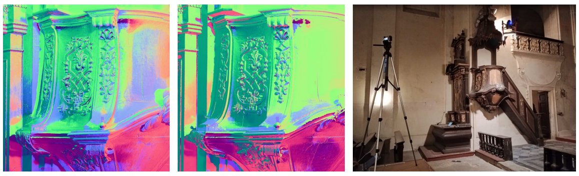


Fig. 12: Comparison of the normal map obtained with SPPA (12 positions) with lighting vectors computed by H-RTI method (a) and with lighting vectors computed based on the pose of UAV (b) obtained by the application-tailored localization system [25]. The setup for this experiment is shown in (c).

VI. Conclusion

The method for the realization of Reflectance Transformation Imaging with the use of a team of autonomous cooperative UAVs is described in this paper. The method is designed for two multi-rotor UAVs equipped with a camera and light source that are capable of self-localization within a given map of an environment. Three approaches to the generation of sequences of RTI positions are presented, but only one was approved by representatives of the heritage institute for its deployment in historical objects. This solution includes self-designed methods for generation of human-predictable trajectories to enable simple monitoring of correct behavior of particular UAVs by safety pilots, while preserving an effort to generate short trajectories. The compromise between these two criteria enables the safe deployment of the system in real-world scenarios. The main advantage of the proposed solution in comparison to already existing methods is the ability to perform the RTI scanning procedure in places that are hardly accessible or even inaccessible to humans.

The proposed approach was integrated into the system for documentation of historical buildings proposed in [10] and its practical applicability was tested in numerous experiments in interiors of churches in realistic simulator Gazebo and within the real experiment in Church of St. Mary Magdalene in Chlumín. Outputs of these tests were evaluated by experts from the field of historical science, who found the results comparable with the results produced by already existing methods, which are limited to accessible locations.

References

[1] H. Mytum *et al.*, “The application of reflectance transformation imaging (rti) in historical archaeology,” *Historical Archaeology*, vol. 52, no. 2, pp. 489–503, 2018.

[2] D. Selmo *et al.*, “Underwater reflectance transformation imaging: a technology for in situ underwater cultural heritage object-level recording,” *Journal of Electronic Imaging*, vol. 26, no. 1, pp. 1–18, 2017.

[3] J. Miles *et al.*, “New applications of photogrammetry and reflectance transformation imaging to an easter island statue,” *Antiquity*, vol. 88, no. 340, pp. 596—605, 2014.

[4] H. Mytum *et al.*, “Reflectance transformation imaging (rti) : Capturing gravestone detail via multiple digital images,” in *Association for Gravestone Studies Quarterly*, 2017.

[5] J. Valcarcel Andrés *et al.*, “Applications of reflectance transformation imaging for documentation and surface analysis in conservation,” *Internetional Journal of Conservation Science*, no. 4, pp. 535–548, 2013.

[6] D. Saunders *et al.*, “Reflectance transformation imaging and imagej: Comparing imaging methodologies for cultural heritage artefacts,” in *EVA*, 2017.

[7] Y. H. Kim *et al.*, “Reflectance transformation imaging method for large-scale objects,” in *CGiV*, 2016.

[8] A. Cosentino, “Macro photography for reflectance transformation imaging: A practical guide to the highlights method,” *e-conservation Journal*, no. 1, pp. 70–85, 2013.

[9] R. Swinbank *et al.*, “Fibonacci grids: A novel approach to global modelling,” *Quarterly Journal of the Royal Meteorological Society*, vol. 132, pp. 1769 – 1793, 2006.

[10] M. Saska *et al.*, “Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control,” in *IEEE ETFA*, 2017.

[11] M. Roberts *et al.*, “Submodular trajectory optimization for aerial 3d scanning,” in *IEEE ICCV*, 2017.

[12] J. P. Fentanes *et al.*, “Algorithm for efficient 3d reconstruction of outdoor environments using mobile robots,” in *IEEE IROS*, 2011.

[13] B. Adler *et al.*, “Finding next best views for autonomous uav mapping through gpu-accelerated particle simulation,” in *IEEE/RSJ IROS*, 2013.

[14] P. S. Blaer *et al.*, “Data acquisition and view planning for 3-d modeling tasks,” in *IEEE/RSJ IROS*, 2007.

[15] M. Nieuwenhuisen *et al.*, “Chimneyspector: Autonomous mav-based indoor chimney inspection employing 3d laser localization and textured surface reconstruction,” in *IEEE ICUAS*, 2017.

[16] N. Hallermann *et al.*, “Vision-based monitoring of heritage monuments – unmanned aerial systems (uas) for detailed inspection and high-accurate survey of structures,” in *STREMAH*, 2015.

[17] H. Qin *et al.*, “Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.

[18] Y. Lyu *et al.*, “Simultaneously multi-uav mapping and control with visual servoing,” in *IEEE ICUAS*, 2015.

[19] M. Saska *et al.*, “Coordination and Navigation of Heterogeneous MAV & UGV Formations Localized by a ”hawk-eye”-like Approach Under a Model Predictive Control Scheme,” *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.

[20] V. Spurný *et al.*, “Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control,” in *MMAR*, 2016.

[21] M. Saska *et al.*, “Predictive control and stabilization of nonholonomic formations with integrated spline-path planning,” *Robotics and Autonomous Systems*, 2015.

[22] T. Báča *et al.*, “Autonomous landing on a moving vehicle with an unmanned aerial vehicle,” *Journal of Field Robotics*, vol. 36, pp. 874–891, 2019.

[23] T. Malzbender *et al.*, “Polynomial Texture Maps,” in *CGIT*, 2001.

[24] T. Báča *et al.*, “Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles,” in *IEEE/RSJ IROS*, 2018.

[25] P. Petráček *et al.*, “Dronument: Reliable deployment of unmanned aerial vehicles in dark areas of large historical monuments,” *IEEE RA-L*, In Press: Accepted for publication on January 9, 2020.

[26] E. B. Saff *et al.*, “Distributing many points on a sphere,” *The Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.

[27] S. Lin *et al.*, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[28] M. Dellepiane *et al.*, “High quality ptm acquisition: Reflection transformation imaging for large objects,” in *International Symposium on VAST*, 2006.

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2023. ACCEPTED TO IEEE RA-M. DOI: 10.1109/MRA.2023.3238213

New Era in Cultural Heritage Preservation

Cooperative Aerial Autonomy for Fast Digitalization of Difficult-to-Access Interiors of Historical Monuments

Pavel Petracek[*×], Vit Kratky[*×], Tomas Baca[*], Matej Petrlik[*], and Martin Saska[*]

***Abstract*—Digital documentation of large interiors of historical buildings is an exhausting task since most of the areas of interest are beyond typical human reach. We advocate the use of autonomous teams of multi-rotor Unmanned Aerial Vehicles (UAVs) to speed up the documentation process by several orders of magnitude while allowing for a repeatable, accurate, and condition-independent solution capable of precise collision-free operation at great heights. The proposed multi-robot approach allows for performing tasks requiring dynamic scene illumination in large-scale real-world scenarios, a process previously applicable only in small-scale laboratory-like conditions. Extensive experimental analyses range from single-UAV imaging to specialized lighting techniques requiring accurate coordination of multiple UAV. The system's robustness is demonstrated in more than two hundred autonomous flights in fifteen historical monuments requiring superior safety while lacking access to external localization. This unique experimental campaign, cooperated with restorers and conservators, brought numerous lessons transferable to other safety-critical robotic missions in documentation and inspection tasks.**

I. Autonomous Aerial Robotics for Heritage Digitalization

Digital documentation of large interiors of historical buildings is an exhausting task since most of the areas of interest are beyond typical human reach. We advocate the use of fully-autonomous teams of cooperating multi-rotor Unmanned Aerial Vehicles (UAVs) to speed up the documentation process by several orders of magnitude while allowing for a repeatable, accurate, and condition-independent solution capable of precise collision-free operation at great heights. In particular, we present a universal autonomy for UAVs cooperating aerially within a team while documenting the interiors of historical buildings for the purposes of restoration planning and documentation works, as well as for assessing the structural state of aging historical sites. We show that the proposed approach of active multi-robot cooperation enables performing documentation tasks requiring dynamic scene illumination in large-scale real-world scenarios, a process previously applicable only manually in areas easily accessible by humans.

The presented system was developed in cooperation with cultural heritage institutions as part of the Dronument project [1] and was deployed fully autonomously in numerous characteristically diverse historical monuments, as exhibited in Fig. 1 and Table III. The included experimental evaluation utilizes UAVs in multiple real-world documentation tasks, and discusses the quality of the obtained results used in subsequent restoration works, as well as suitability of particular techniques for UAVs. The analyses demonstrate the framework's robustness in single and multi-robot deployments in more than two hundred fully-autonomous flights in fifteen historical monuments. In these experiments, the aerial robots rely solely on onboard sensors without access to external localization such as global navigation satellite systems (GNSSs) or motion capture systems, which significantly increases deployability of the system. This unique, extensive, experimental campaign, which cooperated with restorers and conservators, brought numerous lessons learned that are transferable to other safety-critical robotic missions in documentation and inspection tasks. The system also serves as a large part of an official methodological study approved by the Czech National Heritage Institute for its high added value in heritage protection. The methodology (available at [1]) describes the proper usage of UAVs in historical structures for the first time and so prescribes the proposed system to be a standard in this application.

II. Background

Often serving educational, cultural, or social purpose, the preservation of cultural heritage as a valuable reminder of our history is in the greater interest of society. Cultural management and preservation of historical monuments became a relevant topic in the late 19th and 20th centuries when many valuable historical monuments were destroyed while establishing modern infrastructure. By introducing cultural heritage preservation into legislation, the monuments gained protection from human interference. However, being exposed to real-world conditions continually degrades historical buildings and artifacts within. This has initiated the endeavor to actively prevent the irreversible damage of cultural heritage by monitoring its condition and performing restoration and conservation works.

Conservation work on a historical artifact comprises four consecutive phases: the initial survey, the choice of restoration steps and costs evaluation, the actual restoration works, and continued monitoring of the restoration. Both the initial survey and monitoring phase require providing information about the artifact in digital form (usually camera imaging). Thus, these phases are considered a data collection task for which an aerial vehicle, capable of gathering data in a cost-effective and fast manner, can be of great help. This is especially true for areas of interest which are located beyond typical human reach, a situation often arising in tall historical buildings such as churches and cathedrals. Apart from planning restoration works, gathered digital materials can support the reconstruc-

[*] Authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.
[×] *Pavel Petracek and Vit Kratky are co-first authors.*
Corresponding author: pavel.petracek@fel.cvut.cz

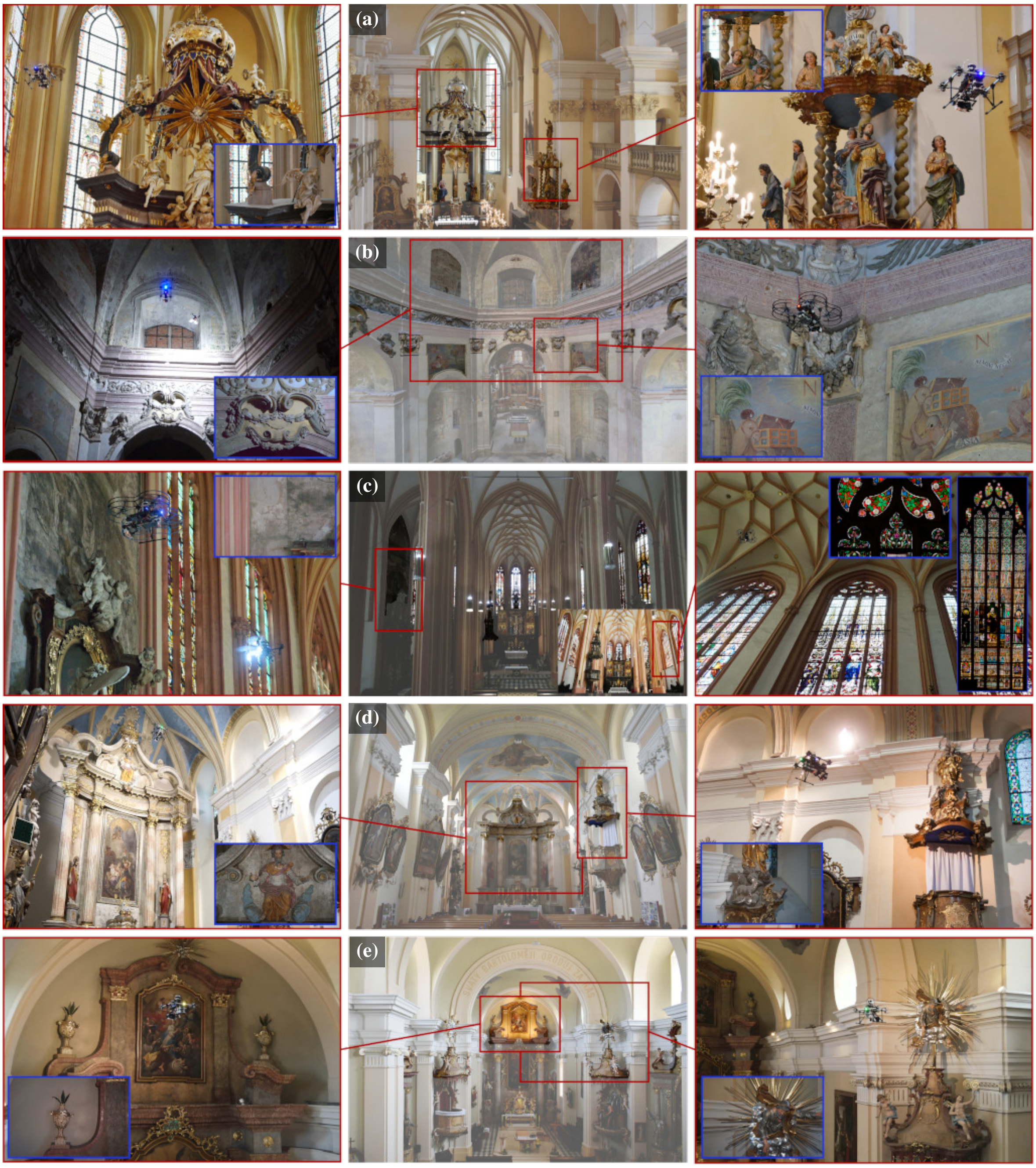


Fig. 1: Illustration of deployment of the presented methodology in selected historical buildings located in the Czech Republic — (a) Church of the Exaltation of the Holy Cross in Prostějov, (b) St. Anne and St. Jacob the Great Church in Stará Voda by Libavá, (c) Church of St. Maurice in Olomouc, (d) Church of the Nativity of the Virgin Mary in Nový Malín, and (e) Church of St. Bartholomew in Zábřeh. Center images show the interiors of the churches with highlighted objects of documentation interest. Side images show actual deployment of UAVs in the particular settings together with example images (highlighted in blue) captured by an onboard camera.

tion of a structure in the event of its sudden accidental destruction (e.g., the burning of the Notre-Dame Cathedral in 2019).

III. Robotics and Automation in Cultural Heritage Preservation

Documentation and digitalization of historical objects requires gathering various types of data, e.g., camera images in visible, infrared (IR) and ultraviolet (UV) spectra, and 3D models. The data gathering is demanding in both time and human resources, particularly in large buildings. This motivates the endeavor to automate data gathering by introducing mobile robotic solutions capable of fast autonomous documentation.The first level of mobile-robot automation can be achieved by applying Unmanned Ground Vehicles (UGVs) as carriers of the documentation sensors. A UGV equipped with a laser scanner and capable of autonomous navigation in constrained environments can sequentially visit several locations to collect a set of scans covering the entire operational space [2]. An advantage of this approach lies primarily in reducing necessary human participation in the scanning process, allowing for the collection of scans from potentially dangerous areas. Several systems applying such an approach were already developed and deployed for scanning historical monuments [3], [4].

Whereas the operational space of UGVs usually does not exceed typical human reach, multi-rotor UAVs capable of 3D navigation in confined environments can be applied for data collection tasks in difficult-to-access areas. In exteriors, UAV solutions abundantly utilize predefined GNSS poses for navigation [5]. In contrast to exteriors, the applicability of UAVs in interiors imposes additional challenges — lack of GNSS localization, navigation in a confined environment, and non-negligible aerodynamic effects. Because of that, UAV systems deployed for indoor data gathering are mainly limited to industrial inspections, with only a few works targeting UAV-based documentation of historical buildings. The specifics of such an application are targeted in this work.

For industrial inspections, the literature typically exploits the environment structure, such as known profiles of tunnels [6] or structured and well-lit warehouses [7]. More general solutions were introduced in the commercial sector introducing semi-autonomous UAV inspection systems[1]— DJI Mavic 3, Elios 3, or Skydio 2+™. In interiors, DJI provides image-based UAV stabilization, Elios allows for human-operated flight with LiDAR and camera-based stabilization and mapping with guarantees of environmental and mechanical protection, and Skydio™ offers automated camera-stabilized flight for interactive 3D reconstruction. Although all these solutions provide an assistive level of autonomy in UAV stabilization, the first two require human-in-the-loop navigation. None of the mentioned solutions offer full interior autonomy, repeatability, modularity, rotor nor sensory redundancy, imaging focusing on capturing high-quality details, and cooperative multi-robot deployment.

As mentioned, aerial data gathering inside historical buildings is rare. A specialized platform for assisting in cultural heritage monitoring called *HeritageBot* was introduced in [8]. However, no evidence of the deployment of this platform in historical monuments is presented. In [9], the authors propose an assistive system to manual control of the UAV during inspection tasks with the experimental deployment of the system inside and outside historical sites.

Among introduced solutions, the most advanced UAV-based systems with the high level of autonomy required for the interiors of historical buildings were introduced in our recent works [10]–[13]. In these publications, we introduced a preliminary application-tailored autonomous UAV system allowing for safe localization and navigation inside historical structures [10], the methodology and algorithms for the realization of advanced documentation techniques found in reflectance transformation imaging (RTI) [11] and raking light (RAK) [12], and an autonomous single-UAV system for realization of documentation missions [13]. All works provide a fully autonomous solution and the possibility of performing documentation techniques in difficult-to-access areas without using mobile lift platforms or scaffolding installation. Here, we progress beyond previous works by introducing a full 3D simultaneous localization and mapping (SLAM) methodology for indoor localization of robots; by advancing robustness to localization drifts and hard-to-detect obstacles with additional sensory redundancy; by improving path, trajectory, and mission planning; by using a UAV team to realize documentation techniques that could not be realized with only a single robot in principle; and by presenting the complete set of results achieved in the Dronument project that are summarized in numerous lessons learned during the unique experimental campaign within highly safety-critical missions.

IV. Documentation Techniques and Associated Constraints

The documentation techniques applied in the field of restoration and cultural heritage preservation aim to capture the current state of the object, survey a potential structural or artistic damage, and determine the age, author and possible dimensions of the elements by identifying the materials and techniques that have been used. For this purpose, diverse methods combining conventional photography in the visible spectrum, photography in invisible spectra making use of different reflective properties of materials, specialized lighting techniques applied for revealing structural details, and even invasive methods based on the collection of material samples are applied. In robotic context, all these methods are associated with varying requirements on sensory equipment, amount of cooperation, and external conditions (mainly illuminance). These relations are summarized in Table I, together with the studied documentation techniques.

The most common documentation technique providing initial information about the studied subject is standard visible spectrum photography (VIS). This technique is applicable to all types of studied objects, ranging from flat paintings and frescoes to 3D structures, including statues and altars. Since the documented areas of historical buildings are often dark, the obtained images suffer from insufficient lighting conditions. Hence, the VIS method often requires additional external lighting to locally increase illuminance, allowing for

[1] DJI Mavic 3: dji.com/cz/mavic-3, Elios 3: flyability.com/elios-3, Skydio 2+™: skydio.com/skydio-2-plus.

TABLE I: Recapitulative table of documentation tasks selected as realizable by aerial vehicles in interiors of historical structures. The squared check marks (☑) identify the realizable documentation methods which were experimentally applied in historical structures, as summarized in section VIII. The last column marks methods for which the ambient light is either required (✓), forbidden (✗), or arbitrary (unmarked).

			Realizable by		Required equipment and lighting conditions		
	Documentation technique		Single robot	Multiple robots	Onboard camera	Onboard light	Ambient light
Spectral analysis	visible spectrum:	photography (VIS)	☑		✓	✓	✓
		transmitography (VISTR)		✓	✓	✓	
		raking light (RAK)	☑	✓	✓	✓	
		three point lighting (TPL)		☑	✓	✓	
		reflectance transformation imaging (RTI)	☑	☑	✓	✓	✗
		light-induced luminescence (VIVL)	✓			✓	
	UV spectrum:	reflectography (UVR)	☑			✓	✗
		fluorescent photography (UVF)	☑	✓	✓	✓	
		false-color reflectography (UVRFC)	✓			✓	✗
	IR spectrum:	reflectography (IRR)	☑			✓	✗
		transmitography (IRRTR)	✓			✓	✗
		fluorescent photography (IRF)	☑	✓	✓	✓	
		false-color reflectography (IRRFC)	✓			✓	✗
	X-ray:	radiography		✓			
Others	3D reconstruction		☑	✓	✓		
	photogrammetry		✓	✓	✓		✓
	environmental monitoring		✓	✓			

the decreased exposure times required to avoid motion blur from instabilities of a multi-rotor vehicle.

Similar to aesthetic photography, light plays a significant role in restoration documentation. Documentation techniques capturing data in the visible spectrum make use of varying lighting intensity and illumination angles to enhance the quality and amount of information that can be derived from the gathered data. The main group of lighting techniques applicable during documentation tasks aims to highlight the 3D characteristics of captured objects, with three point lighting (TPL) being the most routine. TPL illuminates the object with several sources of luminance, each with different intensity and orientation with respect to the camera's optical axis, in order to provide an aesthetically pleasant and realistic view of the 3D object. Another widely used lighting technique is raking light (RAK), which focuses on revealing the surface details of flat objects. While TPL employs several light sources to avoid overshadowed areas, RAK applies a single light as parallel to the scene as possible. The illumination angle in RAK exploits the shadows to highlight the roughness of the surface.

A highly specialized documentation technique used in the field of restoration is the reflectance transformation imaging (RTI) — an image-based rendering method used for obtaining a representation of an image that enables displaying the image under an arbitrary direction of illumination. The necessary inputs of this method include a set of images of an object taken by a static camera, with each image being under illumination from a different but known direction. The captured images and the corresponding lighting vectors are then used for the computation of a polynomial texture map (PTM) representation of the image that enables an interactive illumination and view of the object. Another specialized documentation technique is visible spectrum transmitography (VISTR) which requires a light source to be positioned behind an object of interest (OoI) to transmit the light through this object. However, this method is mainly applied for canvas paintings and thus is rather impractical for realization by UAVs.

Multiple techniques exploit UV and IR lumination and its effects. While the methods based on the visible light focus on revealing structural characteristics and colors, the UV and IR methods aim primarily to identify the materials and hidden layers of artworks. The use of different spectra allows more precise dating of the paintings, as the glow of pigment combinations are unique to certain periods. The first group of methods applying UV and IR lights is based on capturing the fluorescent light in the visible spectrum emitted by an object after absorbing UV or IR radiation energy. These methods are called UV fluorescent photography (UVF) and IR fluorescent photography (IRF) and are used for, e.g., detecting zinc and titanium white (UVF) or cadmium red and Egyptian blue (IRF). The second group of methods applying UV and IR lights captures the reflected light in the corresponding spectra. These methods are called UV reflectography (UVR) and IR reflectography (IRR) and are applicable for, e.g., detecting restored areas, highlighting repairs and re-touchings, enhancing faded paintings (UVR) or reaching the underdrawing layer of paintings (IRR).

Except for VIS, all the above-mentioned methods require positioning the light at a certain angle with respect to the camera. Therefore, these methods are not fully realizable by a single UAV and require a multi-robot coordination. The particular methods can be realized in three different configurations dependent on the requirements of the task. The first configuration employs an autonomous multi-robot team consisting of a UAV carrying a documentation sensor and a set of supporting UAVs providing dynamic lighting of the documented scene. The second configuration applies the UAV as a carrier of the sensor whilst the light is provided by external sources. The third

TABLE II: Typical exposure times of the selected documentation techniques.

Technique	Spectrum	Exposure time (s)
visible spectrum photography	visible	≤ 0.2
raking light	visible	≤ 0.2
three point lighting	visible	≤ 0.2
reflectance transformation imaging	visible	≤ 0.2
UV fluorescent photography	UV	≤ 2.0
UV reflectography	UV	2.0
visible spectrum transmitography	visible	2.0
IR reflectography	IR	4.0
IR transmitography	IR	20.0
light-induced luminescence	visible	25.0
IR fluorescent photography	IR	30.0
radiography	X-ray	≥ 30.0

configuration uses the UAV for positioning the light whereas the data are captured by a static sensor from the ground.

The largest problem in realization of the techniques relying on a UAV carrying a camera is the exposure time required for sharp and detailed imaging. Table II summarizes that the exposure times for some of the methods reach tens of seconds. With constraints on image sharpness, such long times and natural *nonstaticity* of highly dynamical multi-rotor UAVs prevent the realization of these techniques in the camera-carrier mode with satisfactory results. Instead, imaging with a static camera and aerial lighting was investigated for some of these techniques.

The non-spectral tasks applied in the field of preservation mostly focus on the 3D reconstruction and environment monitoring through static sensors measuring physical quantities (e.g., temperature, humidity). The most common techniques applied in 3D reconstruction use visible spectrum images (photogrammetry) or scans produced by laser sensors. From the perspective of the proposed system, the data gathering process for 3D reconstruction does not differ from the realization of VIS and collection of raw data from onboard sensors used for localization and mapping. Monitoring the physical quantities in an environment requires attaching a sensor to the UAV frame and navigating it to the required area. If the measurement process requires permanent monitoring, the sensor must be attached at a specific position in the environment (e.g., adhered to a wall or placed on a mantel). This process is also realizable by UAVs but requires fine control, state estimation, and a mechanism for physical robot-to-environment interaction, as closely tackled in [12].

V. UAV-based Framework for Documentation of Cultural Heritage Interiors

The overall pipeline of the UAV-based framework for interior documentation in historical monuments is showcased in Fig. 2. The framework is composed of three main phases — the pre-deployment phase incorporating pre-flight data gathering and mission planning, the actual deployment of the system in interiors of historical buildings, and post-deployment phase, including processing and utilization of the collected data.

A. Pre-deployment Phase

The first step preceding the entire documentation process is obtaining a model of the environment used for safe navigation of the UAV, as well as for the specification of OoIs that should be scanned during documentation missions. For this purpose, a precise terrestrial 3D scanner Leica BLK360 is employed to obtain a set of scans that are later used for building a complete 3D representation of the target environment, both in form of a global point cloud and a 3D model with a colored texture. The colored 3D model serves for precise specification of the desired camera viewpoints and for presenting the documentation outputs to the public and the end users. The camera viewpoints specifications are made by experts of restoration or historical science who position a virtual camera within the 3D model of the environment using a viewpoint-selection tool shown in Fig. 3a. This tool shows a camera and its view and enables to save the camera viewpoint pose in the global coordinate frame. The optical properties of the camera can be parameterized with respect to the equipment available for real-world documentation, thus allowing for visualizing the desired photo to be captured from a given pose in the colored 3D model.

Given the point cloud representation of the environment and the set of to-be-captured images represented by the respective camera viewpoints in the global coordinate frame, the documentation mission plan is generated as follows. First, the problem of finding an optimal sequence σ^* of camera poses minimizing the overall traveled distance is defined and solved as the Traveling Salesman Problem (TSP). Considering the possible dimensionality of the problem, a solver using an efficient Lin-Kernighan heuristics [14] is employed for the solution of TSP to enable on-site plan generation. Constrained by available computational time, the mutual distance between particular pairs of poses within the solution of TSP are given either by the Euclidean distance or by the length of the collision-free path between the poses. Second, the consequent poses in σ^* are connected by the collision-free paths generated with the use of a grid-based planner [15]. This process creates a path connecting all the poses which can be generally unfeasible if limited flight time of a UAV is taken into account. Hence, the final set of plans $\mathbb{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n\}$ is obtained by splitting σ^* to a set of subsequences $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, where $\sigma^* = \sigma_1 \cup \sigma_2 \cup \ldots \cup \sigma_n$, $\mathbf{P}_i \in \mathbb{P}$ is a collision free path connecting the initial pose with a sequence of poses in σ_i, and equation $\mathrm{t}(\mathbf{P}_i) < t_{max}$ holds, $\forall i \in \{1, \ldots, n\}$, for t_{max} being the maximum flight time of the UAV, and $\mathrm{t}(\mathbf{P}_i)$ being the time needed for following path $\mathbf{P}_i$.

To increase the mission safety, the final step of the pre-deployment phase verifies the paths planned for the documentation mission. First, each plan is verified by humans as collision-free by visualizing it in the 3D model of the environment. Second, the plan feasibility is verified by simulating the entire mission in the realistic Gazebo simulator using the virtual model of the environment with the same software and sensory plugins used during real-world missions. The goal of this two-stage process is to verify that all the generated paths are collision-free and do not traverse potentially risky parts of the environment. The mission specification and plan validation

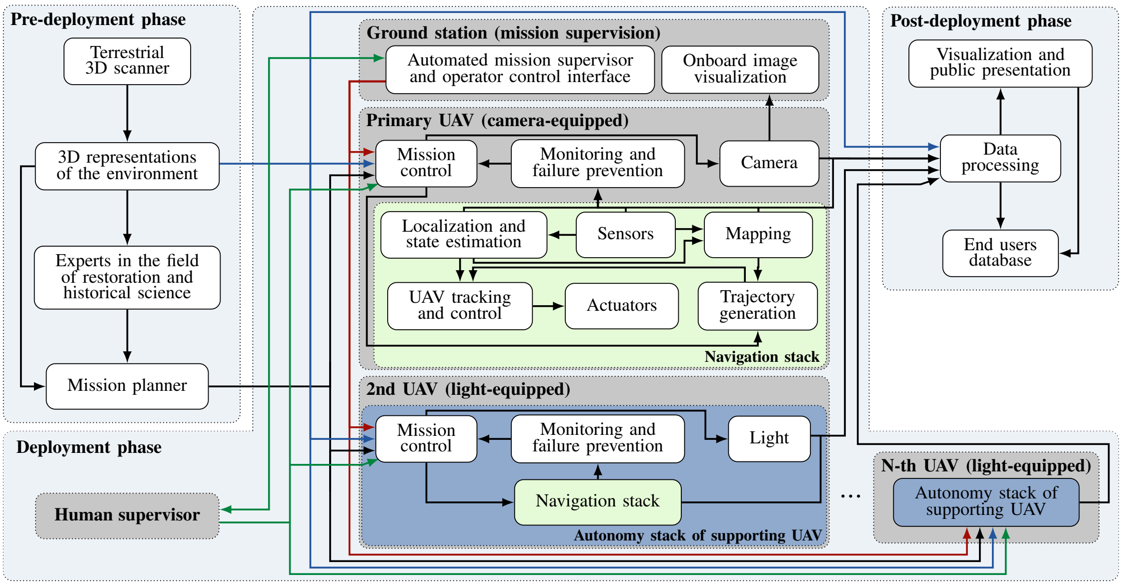

Pre-deployment phase
Terrestrial 3D scanner
3D representations of the environment
Experts in the field of restoration and historical science
Mission planner
Deployment phase
Human supervisor
Ground station (mission supervision)
Automated mission supervisor and operator control interface
Onboard image visualization
Primary UAV (camera-equipped)
Mission control
Monitoring and failure prevention
Camera
Localization and state estimation
Sensors
Mapping
UAV tracking and control
Actuators
Trajectory generation
Navigation stack
2nd UAV (light-equipped)
Mission control
Monitoring and failure prevention
Light
Navigation stack
Autonomy stack of supporting UAV
...
N-th UAV (light-equipped)
Autonomy stack of supporting UAV
Post-deployment phase
Visualization and public presentation
Data processing
End users database

Fig. 2: High-level diagram of the three-phase architecture of the system designed for multi-UAV documentation of interiors of historical buildings. The 3D model of the environment and the mission plan are used as an a-priori generated input for the realization of the documentation mission itself. After the deployment phase, the data gathered during the mission are processed and provided to the end users.

is showcased in Fig. 3.

B. UAV Deployment

The actual system deployment is influenced by the application's specificity imposing strict safety-guarantee requirements. After the necessary hardware checks, all software components are initialized on the onboard computer of each UAV. After successful initialization, all the UAVs automatically align their reference frame with the common frame of coordination by matching their sensory data to the sparse interior map available to each UAV. The outputs of this phase are visually verified by the operator, who checks the correctness of the frames' alignment and validates the mission plan for the last time.

During the following autonomous mission, an automatic centralized supervisor (a ground station) checks the state of all the UAVs in real-time. This supervisor reacts to faults and allows for revealing many possible failures, even preventatively. Available safety actions include stopping all the airborne UAVs at the place at once, navigating them cooperatively to takeoff locations, and landing them at safe locations. Apart from the automatic supervisor, all these actions can be triggered by a human operator supervising the mission in parallel using the ground station. At last, a human operator serves as the final safety measure capable of landing the UAVs manually. The autonomy stack is described in section VI.

C. Post-deployment Phase

To increase the quality and range of the outputs, the data collected during autonomous flights in historical buildings are processed before being provided to the end users. This includes post-processing of onboard sensory data to increase accuracy of pose referencing associated with the captured data frames, stitching images into photomaps, or building a 3D model of the environment in areas occluded in ground-located scans. The generated data then serve for digitalization and archivation, pre- and post-restoration analyses, state assessment and monitoring, material analyses, photogrammetry, and for digital presentation to the public.

VI. Fully Autonomous, Cooperating UAVs

To benefit from extensively tested and field-verified methods, the proposed multi-UAV system is based on the open source MRS UAV system[2] developed within the authors' research group. In this section, let us summarize novel scientific results achieved within the presented project Dronument, whilst the MRS UAV system is described in detail in [16].

A. Reference Frame Alignment

The reference frames of the robots are aligned once during a pre-takeoff phase with each robot performing the alignment independently in four automated phases. This alignment process is mandatory for each robot as the supervising controller does not allow any robot to takeoff unless all robot frames are aligned with the global coordination frame (i.e., the map).

In the *data loading* phase, each robot loads the global map **M** and a single 3D LiDAR data-frame **D** to its memory, applies voxelization to both the objects for dimensionality reduction, and removes outliers in **D** using radius outlier filter. The z-axis of both the point clouds **M** and **D** is assumed to be

[2] github.com/ctu-mrs/mrs_uav_system

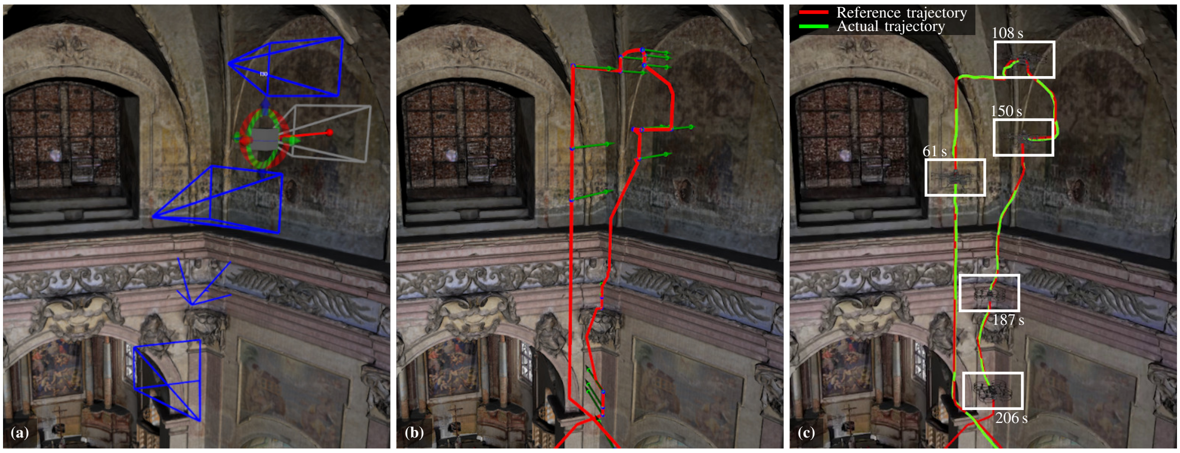


Fig. 3: Pre-deployment phase of the proposed framework — (a) specification of the documentation task by selecting a set of camera viewpoints within the 3D model of the environment, (b) planning trajectory of the robot (in red) which visits all the specified viewpoints (in green), and (c) verification of the mission plan in Gazebo simulator employing identical software that is used during real-world missions.

approximately parallel to the gravity vector. During *global correlation* phase, the origins and orientations of $\mathbf{M}$ and $\mathbf{D}$ are approximately matched. First, convex 3D-space hulls $\mathbf{H_M}$ and $\mathbf{H_D}$ are computed using Qhull [17] with a hull being represented as a set of undirected edges $\mathbf{H} = \{(\mathbf{v}_a, \mathbf{v}_b)_i\}$ (set of vertex pairs). Translation $\mathbf{t}_\mathbf{D}^\mathbf{M} \in \mathbb{R}^3$ of $\mathbf{D}$ to $\mathbf{M}$ is given as $\mathbf{t}_\mathbf{D}^\mathbf{M} = \mathbf{b_M} - \mathbf{b_D}$, where $\mathbf{b_X} \in \mathbb{R}^3$, $\mathbf{X} \in \{\mathbf{M}, \mathbf{D}\}$, represents a polyline barycenter of an edge set $\mathbf{X}$ as

$$\mathbf{b_X} = \frac{\sum_{(\mathbf{v}_a,\mathbf{v}_b)\in\mathbf{X}} \left[\mathbf{v}_a + (\mathbf{v}_b - \mathbf{v}_a)/2\right] ||\mathbf{v}_b - \mathbf{v}_a||_2}{\sum_{(\mathbf{v}_a,\mathbf{v}_b)\in\mathbf{X}} ||\mathbf{v}_b - \mathbf{v}_a||_2}. \tag{1}$$

The UAV is assumed to be taking off from ground locations, hence the grounds are coupled by setting z-axis translation to $\mathbf{t}_\mathbf{D}^\mathbf{M}(z) = \min_{\mathbf{p}\in\mathbf{M}} \mathbf{p}(z) - \min_{\mathbf{p}\in\mathbf{D}} \mathbf{p}(z)$, where $\mathbf{p}(z)$ denotes the z coordinate of point $\mathbf{p}$. Initial transformation to the consequent optimization phases is then given as

$$\mathbf{T_I} = \mathbf{T}(\mathbf{t}_\mathbf{D}^\mathbf{M})\,\mathbf{T}\left(\mathbf{t_D}, \mathbf{z}, \theta\right), \tag{2}$$

where $\mathbf{T} \in \mathbb{R}^{4\times4}$ is a general 3D transformation in the matrix form and $\mathbf{T}(\mathbf{t_D}, \mathbf{z}, \theta)$ is the matrix form of a z-axis rotation at a point $\mathbf{t_D} \in \mathbb{R}^3$ (the origin of $\mathbf{D}$) by angle θ. The rotation angle is given as $\theta = \theta_\mathbf{M} - \theta_\mathbf{D}$, where $\theta_\mathbf{X} = \arctan \xi_y^\mathbf{X} / \xi_x^\mathbf{X}$, $\xi^\mathbf{X} = \left(\xi_x^\mathbf{X}, \xi_y^\mathbf{X}, \xi_z^\mathbf{X}\right) = \arg\max_{\xi\in\Xi(\mathbf{X})} \sqrt{\xi_x^2 + \xi_y^2}$, and $\Xi(\mathbf{X})$ is the set of covariance matrix eigenvectors of the point cloud $\mathbf{X}$.

The following *global registration* phase copes with the lateral symmetry of the environments as typical of large historical structures. Several Iterative Closest Point (ICP) routines $\mathrm{ICP}(\mathbf{T})$ are performed in this phase, each with different initializations $\mathbf{T}$ and loosely set parameters for point association and convergence requirements. Given a number of desired initializations k, this phase selects $\theta^* = \arg\min_{\theta\in\Theta} \mathrm{ICP}\left(\mathbf{T_I}\mathbf{T}(\mathbf{t_D}, \mathbf{z}, \theta)\right)$ where $\Theta = \{2\pi i/k \mid i \in \{0, 1, \ldots, k-1\}\}$. Final *fine-tuning optimization* phase estimates robot origin in the global coordinate frame $\mathbf{T}_\mathbf{D}^\mathbf{M}$ by running $\mathrm{ICP}\left(\mathbf{T_I}\mathbf{T}(\mathbf{t_D}, \mathbf{z}, \theta^*)\right)$ optimization set with high-accuracy parameters and strict convergence criteria.

B. State Estimation, Localization, and Mapping

Estimating the 3D state of a UAV (i.e., pose and its derivatives) in real-time is crucial for the UAV mid-air control and 3D navigation. To keep the robot steady while airborne, follow reference trajectories, and avoid obstacles, the environment needs to be perceived with robot's onboard sensors (e.g., cameras, LiDARs). As state estimation, localization, and mapping are critical for collision-free flight, the utilized algorithms are based on well-tested works with implementation validated in differing real-world scenarios. To estimate the robot state, a bank of Kalman filters [16] extended with smoothing over a short past-measurements buffer fuses onboard inertial measurements with localization outputs, providing real-time feedback to the position control loop [16]. The localization and mapping systems utilize low-drift pose estimation LOAM [18]. An extensive evaluation in [15] showed that fusing LOAM efficiently with [16] provides sufficient accuracy and robustness even in safety-critical applications. The architecture of the control, state estimation, and localization pipelines is analogous to [15]. In contrast to [15], the mapping pipeline uses an a-priori map of the environment to derive a global frame for the robots' missions (see its calibration in subsection VI-A). The a-priori shared map enables multi-robot coordination and global mission planning, but also provides an additional safety level by allowing robust online analysis of localization drift and cross-checking of sensory measurements.

C. Navigation and Trajectory Tracking

The navigation of the UAVs during the mission follows a mission plan $\mathbf{P} \in \mathbb{P}$ generated in the pre-deployment phase, described in subsection V-A. This collision-free plan is represented by a sequence of triplets $\mathbf{P} = \left[(\mathbf{p}_{uav}, \mathbf{p}_{ooi}, \mathbb{I})_1, \ \ldots, \ (\mathbf{p}_{uav}, \mathbf{p}_{ooi}, \mathbb{I})_{|\mathbf{P}|}\right]$, where $\mathbb{I} \in \{0, 1\}$ is the acquisition flag. The triplets with $\mathbb{I} = 1$ specify the UAV poses $\mathbf{p}_{uav}$ in which capturing an image or illuminating the

OoI at pose $\mathbf{p}_{ooi}$ is required. The reference trajectory $\mathbf{R}$ is generated by uniform sampling of the collision-free path given as sequence of $\mathbf{p}_{uav} \in \mathbf{P}$ such that the sampling step respects the required velocity. The UAV is requested to stop at each pose $\mathbf{p}_{uav} \in \mathbf{P}$ where $\mathbb{I} = 1$ to improve the quality of data acquisition by minimizing deviation from the desired pose and reducing the motion blur that would occur in case of non-zero velocity during image capturing. The reference trajectory $\mathbf{R}$ then serves as an input to the trajectory tracking module using model predictive control (MPC). This module, described in our previous works [11], [19], produces a smooth collision-free trajectory while penalizing deviations from the original reference trajectory and respecting dynamic constraints of the UAV. The smooth-sampled reference trajectory is then passed into a feedback controller (implemented within the MRS UAV system [16]) handling tracking of the trajectory.

D. Multi-robot Coordination and Cooperation

Since the characteristics of the expected environment enable reliable use of standard communication channels, the cooperation algorithms rely on the information shared through a Wi-Fi interface among the UAVs and a ground station. Namely, the UAVs share their current poses, planned trajectories, and individual statuses based on the information from their onboard sensors. The same communication channel is utilized for commanding the UAVs from the ground station in case of emergency or a change in the mission plan, and for sharing specific messages among the UAVs during the realization of cooperative documentation techniques. The algorithms handling the autonomous flight are computed on board the UAVs.

During the cooperation, the reference trajectories of the UAVs are generated in a distributed manner on a short horizon corresponding to the optimization horizon used in the MPC-based trajectory tracking module [16]. By applying concepts of leader-follower architectures, the reference trajectories of supporting UAVs are generated with respect to the optimized trajectory of the primary UAV (leader), to the position and the desired distance of the UAV from the OoI, and to the desired lighting angle with respect to the optical axis of the documentation sensor on board the primary UAV. The coordination of the UAVs is part of trajectory optimization (see subsection VI-C) where both the current poses of the UAVs and their planned trajectories are considered to be part of constrained unfeasible space [19]. To prevent the downwash effect, this optimization is also constrained to not allow two nearby UAVs to fly above each other.

VII. Aerial Platforms

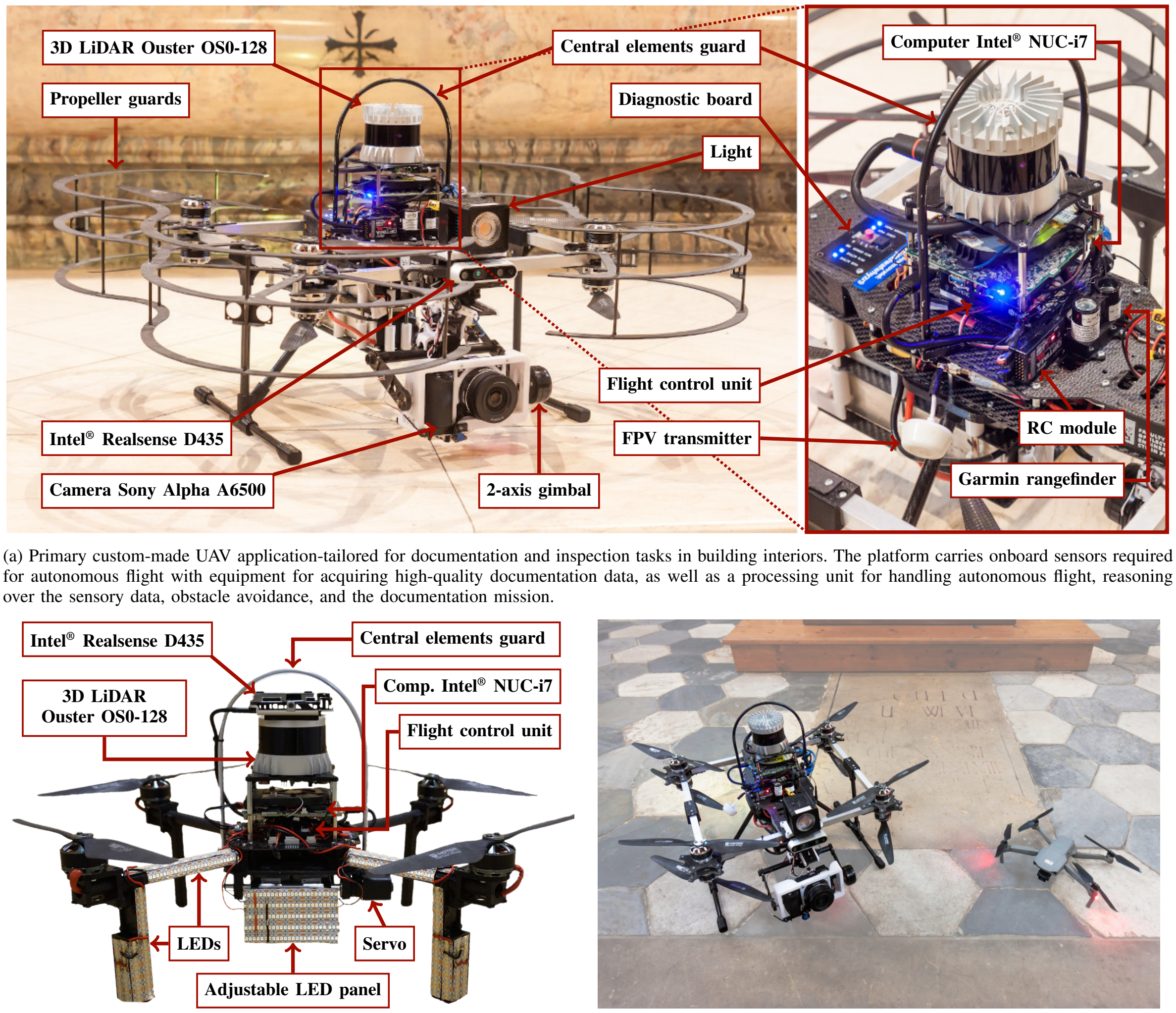
Two custom-made UAV platforms were designed specifically for the proposed application of deployment in interiors of buildings. Both the platforms, as shown in Fig. 4 and described in more detail in [20], support fully autonomous deployment within the tackled domain by carrying sensors for local environment perception together with a powerful computational unit handling the entire autonomous aerial mission. The primary platform is a heavy-weight (5.5 kg without payload) octo-rotor with dimensions of 78 × 81 × 40 cm, capable of carrying up to 1.5 kg payload — enough for a mirrorless interchangeable-lens (MIL) camera with a suitable lens and 2-axis gimbal stabilization, as well as an onboard light source. This platform minimizes its dimensions while maximizing the payload capacities, is equipped with mechanical propeller guards, and carries sensory redundancy for active obstacle avoidance. The secondary platform is a lightweight (3 kg fully loaded) quad-rotor with dimensions of 68 × 68 × 30 cm suited for assisting the primary UAV throughout a documentation process by providing the scene illumination, thus increasing the quality of the gathered digital materials. While cooperating, the supporting UAVs assist in performing tasks inexecutable by a single UAV in principle. As the primary payload, the secondary platform carries a set of high-power light sources. Both the platforms support flights in close proximity to obstacles and to other UAVs. However, relative distances are limited to a minimum of 2 m to limit the aerodynamic influence of downwash, ceiling, and ground effects on the UAV, and the contrary effect of the UAV on the environment (possible damage of not firmly attached objects and fragile plasters).

A. Sensors for Autonomy

For autonomy in GNSS-denied environments, both platforms rely on onboard sensors only. The primary sensor is a 3D light detection and ranging (LiDAR) Ouster OS0-128 with 50 m detection range and 90° vertical FOV supported by a thermally-stabilized triple-redundancy inertial measurement unit, downward and upward looking point-distance sensors Garmin LiDAR Lite, and front-facing (primary UAV) or downward and upward-facing (secondary UAV) color-depth cameras Intel® Realsense D435 for sensory cross-checking in active obstacle avoidance. All the sensory data are processed by an Intel® NUC-i7 onboard computer which utilizes data in real-time algorithms handling the autonomous aerial mission. The low-level control (attitude stabilization) is handled by Pixhawk 2.1, an open-source autopilot used frequently by the robotic community. For safety reasons, the primary UAV carries a visible diagnostic RGB LED which indicates a possible failure to an operator who is authorized to override UAV autonomy for manual landing.

B. Payload

The payload equipment mountable on board the platforms is modular — cameras, lenses, and light sources can be easily interchanged for the purposes of a specific task. For general purposes, the primary UAV carries a 2-axis gimbal FlyDrotec capable of stabilizing up to 850 g payload. The stabilized axes are controllable, a feature useful mainly for controlling the pitch angle of a camera. Throughout our experiments, a MIL camera, the Sony Alpha A6500 with varying lenses, has been used for its integrated image-sensor stabilization, further minimizing the negative effect of mid-flight vibrations on the output image quality. Triggering image capture is automated via the onboard computer, whereas real-time imaging is transmitted to the ground for online visualization for the operator.



(a) Primary custom-made UAV application-tailored for documentation and inspection tasks in building interiors. The platform carries onboard sensors required for autonomous flight with equipment for acquiring high-quality documentation data, as well as a processing unit for handling autonomous flight, reasoning over the sensory data, obstacle avoidance, and the documentation mission.

(b) Secondary UAV tailored for supporting documentation tasks in building interiors. In contrast to the primary UAV (a), this platform is smaller and carries a high-power light instead of sensors for the documentation task.

(c) Comparison of the custom-made UAV platform (a) with lightweight commercial drone DJI Mavic Air 2, which carries a small camera sensor and does not support complex mission planning in building interiors.

Fig. 4: Aerial platforms used for documentation tasks in the Dronument project — primary UAV carrying documentation sensors (a), secondary UAV assisting in cooperative documentation (b), and commercial drone used for qualitative comparison (c). Both (a) and (b) carry environment-perception sensors and computational resources allowing fully autonomous deployment in interiors with poor lighting conditions.

VIII. Experiments and Results

The extreme requirements on safety imposed by the nature of the application requiring the deployment of UAVs in priceless historical buildings imply thorough validation of all the developed software and hardware solutions prior to their deployment in real-world missions. The software solutions ranging from the state estimation and control algorithms to high-level mission control were intensively tested with the use of Gazebo simulator and the MRS simulation package[3] providing realistic behavior of the UAVs. Running the same software with identical parametrization in simulation and on real hardware significantly simplifies the transfer of algorithms from the virtual environment to real-world applications. The 3D models built from the obtained 3D scans are directly used as the simulation environments for algorithms' testing. Together with simulated sensory noises and model inaccuracies, this makes the simulation as analogous to real-world conditions as possible. This methodology proves to be especially useful for discovering possible failures correlated with specific environments and validating the entire autonomous missions in an approximate copy of the real-world scenarios. Although the simulator is highly realistic, running the system in the real world introduces additional constraints. Therefore, even after thorough testing in virtual environments, the first deployments of the system

[3] github.com/ctu-mrs/simulation

TABLE III: Overview of three-phase UAV deployment in historical buildings within the Dronument project. The first phase focused on specifying the use cases, developing the methodology, designing the system, and performing preliminary experiments, including manually controlled flights. The second phase investigated autonomous multi-robot coordination in cooperative documentation and experimented with imaging outside the visible spectrum and with the physical interaction of UAVs with the environment. The third phase deployed the system in a full-operation mode for gathering data valuable to end users and for validating the methodology and overall performance of the autonomy.

Object (approximate floor area over which the system operated)	flights	images	flight time (h:mm:ss)	flight dist. (m)	maximum height (m)	min. obst. dist. (m)	multi-robot	applied methods	
Archbishop's Chateau in Kroměříž (UNESCO, 420 m^2)	10	202	0:24:30	320	9.0	2.0	✗	VIS	**Phase 1 2017–2019**
Vranov nad Dyjí State Chateau (410 m^2)	18	1049	1:26:44	1020	12.0	3.0	✗	VIS	
Klein Family Mausoleum in Sobotín (30 m^2)	7	274	0:17:37	120	3.8	1.6	✗	VIS	
Rondel at State Chateau and Castle Jindřichův Hradec (140 m^2)	8	660	0:51:32	940	7.8	2.1	✗	VIS	
Chapel of All Saints at Chateau Telč (UNESCO, 84 m^2)	6	190	0:14:04	145	4.8	1.7	✗	VIS	
Church of St. Mary Magdalene in Chlumín (224 m^2)	8	86	0:23:40	146	4.8	1.5	✓	RTI	**Phase 2 2019–2021**
Church of the Holy Trinity in Běhařovice (252 m^2)	6	56	0:07:10	30	5.2	2.2	✗	IRF, UVF, IRR	
Church of St. Maurice in Olomouc (1160 m^2)	27	971	1:31:38	1340	16.8	1.5	✓	VIS, TPL, RAK	**Phase 3 2021–2022**
Church of St. Anne and St. Jacob the Great in Stará Voda (505 m^2)	95	6022	4:54:10	4540	19.5	1.7	✓	VIS, TPL, RTI RAK, UVR, IRR	
Church of the Exaltation of the Holy Cross in Prostějov (570 m^2)	7	548	0:19:49	308	15.2	1.7	✗	VIS	
Church of Our Lady of the Snows in Olomouc (918 m^2)	3	255	0:10:06	185	17.1	2.4	✗	VIS	
Church of the Assumption of the Virgin Mary in Cholina (409 m^2)	3	82	0:08:51	132	7.5	1.4	✗	VIS	
Church of the Nativity of the Virgin Mary in Nový Malín (282 m^2)	2	129	0:06:06	68	8.8	1.8	✗	VIS	
Church of the Holy Trinity in Kopřivná (367 m^2)	4	211	0:17:50	247	11.4	2.0	✗	VIS	
Church of St. Bartholomew in Zábřeh (616 m^2)	4	263	0:18:23	258	12.4	1.4	✗	VIS	
Total (6387 m^2)	208	10998	11:32:10	9799	19.5	1.4	✓	VIS, TPL, RTI RAK, UVR, IRR UVF, IRF	

were preceded by test flights in mock-up scenarios and testing interiors in order to reveal potential problems related to transfer of the system from simulation to real hardware.

The final version of the system, as presented in this manuscript, builds on preliminary versions and architectures of both software and hardware stacks and integrates experience from over a year and a half period of experimental deployments. During the experimental campaigns, remaining sources of potential failures were identified and the UAV system upgraded to reach the desired performance and reliability while increasing the number of realizable documentation techniques. The entire system was, to this day, deployed in real-world missions in fifteen historical buildings of various characteristics (summarized in Table III), including one of the largest Baroque halls in the Czech Republic at State Chateau Vranov nad Dyjí and the UNESCO World Heritage Sites, Archbishop's Chateau in Kroměříž and Chateau Telč. Almost twelve airborne hours in more than two hundred flights have been performed for purposes of documentation missions in the given structures. Such an extensive experimental campaign provides an exhaustive validation of the system in real-world conditions and supports its applicability in GNSS-denied environments by identifying and overcoming challenges imposed by specific scenarios. The following sections describe the documentation techniques realized by the system in these structures. The OoIs of the presented documentation missions are showcased in Fig. 1.

A. Visible Spectrum Photography

Imaging in the visible spectrum is the most frequently applied technique as it includes methods providing the widest range of practical information while being relatively easy to perform. Within the fifteen historical structures, OoIs of various characteristics have been imaged by autonomous UAVs. These OoIs range from artistic elements, such as paintings, stained-glass windows, mosaics, stuccoes, and murals located in the most upper parts of the main naves, to complex 3D structures, such as window frames and altars up to 20 m high.

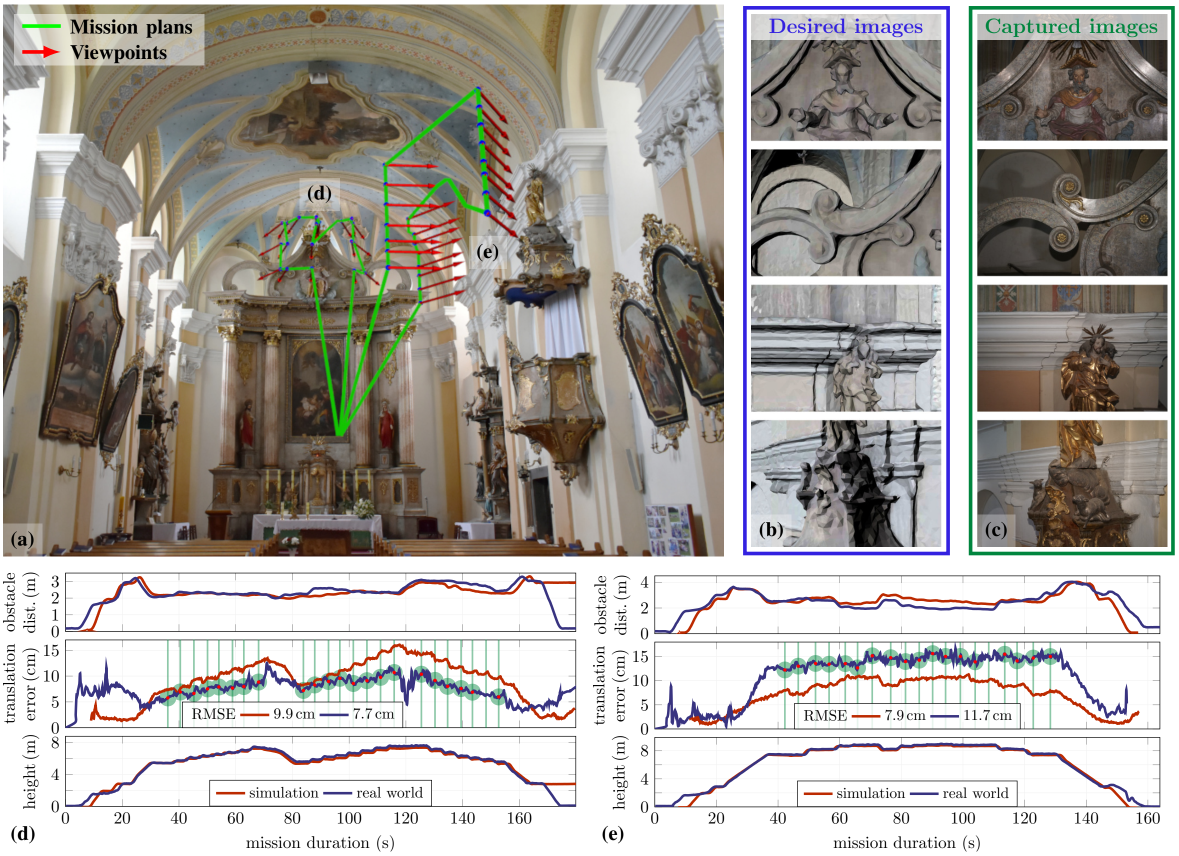


Fig. 5: An example documentation mission in the Church of the Nativity of the Virgin Mary in Nový Malín. The documentation mission was divided into two separate flights (a) focused on documenting the upper part of the altar (d) and baldachin of the pulpit (e). When going from the top, the rows in (d) and (e) show the minimal distance from the UAV frame to an obstacle, the 3D position error with image acquisition times (green vertical lines and red dots), and the height above ground. The desired imagery specified in the 3D model and the images captured on board are compared in (b) and (c). The simulation data are averaged from 5 runs each.

Additionally, objects may include structural damage, such as crevices, cracks, or fractures.

An example of fully-autonomous documentation of a single interior is provided in Fig. 5 depicting the documentation of a baroque church. The specified viewpoints were focused on documentation of two OoIs — the upper part of the altar reaching a height of 10 m and the baldachin of the pulpit. The automated process of viewpoints' specification and autonomous navigation has enabled fast realization of the documentation process in just two single-UAV flights lasting only 366 s in total. With mission specification being part of the pre-deployment phase, the overall time required for in-site deployment reached only 80 min, including equipment unpacking, flight test, mission validation and execution, and packing. Such a high level of autonomy in the process demonstrates superiority via fast, safe, effective, and repeatable data capturing when compared to the slow, imprecise, and dangerous manual control of the UAV in obstacle-filled environments by even a highly trained human operator. Even with assistive systems (stabilization and collision prevention) guiding the human in navigation, manual operation is unsafe in losses of line of sight in the presence of obstacles and inefficient in time and accuracy required to reach the desired viewpoints. Apart from higher efficiency and safety of autonomy in contrast to human-controlled flying, a fully autonomous system allows flight in close proximity to obstacles, enlarging the operational space of the UAV. This is advantageous particularly when documenting elevated OoIs where the inaccuracy in estimating the UAV's distance to the ceiling is proportional to the distance from the human eye, thus making manual navigation in these areas unsafe.

The VIS method can be performed with commercially available products (e.g., DJI Mavic) offering semi-autonomous solutions in small and lightweight packages. However, the limited level of autonomy and sensory modularity makes the realization of the missions in large interiors prolonged (the proposed system is on average ten times faster in the same task), non-repeatable, or even impossible in conditions unfavorable to onboard perception or the desired documentation technique. In Fig. 6, the images obtained by the proposed

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2023. ACCEPTED TO IEEE RA-M. DOI: 10.1109/MRA.2023.3238213

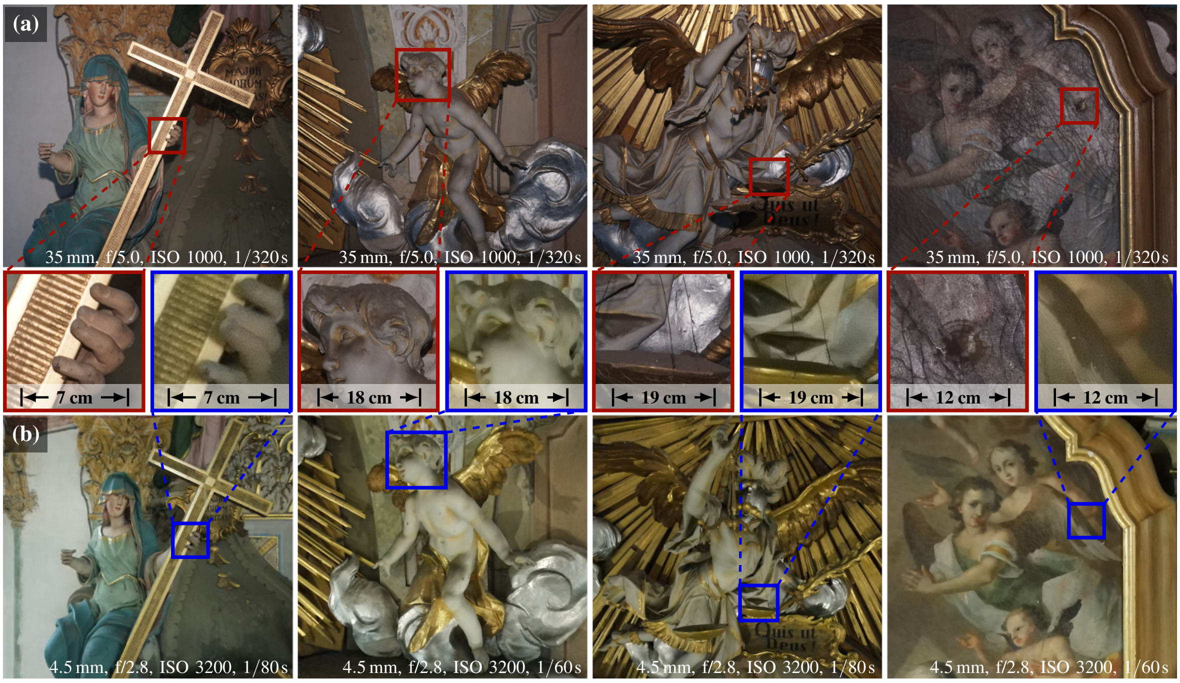

(a)
35 mm, f/5.0, ISO 1000, 1/320 s
35 mm, f/5.0, ISO 1000, 1/320 s
35 mm, f/5.0, ISO 1000, 1/320 s
35 mm, f/5.0, ISO 1000, 1/320 s
7 cm
7 cm
18 cm
18 cm
19 cm
19 cm
12 cm
12 cm
(b)
4.5 mm, f/2.8, ISO 3200, 1/80 s
4.5 mm, f/2.8, ISO 3200, 1/60 s
4.5 mm, f/2.8, ISO 3200, 1/80 s
4.5 mm, f/2.8, ISO 3200, 1/60 s

Fig. 6: Image outputs of VIS methodology as taken by the onboard MIL camera Sony Alpha A6500 (a) and commercial solution DJI Mavic Air 2 (b). Direct comparison of details of the images in the middle row shows that the proposed solution is superior in capturing high-quality details. This highlights the last column in which a hole in the painting is visible in top and absent in bottom image. Although the commercial solution is small and lightweight, its small sensor size of 6.4 × 4.8 mm hinders usability in interior documentation.

system are qualitatively compared to the ones obtained with a commercial product DJI Mavic Air 2. The figure highlights the superior performance of MIL camera imaging allowing for capturing high-resolution details of the OoIs while maintaining a safer distance from the obstacles.

Although VIS realized by a single UAV is a powerful technique, a multi-robot approach is often unavoidable if the lighting conditions are insufficient or documentation of an OoI requires non-direct lighting. An example OoI requiring additional lighting is the mural of St. Christopher in the late Gothic Church of St. Maurice in Olomouc, the documentation of which is shown in Fig. 7. Insufficient external lighting on the mural did not allow capturing bright, high-quality images without the motion blur effect arising from deviations in the reference pose over a long exposure time. Thus, to improve the quality of the images, a secondary UAV provides side lighting (approximately 45° with respect to the camera optical axis), lowering exposure times and highlighting details on the mural, such as small crevices invisible to the human eye from the ground. In the same church, 23 stained-glass windows (each about 8–34 m^2 large) were able to be documented with a single UAV as the windows were well illuminated by the outdoor light and could be captured with short exposure times without additional lighting. The individual images of the mural and the stained-glass windows were rectified and stitched together to compose singular high-resolution orthophotos of each object. The orthophotos were used to assess the state of the OoIs for subsequent restoration works and for enhancing the texture of the 3D model of the church[4]. As compared well in [13], the aerial-based orthophotos outperform the ground-based orthophotos in terms of quality of detail, quality of rectification due to perpendicular optical angles, and absence of occlusions.

B. *Reflectance Transformation Imaging*

RTI method requires a static camera and a dynamic light with a known history of poses. To validate whether the proposed system is feasible for RTI, it was applied to document a vault located 11 m above ground in St. Anne and St. Jacob the Great Church in Stará Voda (see Fig. 1b). This OoI was specifically selected as it can be photographed from a balcony on the opposite side of the central nave, thus allowing for the realization of the RTI technique in two comparable configurations: 1) with the camera (with telephoto lens) mounted on a static tripod with a clear, but misaligned view on the vault and 2) with the camera mounted on board the primary UAV. In both configurations, the light was carried on board the secondary UAV, with the directions of illumination being derived from the poses of this UAV, as estimated on board during the flight.

The comparison of results obtained in each configuration is presented in Fig. 8. The image representation produced from images captured by the tripod-mounted camera yields

[4] Selected OoIs and mapping and 3D reconstruction examples of documented historical structures can be found at mrs.felk.cvut.cz/3d-model-viewer.

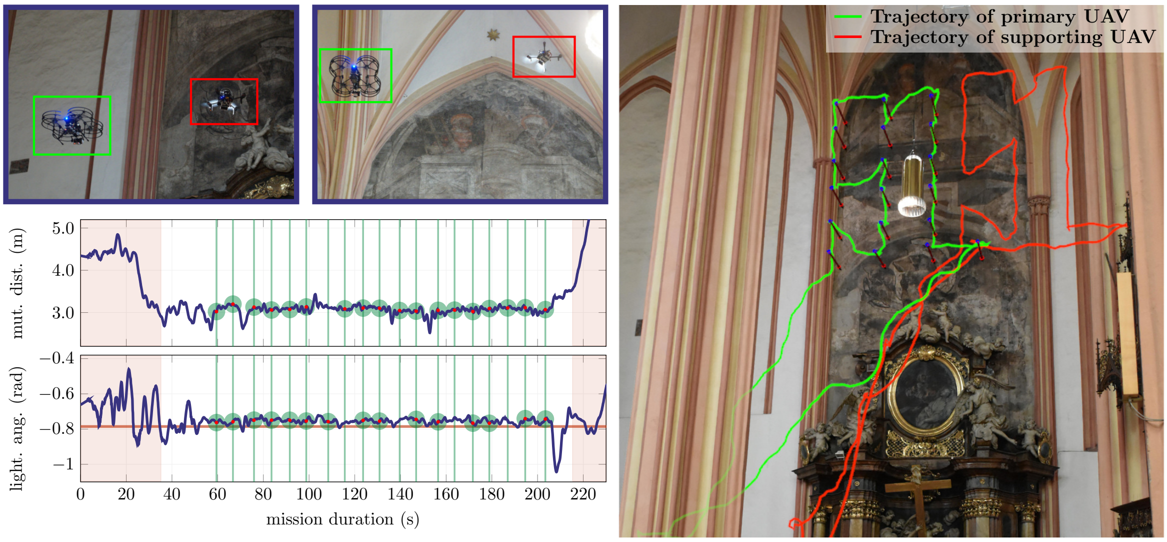


Fig. 7: Deployment of a multi-robot formation for detailed documentation of the late Gothic mural of St. Christopher using additional lighting for enhancing the quality of gathered data. The graphs show mutual distance of the UAVs during the cooperative flight and the angle between the camera optical axis and light, together with the time occasions of image capturing (green lines and circles). The red horizontal line denotes the required angle of lighting. The red areas mark parts of the mission in which the UAVs are not required to maintain the formation.

higher quality, as the choice of the OoI and usage of a telephoto lens fully compensate for the main disadvantages of the methodology in this particular case. These disadvantages are primarily smaller operational space, lower detail resolution of the resulting image caused by the large distance of the camera from the OoI, and often unavoidable occlusions. Although the fully UAV-based approach yields lower image quality since the camera's pose is not static over time, it has wider operational space and enables imaging from appropriate angles, as was verified for other OoIs in the church that could not be reasonably captured by a static camera at all. The non-staticity of the camera's reference pose misaligns the images; thus, their sub-pixel post-alignment is required to avoid blur in the resulting PTM. The experiment shows that the fully UAV-based approach yields comparable results to the single-UAV approach, which is favorable when the OoI can be photographed from the ground — an impossible scenario for most OoIs in difficult-to-reach areas of historical buildings.

C. *Raking Light and Environmental Monitoring*

A common feature of raking light documentation and monitoring of environmental conditions with UAVs stands in the need for robot-environment interaction. In the former, a light is attached to the wall illuminating a planar OoI from a direction perpendicular to the optical axis of the camera. This method is known to highlight even the smallest crevices and cracks in the planar surface. For the latter, a wireless sensor (e.g., for measuring humidity or temperature) is attached to the wall to measure the environmental conditions over longer periods of time. For the purpose of physical environment-UAV interaction itself, we researched a UAV equipped with a system for admittance-based control allowing for stabilization while being attached to a planar surface (and possibly interacting with it) [12]. Before using this technology, the involved risks must be compared to the payoff, particularly inside historical buildings. To minimize the risks, it is more convenient to interact with structural (not artistic) parts of the buildings. The system was successfully tested in real-world mock-up scenarios (see Fig. 9c) with walls of sufficiently good condition.

D. *IR and UV Photography*

Realization of UVF and IRF (fluorescent photography) is methodically similar to VIS with the equipment being a standard MIL camera and a source of light at appropriate frequency. In contrast to VIS, the light emitted by the object illuminated by an IR or UV light source in the visible spectrum is lower. Thus, these methods require higher exposure times, as specified in Table II. The higher exposure times put stricter requirements on image stabilization in the presence of onboard vibrations, inaccuracies, and disturbances that cause UAVs to deviate from their reference pose.

Realization of the UV and IR reflectography requires a camera without UV and IR filters and exposure times of tens of seconds. This makes the use of UAVs for imaging in UV and IR reflectography unfeasible. However, supporting ground-based imaging with aerial lighting is applicable. The UAVs can carry (relatively close to the OoI) high-power LEDs radiating in the desired spectrum. The IR and UV-based methods were tested in St. Anne and St. Jacob the Great Church, Stará Voda (see Fig. 9) and in Church of the Holy Trinity, Běhařovice. The experiments showed that the proposed system can be used in realization of the UV and IR-based methods in historical structures, even in limited lighting conditions.

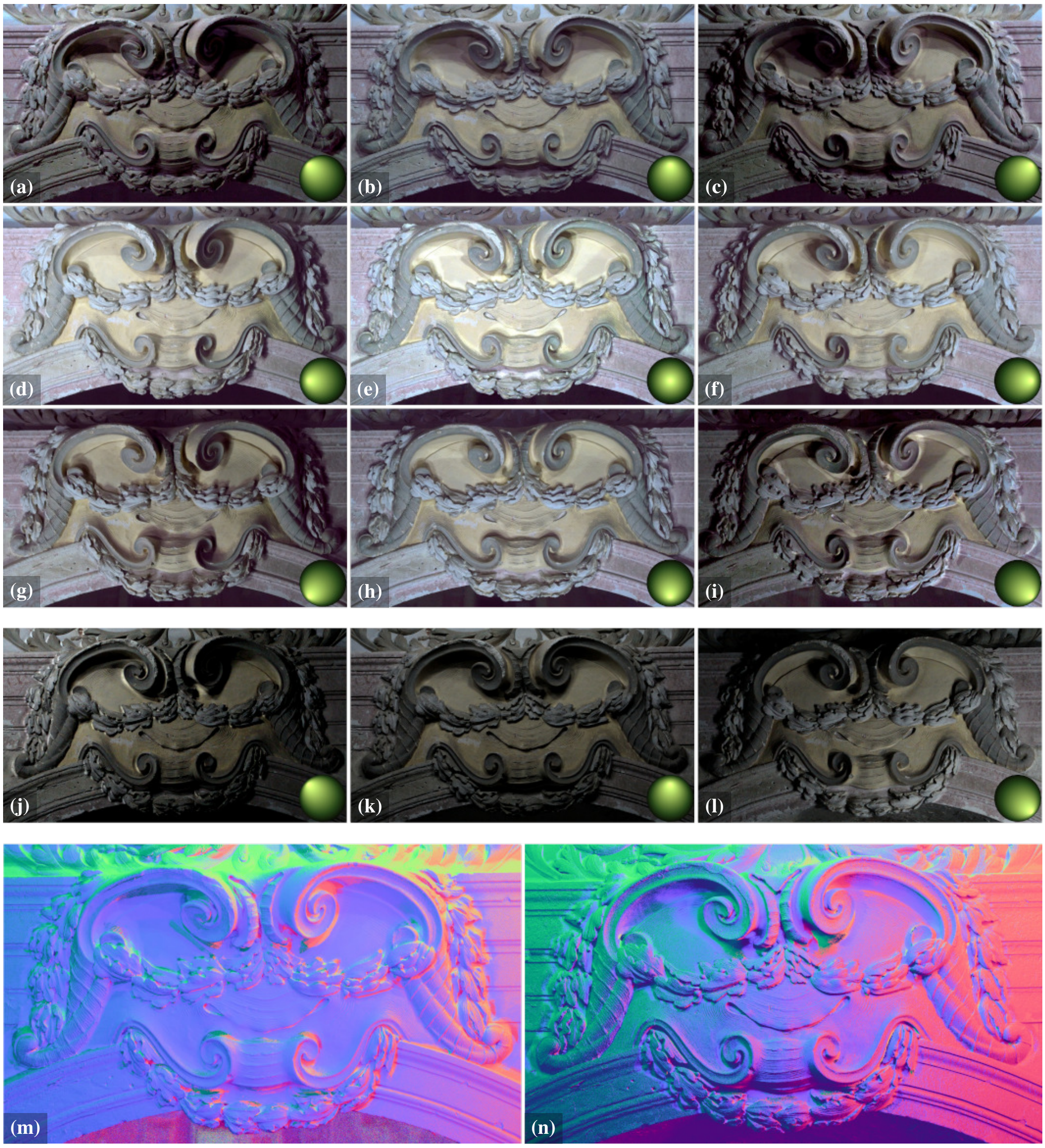


Fig. 8: Comparison of polynomial texture maps (PTM) obtained with a fully UAV-based RTI approach with camera carried by a UAV (a)–(i) and PTM obtained from images taken by a camera mounted on a static tripod (j)–(l). In both cases, the dynamic positioning of light is provided by the secondary UAV. The bottom row shows the normal maps encoded in RGB for fully UAV-based approach (m) and a single UAV approach (n).

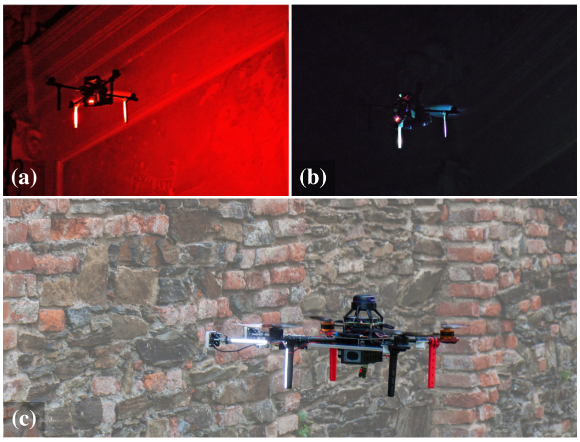


Fig. 9: Deployment of UAVs carrying IR (a) and UV (b) source of light, and a frame-extension mechanism for physical attachment and interaction with static planar surfaces (c).

E. Mapping and 3D Reconstruction

The capacities of UAVs allow capturing the interior under difficult-to-reach angles, not only for imaging purposes, but also for spatial mapping of the structures. Although terrestrial laser scanners yield the most accurate maps, these devices cannot, in principle, document occluded spaces, whereas the larger operational space of UAVs allows for minimizing these occlusions. This advantage is showcased in Fig. 10 where above-ledge areas could not be reconstructed from scans captured at ground. The potential for accurate 3D mapping using UAVs is immense; however, is not the main purpose of the proposed system which outputs dense 3D maps only as a byproduct to the photo-documentation task. The onboard-UAV-built maps contain larger amounts of noise as the mobile laser-scanning technology is less accurate (lightweight, low-power, and moving while scanning) than static scanners, making it harder to align the captured scans, even in post-processing. To achieve the best results for 3D reconstruction, we recommend leveraging the advantages of both methodologies simultaneously.

IX. Discussion

The proposed UAV-based system for documenting historical monuments of differing structures, dimensions, and complexity has demonstrated its wide applicability in real-world documentation tasks, ranging from RGB photography and 3D mapping to multi-robot RTI in areas high above the ground. The high level of autonomy, the ability to fly beyond the visual line of sight between the UAV and a human operator, and the deployability in low lighting conditions (using a worldwide unique method of dynamic illumination by a cooperating UAV team) enable to gather crucial data for heritage protection and documentation that was not possible before. This universally novel system has been used in the very first fully-autonomous multi-robot real-world deployments in such complex and safety-demanding interior structures.

However, deploying mobile robots inherently poses risks to the environment, humans, and equipment therein. This requires careful justification of the UAVs' use that, in our experience, tends to be needlessly overused — conventional technology provides a safer and better quality solution in many documentation tasks. A common example is imaging the interior ceiling or low-height OoIs, where using a static camera with a long-focus lens was identified to be a more appropriate solution. Manual-control UAV solutions are also sufficient if the task is small-scale, the lighting conditions are feasible, repeatability is not required, and the OoIs are few. The need for multi-UAV teams in tasks achievable with sufficient quality by a single UAV, such as the selected example of single-UAV RTI presented in Fig. 8, should also be considered prior a full-scale deployment.

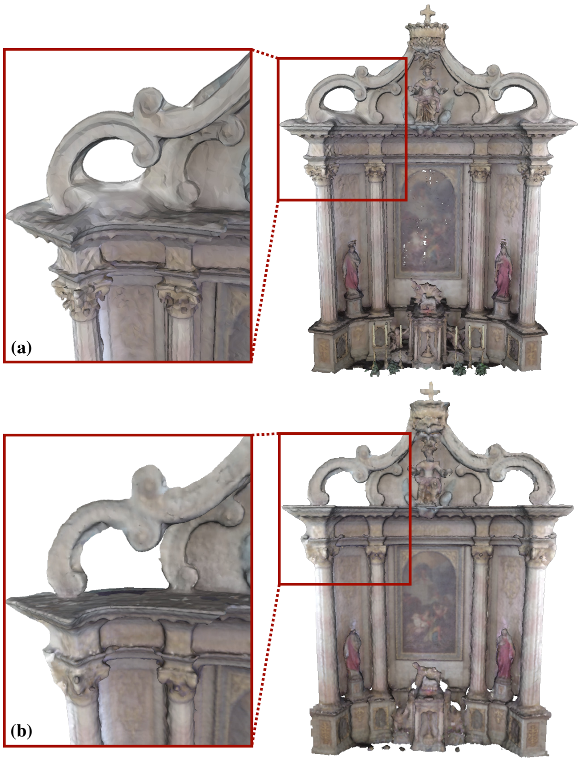


Fig. 10: 3D reconstruction of the altar at the Church of the Nativity of the Virgin Mary in Nový Malín, Czech Republic. The altar reconstructions were done using scans obtained by (a) terrestrial laser scanner Leica BLK360 and (b) Ouster OS0-128 mounted on board an autonomous UAV during the deployment shown in Fig. 5. The meshes were created with the Poisson surface reconstruction and colored using the panoramic RGB images captured by the terrestrial scanner.

X. Conclusion

This work has presented a universally novel study on an autonomous multi-robot UAV-based system for realization of advanced documentation techniques in culturally valuable environments. The system showcases the immense potential of mobile robots for fast, accurate, and mobile digitalization of difficult-to-access interiors. The hardware and software architectures of the self-contained autonomous-UAV-based system were introduced and experimentally validated through almost twelve hours of flight time in more than two hundred real-world flights of single-UAVs and multi-UAV teams in fifteen historical monuments of varying structures. The system design has emerged from close cooperation with a team of restorers, and the data collected during the autonomous missions has been used by the end users in successive restoration works.

The study also assists in identifying the current challenges and future directions of research in aerial documentation and inspection. Based on the high added value for heritage protection, the system has been approved by the Czech National Heritage Institute for indoor usage and is accompanied by an official methodology (available at [1]) describing the proper usage of UAVs in historical structures. It is the first methodology of this authority for using UAVs in historical buildings and so prescribes the system to be a standard in this application.

XI. Acknowledgment

This work was supported by the Ministry of Culture of the Czech Republic through project no. DG18P02OVV069 in program NAKI II, by the Ministry of Education of the Czech Republic through OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 ”Research Center for Informatics,” by the European Union’s Horizon 2020 research and innovation program AERIAL-CORE under grant agreement no. 871479, by CTU grant no. SGS20/174/OHK3/3T/13, and by the Czech Science Foundation (GAČR) under research project No. 20-10280S.

The authors would like to thank Nicolas Staub for an initial survey of the projects’ applicability in documentation and restoration, Jan Bednar for his assistance in creating 3D models for public presentation, Pavel Stoudek for preparing hardware platforms used during the experimental analyses, and Vojtech Krajicek for technical consultation of the restoration techniques. At last, we would like to thank to representatives of the Czech National Heritage Institute, namely Milan Skobrtal and Michaela Cadilova, for our fruitful cooperation during the project and for the opportunity to deploy the proposed system in real-world structures.

References

[1] Multi-Robot Systems group, CTU FEE. (2022) Dronument (Dron & Dronument). [Online]. Available: http://mrs.felk.cvut.cz/dronument

[2] S. Prieto, B. Quintana, A. Adan, and A. Vazquez, “As-is building-structure reconstruction from a probabilistic next best scan approach,” *Robotics and Autonomous Systems*, vol. 94, pp. 186–207, 2017.

[3] P. S. Blaer and P. K. Allen, “Data acquisition and view planning for 3-D modeling tasks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 417–422.

[4] D. Borrmann, R. Heß, H. R. Houshiar, D. Eck, K. Schilling, and A. Nüchter, “Robotic mapping of cultural heritage sites,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W4, pp. 9–16, 2015.

[5] T. Bakirman, B. Bayram, B. Akpinar, M. F. Karabulut, O. C. Bayrak, A. Yigitoglu, and D. Z. Seker, “Implementation of ultra-light uav systems for cultural heritage documentation,” *Journal of Cultural Heritage*, vol. 44, pp. 174–184, 2020.

[6] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, “Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017.

[7] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, “Fast autonomous flight in warehouses for inventory applications,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3121–3128, 2018.

[8] M. Ceccarelli, D. Cafolla, G. Carbone, M. Russo, M. Cigola, L. J. Senatore, A. Gallozzi, R. Di Maccio, F. Ferrante, F. Bolici, S. Supino, N. Colella, M. Bianchi, C. Intrisano, G. Recinto, A. Micheli, D. Vistocco, M. R. Nuccio, and M. Porcelli, “HeritageBot Service Robot assisting in Cultural Heritage,” in *IEEE International Conference on Robotic Computing*, 2017, pp. 440–445.

[9] N. Hallermann, G. Morgenthal, and V. Rodehorst, “Vision-based monitoring of heritage monuments — Unmanned Aerial Systems (UAS) for detailed inspection and high-accurate survey of structures,” *WIT Transactions on The Built Environment*, vol. 153, pp. 621–632, 2015.

[10] P. Petracek, V. Kratky, and M. Saska, “Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2078–2085, 2020.

[11] V. Kratky, P. Petracek, V. Spurny, and M. Saska, “Autonomous Reflectance Transformation Imaging by a Team of Unmanned Aerial Vehicles,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020.

[12] D. Smrcka, T. Baca, T. Nascimento, and M. Saska, “Admittance Force-Based UAV-Wall Stabilization and Press Exertion for Documentation and Inspection of Historical Buildings,” in *International Conference on Unmanned Aircraft Systems*, 2021, pp. 552–559.

[13] V. Kratky, P. Petracek, T. Nascimento, M. Cadilova, M. Skobrtal, P. Stoudek, and M. Saska, “Safe Documentation of Historical Monuments by an Autonomous Unmanned Aerial Vehicle,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, pp. 738/1–16, 2021.

[14] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[15] V. Kratky, P. Petracek, T. Baca, and M. Saska, “An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments,” *Journal of Field Robotics*, vol. 38, no. 8, pp. 1036–1058, 2021.

[16] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 2021.

[17] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The Quickhull Algorithm for Convex Hulls,” *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

[18] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[19] M. Saska, V. Kratky, V. Spurny, and T. Baca, “Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control,” in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2017, pp. 1–8.

[20] D. Hert, T. Baca, P. Petracek, V. Kratky, V. Spurny, M. Petrlik, M. Vrba, D. Zaitlik, P. Stoudek, V. Walter, P. Stepan, J. Horyna, V. Pritzl, G. Silano, D. Bonilla Licea, P. Stibinger, R. Penicka, T. Nascimento, and M. Saska, “MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments,” in *IEEE International Conference on Unmanned Aircraft Systems*, 2022, pp. 1264–1273.

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2021. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2021.3098811

Autonomous Aerial Filming with Distributed Lighting by a Team of Unmanned Aerial Vehicles

Vít Krátký[1], Alfonso Alcántara[2], Jesús Capitán[2], Petr Štěpán[1], Martin Saska[1] and Aníbal Ollero[2]

***Abstract*—This paper describes a method for autonomous aerial cinematography with distributed lighting by a team of unmanned aerial vehicles (UAVs). Although camera-carrying multirotor helicopters have become commonplace in cinematography, their usage is limited to scenarios with sufficient natural light or of lighting provided by static artificial lights. We propose to use a formation of unmanned aerial vehicles as a tool for filming a target under illumination from various directions, which is one of the fundamental techniques of traditional cinematography. We decompose the multi-UAV trajectory optimization problem to tackle non-linear cinematographic aspects and obstacle avoidance at separate stages, which allows us to re-plan in real time and react to changes in dynamic environments. The performance of our method has been evaluated in realistic simulation scenarios and field experiments, where we show how it increases the quality of the shots and that it is capable of planning safe trajectories even in cluttered environments.**

***Index Terms*—Multi-Robot Systems, Aerial Systems: Applications, Motion and Path Planning**

I. INTRODUCTION

THE interest in *Unmanned Aerial Vehicles* (UAVs) for aerial photography and filming is growing fast [1]–[5]. This is mainly due to their manoeuvrability and the capacity to create unique shots when compared to standard cameras. The use of UAVs as flying cameras presents not only a remarkable potential for recreational cinematography, but also for the monitoring of inspection operations in outdoor infrastructures with complex access. For instance, the EU-funded project, AERIAL-CORE, proposes UAVs to surveil the safety of human workers during maintenance operations of electrical power lines (see Fig. 1). In this industrial setup, a high-quality video is key, as it is used by supervising ground operators to monitor safety during the maintenance work. Multi-UAV teams expand upon these possibilities as they could provide alternative points of view or even supplementary illumination. Similarly in our DRONUMENT project of NAKI II program, efficient variable illumination plays a key role for documentation of historical buildings interiors.

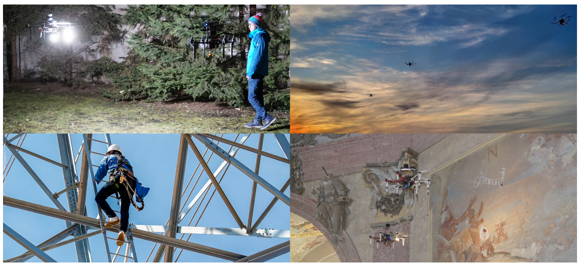


Fig. 1: UAV filming applications to provide external lighting; to capture smooth shots outdoors; and to monitor dangerous maintenance operations at electrical lines[1]. Pictures were obtained within AERIAL-CORE and DRONUMENT projects, for which the proposed technology is being developed. Videos of the work in this paper can be seen on the multimedia material page (http://mrs.felk.cvut.cz/papers/aerial-filming).

Proper lighting techniques are fundamental in bringing out details in an image and in creating more natural-looking film scenes. Thus, cinematography sets are packed with different lighting sources, as digital sensors are not as reactive to light as the human eye. This can also be relevant in monitoring maintenance operations scheduled at times of the day with poor illumination. Although aerial cinematography has been attractive to the scientific community as of late, lighting techniques have yet to be applied to improve the performance of filming. Filmmakers apply many types of lighting techniques making use of various equipment. In this work, we only consider direct lighting techniques that do not require additional equipment apart from light sources. Although static lights could produce more pleasant footage in some situations, we believe that UAVs are not optimal for this purpose. Therefore, we only use UAVs as dynamic sources of light to provide lighting to a dynamic scene.

In this context, navigating a team of UAVs for filming tasks with distributed lighting is complex. Smooth and safe trajectories are required to achieve pleasing shots that do not compromise safety in dynamic scenarios. We propose

Manuscript received: February 24, 2021; Revised May 26, 2021; Accepted July 3, 2021.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by EU project AERIAL-CORE (H2020-2019-871479), by MULTICOP (US-1265072) in FEDER-Junta de Andalucia Programme, by project no. DG18P02OVV069 in program NAKI II, by CTU grant no SGS20/174/OHK3/3T/13, and by OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics". *(Vít Krátký and Alfonso Alcántara are co-first authors.)*

[1] Vít Krátký, Petr Štěpán, and Martin Saska are with Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic, {kratkvit|stepan|martin.saska}@fel.cvut.cz.

[2] Alfonso Alcántara, Jesús Capitán, and Aníbal Ollero are with GRVC Robotics Laboratory, University of Seville, Spain {aamarin|jcapitan|aollero}@us.es.

Digital Object Identifier (DOI): see top of this page.

[1] https://aerial-core.eu

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2021. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2021.3098811

a method for online trajectory planning and execution with multiple UAVs. Our team obeys a leader-follower scheme where the formation leader carries an onboard camera to film a moving target and the followers generate trajectories that enable distributed lighting of the target, while maintaining desired lighting angles. We formulate a non-linear, optimization-based method that plans visually pleasant trajectories for the filming UAV and distributes the others in a specified formation. Moreover, we tackle safety by including a systematic framework for obstacle avoidance. Safe flight corridors for the UAVs are generated by forming sets of convex polyhedrons that model free space. Optimal and safe trajectories are thereafter computed within these convex sets.

A. *Related works*

There have been several works focusing on filming dynamic scenes with a single UAV. Commercial products exist (e.g., *DJI Mavic* [6] or *Skydio* [7]) that implement semi-autonomous functionalities, such as *auto-follow* features for tracking an actor with collision avoidance. However, they do not address cinematographic principles. An integrated system for outdoor cinematography combining vision-based target localization with trajectory planning and collision avoidance has been proposed [8], [9]. Smoothness is achieved by minimizing trajectory jerk; shot quality by defining objective curves that fulfill relative angles with respect to the actor. Optimal trajectories for cinematography have also been computed in real-time through receding horizon optimization with non-linear constraints [10]. A user inputs framing objectives for the targets on the image to minimize errors on the image target projections, sizes, and relative viewing angles. Some authors have approached UAV cinematography by applying machine learning [11], [12]. Particularly, such techniques have been applied to demonstrations imitating professional cameraman's behaviors [13] or for reinforcement learning to achieve visually pleasant shots [14]. These works have presented valuable results for online trajectory planning, although they have not addressed the specific complexities for multi-UAV systems.

Regarding the methodology for multiple UAVs, a non-linear optimization problem was solved in receding horizon in [15], where collision avoidance to filmed actors and mutual collisions of UAVs were considered. Aesthetic objectives are introduced by the user as virtual reference trails. A specific camera parameter space is proposed in [16] to ensure cinematographic properties and to fulfill dynamic constraints along the trajectories. The motion of multiple UAVs around dynamic targets is coordinated through a centralized master-slave approach. A greedy framework for multi-UAV camera coordination is proposed in [17]. A decentralized planner computes UAV trajectories considering smoothness, shot diversity, collision avoidance, and mutual visibility. We have also addressed the trajectory planning for multi-UAV cinematography in previous work. We presented an architecture to execute cinematographic shots (with different types of camera motion) using multiple UAVs [18] and developed a distributed method to plan optimal trajectories reducing jerky camera movements [19]. In this paper, our focus is on the specifics of outdoor and dynamic settings when compared to indoor scenarios [15]. Therefore, we have integrated obstacle avoidance in a more fundamental manner using local maps. Moreover, a novel problem with respect to previous work has been introduced, as we perform scene illumination with multiple UAVs to increase the quality of image shots.

The modification of lighting angles to improve images is fundamental in cinematography [20]. A camera with an onboard light on a UAV can compensate for insufficient lighting, but positioning lights at different angles with respect to the camera axis would require several UAVs. Despite the unquestionable importance of lighting for shot quality, its usage for aerial cinematography is not well-studied. Utilizing UAVs to provide supplementary illumination has been proposed for building documentation tasks [21] or tunnel inspection [22]. A formation with a filming UAV and others carrying lights was deployed to document the overshadowed parts of historical buildings [23]. A similar system has been used to carry out specialized documentation techniques [24]. However, these works have proposed lighting for tasks in static scenes, whereas the present paper deals with filming of moving targets in dynamic and potentially cluttered environments, e.g., to monitor inspection operations in large outdoor infrastructures.

In order to guarantee safe trajectories in multi-UAV cinematography, most works [15], [16], [19] only consider collision avoidance with actors, other UAVs, or static objects that can be modelled with previously known no-fly zones. The work in [9] integrates local mapping with onboard sensors to penalize proximity to obstacles and solves an unconstrained optimization problem. Another approach to obstacle avoidance applied for standard UAV trajectory planning is to create a convex representation of free space via a set of linear inequality constraints [25]–[28], to obtain a QP formulation for real-time motion planning. We have been inspired by these single-UAV works to develop a fundamental framework for the representation of obstacles in our non-linear optimization problem for multi-UAV cinematography.

B. *Contributions*

Our main contributions are summarized as the following:

- We formulate a novel optimization problem for aerial filming with distributed lighting. Using a leader-follower scheme, we plan and execute trajectories in a distributed manner. Optimization is run in receding horizon to compute smooth trajectories with pleasing footage for the UAV filming (the leader), which takes shots of a dynamic target indicated by an external user. The followers compute their trajectories to maintain a formation with specified lighting angles on the target.
- We propose a new method to tackle non-convex trajectory optimization with obstacle avoidance in real time. We decompose the problem in two parts. Non-linear cinematographic aspects are formulated in a problem without obstacle avoidance to generate reference trajectories. These are used to generate collision-free regions which are convex and to transform the problem into a final QP optimization task.

- We present experimental results for different types of cinematographic shots. We prove that our method is capable of computing smooth trajectories for reducing jerky movements and show that the distributed formation improves the illumination of footage. The system is evaluated with field experiments and also in various realistic simulated scenarios, including the filming of a moving target in a cluttered environment.

II. SYSTEM OVERVIEW

Fig. 2 depicts the architecture of the entire system. The leader UAV carries a camera for filming while several others carry light sources to provide proper illumination. A human director specifies the cinematographic parameters for the scene. These parameters include the shot type (i.e., the camera motion relative to the target), the camera shooting angle for the leader, and the desired lighting angles for the followers. This information, together with an estimation of the target trajectory, is used to generate reference trajectories for the UAVs (Section III-B). These initial trajectories do not consider obstacle avoidance, but only cinematographic aspects. The leader attempts to execute the commanded shot smoothly, whereas the followers maintain a surrounding formation with the desired lighting angles.

Safety is ensured by integrating information from a local map for collision avoidance (Section III-C). Firstly, a collision-free path is generated for each UAV using the map and the initial cinematographic trajectories as guidelines. Then, a safe corridor along each of these paths is computed, consisting of a set of obstacle-free polyhedrons generated by the convex decomposition of free space (see Fig. 5). Finally, the UAV trajectories are obtained as a result of a trajectory optimization process that computes dynamically feasible trajectories inside each safe corridor (Section III-D). Inter-UAV collision avoidance is achieved by including the team-mates planned trajectories as obstacles in the map.

The entire pipeline shown in Fig. 2 (except for the *Human director* component) runs on board each UAV in a receding horizon manner. This enables the online planning to react properly to changes in the behavior of the target being filmed, as well as to malfunctioning team-members or previously unseen obstacles. Note that either the *Cinematographic trajectory generator* or the *Lighting trajectory generator* is activated on each UAV, depending on whether it carries a camera or light. The component for trajectory tracking on each UAV is the low-level control pipeline described in [29].

III. AUTONOMOUS AERIAL CINEMATOGRAPHY

In this section, we begin by detailing the UAV dynamic model (Section III-A). Then, we describe our procedure to generate optimal and safe trajectories for each UAV (Sections III-B, III-C, and III-D). Lastly, we explain how the orientation of a UAV is controlled (Section III-E).

A. Multi-rotor aerial vehicle dynamic model

An independent trajectory tracker [29] for UAV attitude control is used, which allows for planning with a simplified positional dynamic UAV model. In addition, the orientation of the camera or light source onboard (depending on the UAV) needs to be modelled. We assume the existence of a gimbal mechanism to compensate angle deviations due to changes in UAV attitude. Therefore, it is assumed that camera roll is negligible and we only control pitch and heading. Since the heading of a multi-rotor vehicle can be controlled independently of its position, we fix the relative position between the camera/light and the UAV to always point forward and control its heading through the UAV heading. The positional part of the dynamic model is defined as a linear double integrator:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= \mathbf{a}, \end{aligned} \tag{1}$$

where $\mathbf{p} = [p_x \ p_y \ p_z]^T \in \mathbb{R}^3$ is the UAV position, $\mathbf{v} = [v_x \ v_y \ v_z]^T \in \mathbb{R}^3$ the linear velocity, and $\mathbf{a} = [a_x \ a_y \ a_z]^T \in \mathbb{R}^3$ the linear acceleration. The orientation of the camera/light may be modelled similarly:

$$\begin{aligned} \dot{\mathbf{o}} &= \boldsymbol{\omega}, \\ \dot{\boldsymbol{\omega}} &= \boldsymbol{\theta}, \end{aligned} \tag{2}$$

where $\mathbf{o} = [\varphi \ \xi]^T$ represents an orientation with respect to a global frame given by its heading and pitch angles, $\boldsymbol{\omega} \in \mathbb{R}^2$ are the corresponding angular rates, and $\boldsymbol{\theta} \in \mathbb{R}^2$ the angular accelerations. For the description of the proposed method, we define a full positional state of the UAV $\mathbf{x}_p = [\mathbf{p}^T \ \mathbf{v}^T]^T \in \mathbb{R}^6$, a vector of positional control inputs $\mathbf{u}_p = \mathbf{a}$, an orientation state $\mathbf{x}_o = [\mathbf{o}^T \ \boldsymbol{\omega}^T]^T \in \mathbb{R}^4$, and a vector of orientation control inputs $\mathbf{u}_o = \boldsymbol{\theta}$.

B. Generation of reference trajectories

The first step of our method for trajectory planning is to generate a reference trajectory D_j for each UAV j. The problem complexity is alleviated by removing collision avoidance constraints and focusing only on cinematographic aspects. For the filming UAV, the objective is to reach a position relative to the target as provided by the shot type C_s, while minimizing changes in the camera angle to produce pleasant images. A specific camera shooting angle ψ_d over the target needs to be maintained. The following non-linear optimization problem is formulated[2] for the filming UAV:

$$\underset{\mathbf{u_0},\ldots,\mathbf{u_{N-1}}}{\text{minimize}} \sum_{k=1}^{N} (||\mathbf{u}_{k-1}||^2 + \alpha_1 J_{\psi,k}) + \alpha_2 J_N, \tag{3}$$

$$\text{subject to } \mathbf{x}_0 = \mathbf{x}', \tag{3.a}$$

$$\mathbf{x}_{k+1} = \mathrm{f}_p(\mathbf{x}_k, \mathbf{u}_k) \quad \forall k \in \{0, \ldots, N-1\}, \tag{3.b}$$

$$\mathbf{v}_{min} \le \mathbf{v}_k \le \mathbf{v}_{max} \quad \forall k \in \{1, \ldots, N\}, \tag{3.c}$$

$$\mathbf{u}_{min} \le \mathbf{u}_k \le \mathbf{u}_{max} \quad \forall k \in \{0, \ldots, N-1\}, \tag{3.d}$$

$$q_{z,min} \le q_{z,k} \quad \forall k \in \{1, \ldots, N\}, \tag{3.e}$$

where $\mathrm{f}_p(\cdot)$ represents the positional part of the dynamic model defined in Section III-A; $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ are velocity limitations; and $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ control inputs limitations.

The first two terms in the cost function pursue smooth trajectories by penalizing UAV accelerations and reducing

[2] For simplicity of description, $\mathbf{x} := \mathbf{x}_p$, and $\mathbf{u} := \mathbf{u}_p$. We use the Runge-Kutta method for numerical integration.

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2021. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2021.3098811

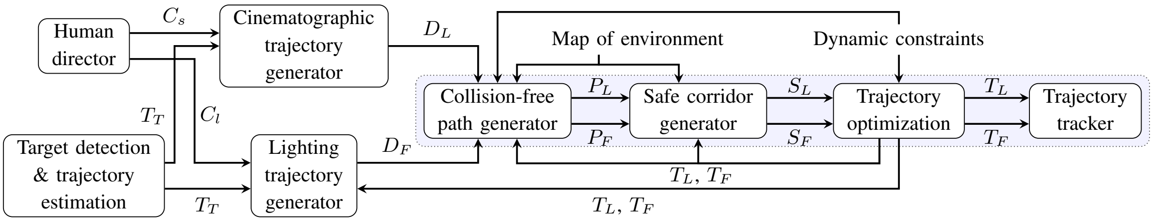


Fig. 2: The architecture of the proposed system. C_s and C_l represent the desired type of cinematographic shot and lighting configuration specified by a human director; T_T is the target estimated trajectory; D_L, D_F are reference trajectories for the leader UAV and the follower UAVs, respectively; P_L, P_F are collision-free paths generated along the desired trajectories; S_L, S_F are safe corridors along the collision-free paths; and T_L, T_F are optimized trajectories for the camera and lighting UAVs, respectively. The modules enclosed in the blue rectangle run on both types of UAVs.

gimbal movements. The director specifies an aesthetic objective through the desired camera shooting angle ψ_d to film the target (see Fig. 3). Emphasis is given on positioning the UAV to keep this angle constant without moving the gimbal. In doing so, the angular changes in the gimbal are reduced to favor less jerky camera motion and therefore, pleasant footage. In order to define J_ψ, the relative position between the UAV camera and the target is introduced as:

$$\mathbf{q} = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^T = \mathbf{p}_L - \mathbf{p}_T. \tag{4}$$

Then, we define J_ψ as:

$$J_{\psi,k} = \left(\tan(\psi_d) - \frac{q_{z,k}}{\sqrt{q_{x,k}^2 + q_{y,k}^2}} \right)^2. \tag{5}$$

The target position is predicted within the time horizon with a motion model (a constant speed model in our experiments). Prediction errors are tackled by recomputing UAV trajectories with a receding horizon. By minimizing the previous cost, we implicitly minimize variations in camera pitch angle as the relative pitch with respect to the target is kept constant. Moreover, the camera heading corresponds with the UAV heading, whose variations are also smoothed as explained in Section III-D. Therefore, the idea is to generate UAV trajectories where the gimbal only needs to move slightly to compensate for small disturbances.

The terminal cost $J_N = ||\mathbf{x}_{xy,d} - \mathbf{x}_{xy,N}||^2$ guides the UAV to a desired state imposed by the shot type, e.g., at a certain distance beside the target's final position in a lateral shot. Note that a final UAV height is not imposed, as we want the planner to compute the optimal p_z to maintain the camera shooting angle commanded by the director. Lastly, the constraint (3.e) establishes a minimum distance above the target for safety purposes.

The reference trajectories for the lighting UAVs are computed to achieve a desired leader-follower formation around the target. The desired position of the followers is influenced by the corresponding leader position $\mathbf{p}_L$ and camera orientation $\mathbf{o}_L$, the target position $\mathbf{p}_T$, the desired lighting angles of j-th light χ_j and ϱ_j, and the desired distance of the light to the target d_j. The desired position of j-th follower $\mathbf{p}_j$ is then given by the equation:

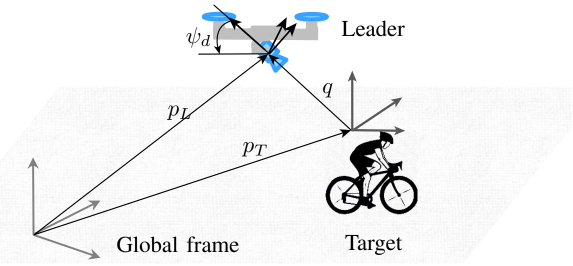


Fig. 3: Reference frames and camera shooting angle. The origins of the camera and UAV frames coincide.

$$\mathbf{p}_j = \mathbf{p}_T + d_j \begin{bmatrix} -\cos(\varphi_j)\cos(\xi_j) \\ -\sin(\varphi_j)\cos(\xi_j) \\ \sin(\xi_j) \end{bmatrix}, \tag{6}$$

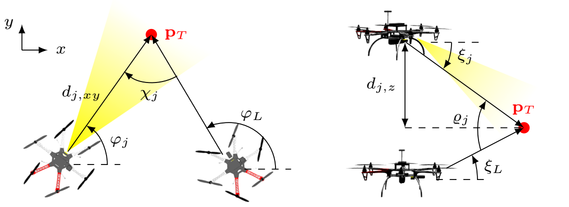
where $\varphi_j = \varphi_L + \chi_j$ and $\xi_j = \xi_L + \varrho_j$ are desired lighting angles relative to the camera's optical axis (see Fig. 4). To avoid jumps in the desired followers' positions caused by quick changes in the target position (e.g., due to a transition to a new target), a virtual target, located in front of the camera at a certain distance along its optical axis, is used. The position of this virtual target is given by:

$$\mathbf{p}_v = \mathbf{p}_L + d_v \begin{bmatrix} \cos(\varphi_L)\cos(\xi_L) \\ \sin(\varphi_L)\cos(\xi_L) \\ \sin(\xi_L) \end{bmatrix}, \tag{7}$$

where d_v is the desired distance between the virtual target and the camera center and $\mathbf{p}_v$ denotes the virtual target position. Substituting position $\mathbf{p}_v$ for $\mathbf{p}_T$ in (6), a more consistent formation scheme is acquired, where less aggressive maneuvers are required; and the lighting always focuses on the scene in front of the camera, which is relevant in obtaining pleasant videos.

C. Generation of safe corridors

The initial reference trajectories are computed without considering obstacles. They are, therefore, used as seed to generate a safe corridor S_j for each UAV j where collision-free trajectories can then be computed. Firstly, we convert each trajectory D_j into a collision-free path P_j. We iterate over each of the N waypoints in D_j and add it directly to P_j if



(a) top view (b) side view

Fig. 4: The leader-follower scheme defined by (6).

it is collision-free. Otherwise, we label the previous collision-free waypoint as A and keep moving along D_j until we find the next collision-free waypoint B. Then, we try to find an alternative collision-free path from A to B, to be appended to P_j and continue iterating. For that alternative path, we use the *Jump Point Search* (JPS) algorithm introduced in [30], [31] and extended to 3D in [27]. A real-time performance is ensured by introducing a timeout for the JPS path search.

If the JPS algorithm fails to find a path within the given timeout from A to B, we run it again to connect A directly to the last waypoint in D_j (let this waypoint be C). If this is not found either, we append to P_j the path to the node closest to C from all those expanded during the JPS search. Once completed, P_j consists of an arbitrary number of points equal to or greater than N. Since P_j is used for the generation of the safety corridors for particular points in D_j, it is post-processed so that $|P_j| = |D_j| = N$. P_j is sampled so that the waypoint distribution is close to the initial points in D_j. Since these collision-free paths are used as a guide for trajectory optimization in subsequent steps, the distance sampling step d_s is limited to help avoid the dynamic infeasibility of the final trajectories. If the sampled P_j consists of more than N waypoints, the overflowing points are discarded for the subsequent steps of the trajectory optimization process. The process to create a collision-free path P_j and its corresponding safe corridor S_j is illustrated in Fig. 5.

Safe corridors are generated around the collision-free paths with a prefixed initial position of the UAV (i.e., $N+1$ waypoints), using a map of the environment represented by a point cloud O_{pcl} and the convex decomposition method proposed in [27]. This method is based on an iterative procedure for the generation of polyhedrons. It begins by inflating an ellipsoid aligned with each path segment. In the next step, tangent planes are constructed at the contact points between the ellipsoid and any obstacles. Afterwards, all points lying behind this plane are removed from O_{pcl}. Yet again, the next iteration starts by inflating the ellipsoid up to the nearest point in O_{pcl}. This procedure is terminated if there are no remaining points in O_{pcl}. The generated tangent planes define an obstacle-free polyhedron $\mathcal{P}$ enclosing the corresponding path segment and the set of all polyhedrons along the path constitutes the safe corridor.

D. Trajectory optimization

Given a collision-free path P and its corresponding safe corridor S, a final optimal trajectory is computed through a QP problem in receding horizon. The particular optimization task [3] attempts to track a desired trajectory $\mathbf{p}_d$ corresponding to the reference trajectory D_j:

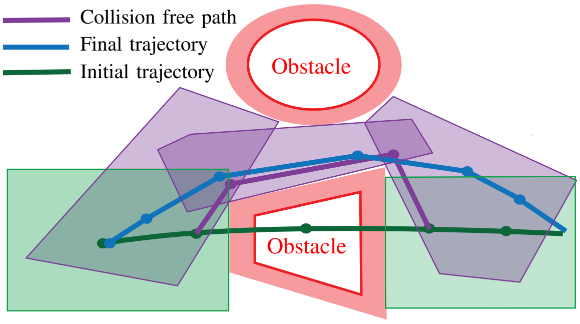


Fig. 5: The safe corridor generation process. The initial reference trajectory (green) is converted into a collision-free path (purple), and the obstacle-free polyhedrons are generated along this path. The final optimized trajectory within the safe corridor is also shown (blue). We inflate the obstacles for safety purposes (light red).

$$\underset{\mathbf{u_0},\ldots,\mathbf{u_{N-1}}}{\text{minimize}} \sum_{k=1}^{N} (||\mathbf{p}_{d,k} - \mathbf{p}_k||^2 + \beta||\mathbf{u}_{k-1}||^2), \tag{8}$$

$$\text{subject to } \mathbf{x}_0 = \mathbf{x}', \tag{8.a}$$

$$\mathbf{x}_{k+1} = \mathrm{f}_p(\mathbf{x}_k, \mathbf{u}_k) \quad \forall k \in \{0, \ldots, N-1\}, \tag{8.b}$$

$$\mathbf{v}_{min} \leq \mathbf{v}_k \leq \mathbf{v}_{max} \quad \forall k \in \{1, \ldots, N\}, \tag{8.c}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \quad \forall k \in \{0, \ldots, N-1\}, \tag{8.d}$$

$$\mathbf{p}_k \in \mathcal{P}_k \quad \forall k \in \{1, \ldots, N\}, \tag{8.e}$$

where $\mathrm{f}_p(\cdot)$ represents the positional part of a dynamic model defined in Section III-A; $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ are velocity limitations; $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ control inputs limitations; and $\mathcal{P}_k$ is a convex polyhedron representing a free space associated with k-th transition point. The last constraint ensures a safe resulting trajectory without collisions. Given that the constraint (8.e) can be decoupled in a set of linear constraints, the problem becomes a quadratic convex program.

The optimization formulation is the same for both the leader and follower UAVs. However, there are a couple of relevant differences. First, the desired reference trajectories are computed in a different manner, following either filming or lighting criteria (see Section III-B). Second, the followers encode mutual-collision avoidance through constraint (8.e). To prevent negative effects on the cinematographic quality of the performed shot, the entirety of mutual collision avoidance is left to the followers. A fixed priority scheme is defined for the UAVs, and the occupied space O_{pcl} of each follower is updated with the current planned trajectories from the leader and other followers of a higher priority. O_{pcl} is updated with spherical objects of the desired collision avoidance radius at each waypoint of the UAV trajectories to be avoided. A similar procedure is followed to incorporate the target's predicted trajectory (also for the leader in this case). To hold with real-time performance, the occupied space O_{pcl} is assumed static for a given horizon time, but it is updated at each planning step, accommodating all static and dynamic obstacles.

[3] For simplicity of description, $\mathbf{x} := \mathbf{x}_p$, and $\mathbf{u} := \mathbf{u}_p$.

Another crucial issue for the applications of multi-UAV cinematography is how to prevent other UAVs from appearing in the *Field of View* (FoV) of the filming UAV. However, including this in the optimization task as either a constraint or a cost term can remarkably increase the complexity of the problem. We considered including the FoV of the leader camera as an obstacle in the local maps of the followers, so that they may avoid it. Even so, relatively small changes in camera orientation could result in significant changes in the map representation and lead to unstable planned trajectories. Therefore, the camera's FoV is avoided by the lighting UAVs only through penalizing deviations from the desired trajectories $\mathbf{p}_d$. Thus, FoV avoidance is mostly determined by the choice of lighting parameters that describe the desired formation.

Finally, occlusions caused by obstacles in the FoV of the camera or the lights are also a relevant aspect when filming. Occlusion throughout a significant part of the shot renders the shot useless, and in the case of onboard detection of the target, it also disables target following. However in most cases, occlusions are temporary and avoiding them is always a trade-off between significant deviation from the desired cinematographic trajectory and having part of the video occluded. In this work, the trajectories are generated so that they are close to the desired cinematographic shots specified by a director. The possible occlusions have to be resolved by redefining the shot to be performed.

E. Orientation control

In this application, both the camera and the light sources need to always be pointing at the filmed target. Hence, their desired orientation is given by:

$$\mathbf{o}_d = \left[\varphi_d \ \xi_d\right]^T = \left[\arctan(q_y, q_x) \ \sin\left(\frac{q_z}{||q||}\right)\right]^T. \quad (9)$$

Orientation control is also formulated as a constrained quadratic optimization problem in receding horizon in order to achieve smoother orientation changes. For simplicity of description, $\mathbf{x} := \mathbf{x}_o$ and $\mathbf{u} := \mathbf{u}_o$ in the following problem formulation:

$$\underset{\mathbf{u_0},\ldots,\mathbf{u_{N-1}}}{\text{minimize}} \ \sum_{k=1}^{N}(||\mathbf{o}_{d,k} - \mathbf{o}_k||^2 + \gamma||\mathbf{u}_{k-1}||^2), \quad (10)$$

$$\text{subject to } \mathbf{x}_0 = \mathbf{x}', \quad (10.a)$$

$$\mathbf{x}_{k+1} = \mathrm{f}_o(\mathbf{x}_k, \mathbf{u}_k) \quad \forall k \in \{0,\ldots,N-1\}, \quad (10.b)$$

$$\boldsymbol{\omega}_{min} \leq \boldsymbol{\omega}_k \leq \boldsymbol{\omega}_{max} \quad \forall k \in \{1,\ldots,N\}, \quad (10.c)$$

$$\xi_{min} \leq \xi_k \leq \xi_{max} \quad \forall k \in \{1,\ldots,N\}, \quad (10.d)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \quad \forall k \in \{0,\ldots,N-1\}, \quad (10.e)$$

where $\mathrm{f}_o(\cdot)$ represents the orientation aspect of the dynamic model defined in Section III-A; ω_{min}, ω_{max} are limitations on the angular velocities; $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ control inputs limitations; and ξ_{min}, ξ_{max} represent hardware limitations of the gimbal to adjusting pitch angles. The heading and pitch angles of the camera or light can be controlled independently. Thus, Problem (10) was decoupled into two simpler problems. The optimal solution for each problem can be found analytically with a standard framework for linear MPC (*Model Predictive Control*).

IV. EXPERIMENTAL EVALUATION

In this section, experimental results are presented to demonstrate the performance of our method for multi-UAV trajectory planning. We have assessed that the proposed method is capable of computing smooth cinematographic trajectories in real-time. Additionally, we have evaluated that the trajectories of the follower UAVs which provide lighting for the target are capable of complying with formation constraints to improve the quality of the shot. The safety of our method has also been proved through experiments in the presence of multiple obstacles.

A. Experimental setup

We implemented our architecture described in Section II in C++ using the ROS framework. The ACADO Toolkit [32] was used to solve the optimization problems. We conducted software-in-the-loop simulations using Gazebo to simulate physics and to equip the UAVs with a camera and lights. To solve the optimization problems, a horizon length of 8 s and a time step of 0.2 s were chosen. The cinematographic parameters were set to $\psi_d = 6°$ and $q_{z,min} = 0.5\,\mathrm{m}$. The maximum distance sampling step was set to $d_{s,max}$ = 0.5 m.

B. Simulation - Cinematography trajectories

The objective of this simulation was twofold: to demonstrate how the method computes smoother camera trajectories for the leader UAV while complying with cinematographic aspects, and how the trajectories of the followers keep with formation constraints to light the target properly. We simulated a human worker performing a maintenance operation on a transmission tower while monitored by a team of three UAVs (one filming and two lighting the worker). While the worker approached and climbed the tower, the system was commanded to perform a lateral shot followed by a sequence of fly-over shots.

The fly-over shots were selected to film the operation as they impose relative motion between the camera and the target. This feature is regarded as richer from a cinematographic point of view. We further demonstrate how our method is able to execute these relative movements more aesthetically than a baseline approach where the specific term to smooth variations in camera angles has been removed (i.e., $\alpha_1 = 0$ in Problem 3). Fig. 6 compares the trajectories for the camera carrying UAV generated with both our method and the baseline approach. The baseline approach generates straight trajectories, whereas our method results in orbital trajectories, which have been used in the cinematography literature to produce more pleasant videos. For instance, [16], [17], [33] apply the arcball principle [34] to create a spherical surface around the target for aesthetic camera motion. We can also see in Fig. 6 that our method reduces the jerk of the camera angles. Note that in aerial cinematography literature, the jerk of the camera motion (third derivative of the angles) has been identified as a key aspect for shot quality [9], [35]. We measured the root mean

© IEEE ROBOTICS & AUTOMATION SOCIETY (RAS), 2021. ACCEPTED TO IEEE RA-L. DOI: 10.1109/LRA.2021.3098811

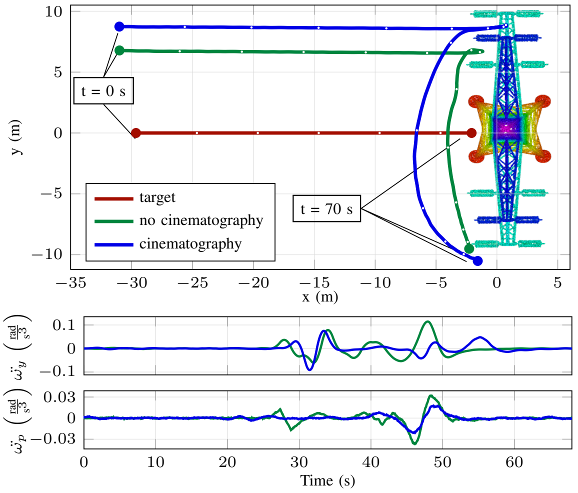


Fig. 6: Trajectories for the camera carrying UAV while monitoring a worker on a transmission tower. For simplicity, only the lateral shot and the first fly-over shot are shown. We compare the trajectories generated by our method (blue) with those from a baseline approach without cinematographic costs (green). The upper image displays a top view of the UAV's and target's trajectories. The small white dots on the trajectories depict transition points sampled every 5 s to give a notion of the speed. The bottom image depicts the temporal evolution of the jerk of the camera angles

.

square of the jerk of φ and ξ along the full trajectories and obtained 0.0197 rad s^{-3} and 0.0048 rad s^{-3}, respectively, for our method; and 0.0265 rad s^{-3} and 0.0075 rad s^{-3}, respectively, for the baseline without the cinematographic cost term.

Fig. 7 shows the trajectories followed by the whole UAV formation throughout the experiment to film the maintenance operation. It can be seen that the formation is properly maintained to avoid collisions between the UAVs and the tower, and to provide required lighting of the filmed object. Moreover, none of the UAVs appear in the camera's field of view. The temporal evolution of the deviations from the desired orientation of each light and their distance from entering the camera FoV during this simulation are shown in Fig. 8. A video of the complete simulation can be found at the site with multimedia materials.

C. Simulation - Cluttered environment

The aim of this simulation was to demonstrate the performance of our method for trajectory planning in a cluttered environment while assessing its scalability with numerous obstacles. We simulated a forest-like scenario with multiple trees as obstacles. As a human target walks through the forest, the filming UAV executes a chase shot from behind while the lighting UAVs follow the leader side by side. Fig. 9 depicts the distribution of the obstacles around the forest and the generated trajectories for the UAVs. In this figure, it is visible that the UAVs were able to follow the human in formation and to simultaneously avoid obstacles.

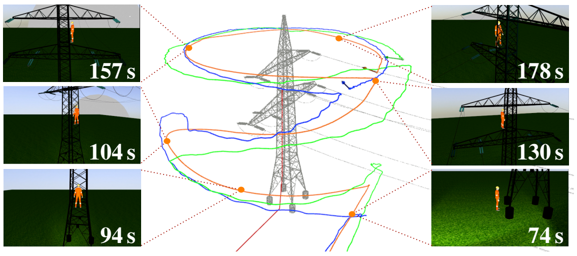


Fig. 7: An illustration of the experiment where an operator is filmed working on a transmission tower. The trajectories of the camera carrying leader (orange), both followers carrying lights (blue and green), and the human worker (red) are shown. The obstacle map is represented by a point cloud, including the power lines and tower. The worker is tracked with a lateral shot as he walks to the tower and then with a sequence of fly-over shots while he climbs up. Several onboard images taken during the experiment are also shown.

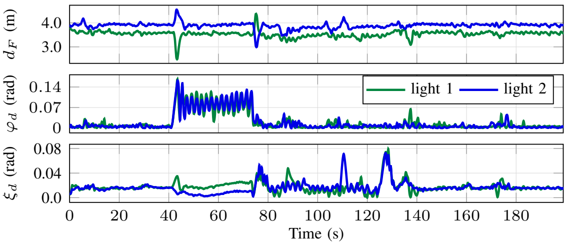


Fig. 8: Temporal evolution of the distance d_F of UAVs carrying lights from entering the camera FoV, deviation from desired heading φ_d and deviation from desired pitch angle of light ξ_d.

Finally, we analyze the scalability of our method in terms of computational demand. Simulations were run with a 4-core Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz. Table I shows the results of our method that correspond to the total planning time for each iteration that was run on the leader UAV. As expected, most time was spent during the non-convex optimization step described in Section III-B. The results for the followers are not included because they skip this non-convex optimization and thus, consume less time. The results are similar for the two simulations, although the second scenario was significantly more cluttered.

Since the map of the environment is transformed into safe corridors made of convex polyhedrons, cluttered environments do not represent an increase in the computational demands of the trajectory optimization method. Therefore, we are able to plan the leader's trajectories at a rate of 1 Hz with horizon lengths of 8 s. This rate is adequate for real-time performance in the dynamic scenarios that we target. The lower computational complexity required to generate the initial trajectories of the followers allows us to plan follower's trajectories at a higher rate of 2 Hz, enabling faster reactions to changes of the leader's behaviour and thus a more efficient mutual collision avoidance.

D. Real world experiment

In order to demonstrate our method, we performed field experiments generating trajectories for a real team of UAVs (see Fig. 10). Thus, we proved the real-time performance of

TABLE I: The planning times of our method per iteration. The total average values are shown for the two experiments. The percentage of time consumed at each step is shown thereafter. ITG stands for the procedure indicated in Section III-B, SCG for procedure described in Section III-C and FTO for trajectory optimization described in Section III-D.

	Time (s)			
	Total $(Avg \pm std)$	ITG (%)	SCG (%)	FTO (%)
Tower	0.70923 ± 0.10557	70.9982	11.81564	17.18615
Forest	0.71274 ± 0.05792	72.41338	8.77989	18.80673

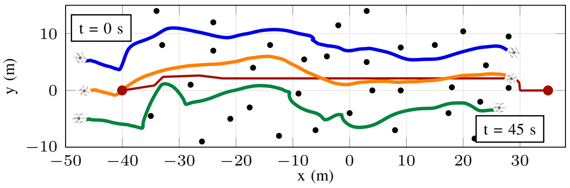


Fig. 9: A top view of the trajectories generated in the cluttered forest scenario. The trajectories of the target (red), the leader (orange), and both followers (blue and green) are shown. The black dots represent trees.

the proposed approach onboard. A sequence of shots was commanded to film a dynamic target in an outdoor scenario. A video of the experiment can be found at the multimedia material site.

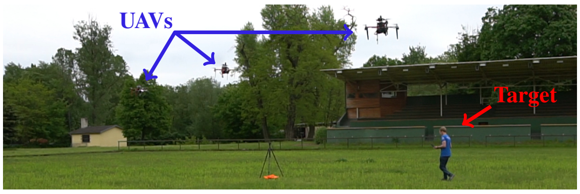


Fig. 10: A snapshot of a real experiment in an outdoor scenario. UAVs were localized using RTK GPS. The target location was generated from a noisy ground truth, without onboard visual detection.

V. CONCLUSION

This paper has presented a method for autonomous aerial cinematography with distributed lighting by a team of UAVs. We have proposed a novel methodology for multi-UAV trajectory planning, addressing non-linear cinematographic aspects and obstacle avoidance in separate optimization steps. We have demonstrated that the method is capable of generating smooth trajectories complying with aesthetic objectives for the filming UAV; and trajectories for the follower UAVs that allow them to keep a formation lighting the target properly and staying out of the camera FoV. Besides, our results indicate that we can plan trajectories in a distributed and online manner, and that the method is suitable for obstacle avoidance even in cluttered environments. As future work, we plan to address occlusions caused by obstacles within the camera FoV. Our idea is to compute the regions where these occlusions would take place and include them in the representation of the occupied space.

REFERENCES

[1] I. Mademlis *et al.*, "High-level multiple-UAV cinematography tools for covering outdoor events," *IEEE Trans. on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019.

[2] A. Sabirova *et al.*, "Drone cinematography system design and new guideline model for scene objects interaction," in *2020 Int. Conf. Nonlinearity, Information and Robotics*, 2020.

[3] B. F. Jeon *et al.*, "Detection-Aware Trajectory Generation for a Drone Cinematographer," *ArXiv e-prints*.

[4] L.-E. Caraballo *et al.*, "Autonomous planning for multiple aerial cinematographers," in *IEEE/RSJ IROS*, 2020.

[5] P. Moreno *et al.*, "Aerial multi-camera robotic jib crane," *IEEE RA-L*, vol. 6, no. 2, pp. 4103–4108, 2021.

[6] DJI, "Mavic pro 2," 2018. [Online]. Available: www.dji.com/es/mavic

[7] Skydio, "Skydio 2," 2019. [Online]. Available: www.skydio.com

[8] R. Bonatti *et al.*, "Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments," in *IEEE/RSJ IROS*, 2019.

[9] R. Bonatti *et al.*, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *JFR*, vol. 37, no. 4, pp. 606–641, 2020.

[10] T. Nägeli *et al.*, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE RA-L*, vol. 2, no. 3, pp. 1696–1703, 2017.

[11] N. Passalis *et al.*, "Deep reinforcement learning for frontal view person shooting using drones," in *IEEE EAIS*, 2018.

[12] Y. Dang *et al.*, "Imitation learning-based algorithm for drone cinematography system," *IEEE Trans. Cogn. Devel. Syst.*, pp. 1–1, 2020.

[13] C. Huang *et al.*, "Learning to capture a film-look video with a camera drone," in *IEEE ICRA*, 2019, pp. 1871–1877.

[14] M. Gschwindt *et al.*, "Can a robot become a movie director? Learning artistic principles for aerial cinematography," in *IEEE/RSJ IROS*, 2019.

[15] T. Nägeli *et al.*, "Real-time planning for automated multi-view drone cinematography," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–10, 2017.

[16] Q. Galvane *et al.*, "Directing cinematographic drones," *ACM Trans. Graph.*, vol. 37, no. 3, pp. 1–18, 2018.

[17] A. Bucker *et al.*, "Do You See What I See? Coordinating multiple aerial cameras for robot cinematography," *arXiv*, 2020.

[18] A. Alcántara *et al.*, "Autonomous execution of cinematographic shots with multiple drones," *IEEE Access*, pp. 201 300–201 316, 2020.

[19] ——, "Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles," *RAS*, vol. 140, p. 103778, 2021.

[20] B. Hall, *Understanding cinematography*. Crowood, 2015.

[21] P. Petráček *et al.*, "Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments," *IEEE RA-L*, vol. 5, no. 2, pp. 2078–2085, 2020.

[22] M. Petrlík *et al.*, "A robust UAV system for operations in a constrained environment," *IEEE RA-L*, vol. 5, no. 2, pp. 2169–2176, 2020.

[23] M. Saska *et al.*, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *IEEE ETFA*, 2017.

[24] V. Krátký *et al.*, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE RA-L*, vol. 5, no. 2, pp. 2302–2309, 2020.

[25] J. Chen *et al.*, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *IEEE ICRA*, 2016.

[26] K. Mohta *et al.*, "Fast, autonomous flight in GPS-denied and cluttered environments," *JFR*, vol. 35, no. 1, pp. 101–120, 2018.

[27] S. Liu *et al.*, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE RA-L*, vol. 2, no. 3, pp. 1688–1695, 2017.

[28] J. Tordesillas *et al.*, "FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments," *IEEE/RSJ IROS*, 2019.

[29] T. Báča *et al.*, "The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *JINT*, vol. 26, 2021.

[30] D. Harabor *et al.*, "Online graph pruning for pathfinding on grid maps." in *AAAI Conf. on Artificial Intelligence*, vol. 25, no. 1, 2011.

[31] ——, "Improving jump point search," in *Int. Conf. on Automated Planning and Scheduling*, vol. 24, no. 1, 2014.

[32] B. Houska *et al.*, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[33] C. Lino *et al.*, "Intuitive and Efficient Camera Control with the Toric Space," *ACM Trans. Graph.*, 2015.

[34] K. Shoemake, "Arcball: a user interface for specifying three-dimensional orientation using a mouse," in *Proceedings of Graphics Interface*, 1992.

[35] C. Gebhardt *et al.*, "Airways: Optimization-Based Planning of Quadrotor Trajectories according to High-Level User Goals," in *Proceedings of the Conf. on Human Factors in Computing Systems*, 2016.

Chapter 5

Collision-Aware Time-Optimal Formation Reshaping

In this chapter, we present the core publication [7c] introducing an optimal algorithm to address the time-optimal formation reshaping in three-dimensional environments with guarantees on mutual collision avoidance among robots. The proposed deterministic, centralized Collision-Aware Time-Optimal formation Reshaping Algorithm (CAT-ORA) consists of two key components: (i) an algorithm for solution of robot-to-goal assignment and (ii) trajectory generation approach producing trajectories minimizing the total duration of the formation reshaping process. The robot-to-goal assignment builds upon the Hungarian algorithm [132], [133], which has been adapted to effectively implement mutual collision avoidance through direct constraints on mutually exclusive robot-goal pairs. The approach to generate a set of collision-free reshaping trajectories is based on a closed-form solution to minimum time trajectory generation problem, while maintaining similar progress along the path for all trajectories, which is crucial for collision avoidance guarantees. The algorithm is structured such that the robot-to-goal assignment and trajectory generation can be solved in a decoupled manner without compromising the optimality of the solution.

The proposed algorithm demonstrates superior performance in terms of minimizing the makespan of the formation reshaping process compared to methods commonly applied in mobile robotics. It does so while keeping computational demands at a level suitable for real-time deployment, even for large formations consisting of up to 100 robots. The properties of CAT-ORA have been evaluated by thorough numerical and theoretical analysis, including the proof of optimality. The applicability of the algorithm in practical scenarios was validated in numerous simulations with hundreds of robots and real-world experiments involving up to 19 UAVs. Notably, the results highlight a significant advantage of the robot-to-goal assignment aspect within the proposed algorithm, which significantly reduces the maximum length of the assigned paths — by up to 48% compared to methods used in contemporary cooperative motion planning and formation control. This reduction is especially beneficial for aerial vehicles with constrained operational time, enhancing their efficiency in real-world scenarios.

The author's contribution to this work is 70% and lies in designing and implementing the algorithm, conducting theoretical and numerical analysis, and preparing the manuscript. The coauthors assisted with the experiments and provided feedback on the proposed method and the manuscript.

[7c] **V. Krátký**, R. Pěnička, J. Horyna, P. Štibinger, T. Báča, M. Petrlík, P. Štěpán, and M. Saska, "CAT-ORA: Collision-aware time-optimal formation reshaping for efficient robot coordination in 3D environments," *submitted to Transactions on Robotics*, 2023

CAT-ORA: Collision-Aware Time-Optimal Formation Reshaping for Efficient Robot Coordination in 3D Environments

Vit Kratky* , Robert Penicka, Jiri Horyna, Petr Stibinger, Tomas Baca, Matej Petrlik, Petr Stepan, Martin Saska

***Abstract*—In this paper, we introduce an algorithm designed to address the problem of time-optimal formation reshaping in three-dimensional environments while preventing collisions between agents. The utility of the proposed approach is particularly evident in the realm of mobile robotics, where the agents benefit from being organized and navigated in formation for a variety of real-world applications requiring frequent alterations in formation shape for efficient navigation or task completion. Given the constrained battery life inherent to mobile robots, the time needed to complete the formation reshaping process is crucial for efficient operation, especially for Unmanned Aerial Vehicles (UAVs). The proposed Collision-Aware Time-Optimal formation Reshaping Algorithm (CAT-ORA) builds upon the Hungarian algorithm for the solution of the robot-to-goal assignment implementing the inter-agent collision avoidance through direct constraints on mutually exclusive robot-goal pairs combined with a trajectory generation approach minimizing the duration of the reshaping process. Theoretical validations confirm the optimality of the proposed CAT-ORA, with its efficacy further showcased through simulations, and a real-world outdoor experiment involving 19 UAVs. Thorough numerical analysis shows the potential of CAT-ORA to decrease the time required to perform complex formation reshaping tasks by up to 48% compared to commonly used methods in randomly generated scenarios.**

***Index Terms*—Multi-Robot Systems, Path Planning for Multiple Mobile Robots or Agents, Collision Avoidance, Formation Reshaping**

Supplementary material

Video: https://mrs.felk.cvut.cz/tro2023catora

I. Introduction

Teams of autonomous mobile robots have found practical applications in various real-world scenarios, including search and rescue operations [1], [2], environmental monitoring [3], and automated warehouse systems [4]. In most cases, these teams consist of robots working together to achieve a common objective while independently navigating through the environment and avoiding collisions. However, in certain applications, it is advantageous for mobile robots to be arranged in a specific formation to accomplish desired tasks, such as documenting historical buildings [5], monitoring wildfires [6], or creating drone light shows [7], [8]. In these scenarios, the robots often need to adjust their positions relative to one another to achieve the required formation shape for the mission's execution. Considering the formation shape adaptation as part of a robotic mission, the time efficiency of this process becomes of great importance. This applies specifically to vehicles with operational time significantly constrained by battery endurance, such as multi-rotor Unmanned Aerial Vehicles (UAVs), especially in time critical missions such as search and rescue and applications requiring highly dynamic performance, such as drone light shows.

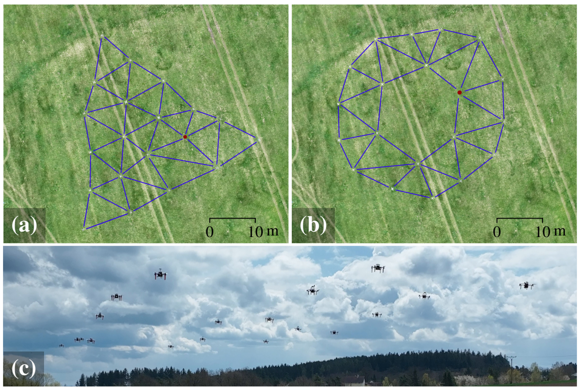


Fig. 1: Deployment of the introduced Collision-Aware Time-Optimal formation Reshaping Algorithm (CAT-ORA) in a small-scale drone visual performance with 19 UAVs. The images show the transition of UAVs guided by the CAT-ORA from a triangular shape (a) to a ring shape (b). This transition was performed within 7 seconds. The blue lines highlight the shape of the formation in top view images, while (c) captures the flying formation from the side.

This paper tackles the Time-Optimal Formation Reshaping Problem (TOFREP) with collision avoidance guarantees. The problem involves finding the assignment of robots to goals coupled with the generation of minimum-time collision-free trajectories. From the robotics perspective, the formation reshaping problem is a specific instance of cooperative motion planning. However, instead of having specific goals assigned to individual robots, the group of robots is given a set of unassigned goals to visit. The algorithms for the solution of assignment problems have been widely tackled in literature [9]–[20]. However, since robots are physical entities sharing an environment, mutual collision avoidance has to be considered

Authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague 6, Czech Republic.

*Corresponding author, {vit.kratky|penicrob|horynjir|stibipet |bacatoma|matej.petrlik|stepan|martin.saska}@fel.cvut.cz

This work was partially funded by the CTU grant no. SGS23/177/OHK3/3T/13, by the Czech Science Foundation (GAČR) grant no. 23-06162M, and by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590).

during the assignment process. This consideration implies that, in general, the individual cost of assigning two robot-goal pairs in a matching depends on the other assigned pairs, preventing a direct use of algorithms for the solution of general assignment problems.

Previous works in the field of formation reshaping vary in the level of decentralization, complexity, dimensions of the environment, optimization criteria, and applied methodology [21]–[28]. Although the completion time of the reshaping process is a critical factor for algorithms deployed on robots with limited operational time, only a few works have taken the time criterion into account [24]–[26]. However, none has addressed the minimization of completion time while simultaneously accounting for mutual collision avoidance among robots and the dynamic constraints associated with robots as physical entities. Furthermore, these works lack guarantees regarding the solution completeness and quality, which is one aspect that limits the transfer of the algorithms to industrial applications where we observe a clear tendency to favour robotic systems with predictable and well-defined behavior guided by clear, understandable rules.

To this end, we address the problem of the time-efficient collision-free formation reshaping by introducing a centralized, deterministic Collision-Aware Time-Optimal formation Reshaping Algorithm (CAT-ORA) with guarantees on a minimum mutual distance of involved agents during the reshaping process (mutual collision avoidance). Our proposed approach comprises two key components that must be designed together to allow us to provide theoretical guarantees of the overall complex robotic system's behavior: (i) an algorithm for optimal robot-to-goal assignment considering mutual collision avoidance among robots, and (ii) a computationally efficient trajectory generation approach minimizing the completion time of a set of trajectories. CAT-ORA builds upon the Hungarian algorithm [9], [10] adapted to solve the robot-to-goal assignment as Linear Bottleneck Assignment Problem (LBAP) [29] effectively managing potential collisions between assigned robot-goal pairs. The designed approach to the generation of a set of trajectories is based on a closed-form solution to the minimum-time single-trajectory generation problem [30] adapted to generate a set of trajectories minimizing their makespan, while keeping collision-free properties.

The optimality, efficacy, and other attributes of CAT-ORA have been confirmed through theoretical validation, statistical evaluation, and a real-world demonstration of a formation flight in a small-scale visual entertainment performance involving up to 19 UAVs (see Fig. 1). The results show the capability of CAT-ORA to decrease the time required to perform formation reshaping tasks by up to 48% compared to Linear Sum Assignment Problem (LSAP)-based solution [9], [10] while providing collision avoidance guarantees. The CAT-ORA finds a solution within a few milliseconds for instances of up to 32 robots, making it highly suitable for real-time applications that require on-the-fly computations.

Building on the capabilities of the CAT-ORA, this work contributes by directly optimizing the completion time (the so-called makespan) of the formation reshaping process in a 3D environment, while considering inter-agent collision avoidance. We show that the robot-to-goal assignment problem can be formulated as LBAP with additional constraints ensuring inter-agent collision avoidance. The proposed algorithm demonstrates that this problem can be solved in real time while significantly reducing the makespan of the formation reshaping process. This outcome holds significant value, especially for robots with limited operational time, and can be employed to enhance existing approaches or serve as a foundation for future research in formation reshaping, particularly concerning the autonomous deployment of cooperating multi-robot systems in real-world environments.

We consider the particular contributions of this paper to be:

- We present a lower bound on a minimum mutual distance of trajectories for the solution of the robot-to-goal assignment as LBAP along with its theoretical proof.
- We introduce a deterministic, complete algorithm for solving the robot-to-goal assignment problem, minimizing the maximum length of the path among assigned robot-goal pairs while respecting constraints on mutually exclusive robot-goal pairs.
- We provide a closed-form solution for generating a set of trajectories connecting given start and goal positions and minimizing the makespan while preserving the guarantees of collision-free properties.
- We combine the contributions mentioned above to build CAT-ORA, the first known complete approach for the solution of the formation reshaping problem minimizing the makespan while providing collision avoidance guarantees in 3D environments. We provide verification of its properties through several proofs, numerical analysis, and a real-world experiment.
- We provide a quantitative and theoretical analysis of the CAT-ORA solution compared to the LSAP-based solution in terms of the makespan of a reshaping process, showing its superior performance.

II. Related Work

The introduced time-optimal formation reshaping problem can be considered a robot-to-goal assignment problem closely coupled with a minimum-time trajectory generation. One of the most frequently used methods for solving assignment problems is the Hungarian algorithm [9], [10], capable of providing a solution minimizing the sum of individual associations costs with time complexity $O(n^3)$, where n is the number of robots. Since the Hungarian algorithm was introduced, many algorithms tackling the assignment problem and its variants with different characteristics and performance have been developed [11]–[15]. Notably, there have been innovative approaches grounded in game-theory principles [18], swarm-intelligence [19], reinforcement learning [31], and market-based methodologies [20]. Despite these advancements, the Hungarian algorithm remains in use and has been utilized in many works as an efficient centralized algorithm with provable properties to solve the assignment problem.

First introduced in [32], a specific case of the assignment problem minimizing the maximum individual cost (bottleneck) is called Linear Bottleneck Assignment Problem (LBAP). Similarly to the original problem, LBAP can be solved using the Hungarian algorithm by modified cost substitutions [32], applying threshold algorithms [16] or shortest augmenting path algorithms [17]. An important adaptation of the Hungarian algorithm, used later in this work, is its dynamic variant [33]. Given the initial assignment, the dynamic variant enables solving an assignment problem with changed costs in approximately one hundredth of the computational requirements of the Hungarian algorithm starting from scratch. Despite the variety of approaches developed to solve assignment problems, their direct use to formation reshaping is restricted by neglecting the collision resolution among robots.

Methods suitable for addressing the formation reshaping problem exhibit differences in terms of decentralization, guarantees on the quality of the solution, and employed methods. The centralized approaches [21]–[26] mostly take advantage of the complete information for generating optimal solutions, but often impose an assumption on the collision-free environment. In [21], the authors propose a centralized approach based on the concurrent solution of assignment of goals and planning of trajectories (CAPT algorithm), which is further extended to a decentralized approach. The CAPT incorporates a solution of LSAP minimizing the sum of squared traveled distances combined with constant-velocity and minimum-snap trajectories, which was proved to yield collision-free trajectories under the assumption on a minimum initial distance between agents. The problem is extended to a variable goal formation (variable scale and translation) in [22], where the authors show that the problem of task assignment with variable goal formation can be transformed to LSAP via cost substitution. However, the approach is limited to 2D, and the mutual collisions are prevented by adapting the scale of the final formation.

Unlike the centralized solutions, the distributed approaches often suffer from incomplete information, leading to suboptimal solutions (e.g., approach applying local task swapping [27]) and limited guarantees on its quality (e.g., learning-based approach [28]). The robot-to-goal assignment problem is also solved in several works on distributed control of multi-rotor UAV formation in obstacle-free regions [34], as well as in complex environments [35], [36]. These works apply distributed task assignment algorithms [20], [37] to assign the robots to local goals during alignment to the target formation. Although [34]–[36] are proposed primarily for multi-rotor helicopters, they utilize the sum of squared distances as the minimization criterion for the assignment problem. Such choice provides certain guarantees on the mutual distance of trajectories if solved optimally [21]. However, it does not reflect the problem being solved since minimizing squared traveled distances for in-flight multi-rotor UAVs is neither optimal from the point of view of duration, energy consumption, nor any other appropriate criterion.

The time criterion was considered only in a few works dealing with formation reshaping. In [24], the authors aim to minimize the total time in motion and build the solution of an assignment problem on duration of time-optimal trajectories. The algorithm relies on collision resolution via a combination of time delays and altitude adaptation, which limits its application to 3D environments. The algorithm presented in [25] directly approaches the minimization of the makespan by defining the problem as LBAP. The proposed solution considers a variable goal formation but is limited to 2D and ignores inter-agent collisions. Another algorithm considering the minimization of the makespan [26] also solves the assignment as a variant of LBAP, but it considers constant-velocity trajectories only. The authors provide proof of collision avoidance guarantees; however, these are only valid in 2D environments with initial and goal configurations constrained to the grid. Although some related works show impressive results, the oversight regarding the mutual collisions or minimum-time objective of the robot-to-goal assignment limits their efficient use in real-world applications.

III. Problem Definition

The Time-Optimal Formation Reshaping Problem (TOFREP), tackled in this manuscript, is defined as follows. Given the set of initial configurations of n unlabeled robots $\mathbb{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n\}$ and set of n goal configurations $\mathbb{G} = \{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n\}$, find a set of collision-free trajectories $\mathbb{T}$ that guide the robots from $\mathbb{S}$ to $\mathbb{G}$ while minimizing the makespan of the reshaping process.

Let us define the makespan of reshaping the formation F given the assignment $\phi : \mathbb{S} \to \mathbb{G}$ as

$$M_r(\mathbb{S}, \mathbb{G}, \phi) = \max_{(i,j)\in\phi} \mathrm{tf}(\mathrm{T}(\mathbf{s}_i, \mathbf{g}_j)), \tag{1}$$

where $\mathrm{tf}(\mathrm{T}(\mathbf{a}, \mathbf{b}))$ represents the time required to reach position $\mathbf{b}$ from position $\mathbf{a}$ following trajectory $\mathrm{T}(\mathbf{a}, \mathbf{b})$. Then, the TOFREP is defined as

$$\begin{aligned} &\underset{\phi\in\Phi, \mathrm{T}\in\mathscr{T}}{\text{minimize}}\; M_r(\mathbb{S}, \mathbb{G}, \phi), \\ &\text{subject to}\; \mathrm{dist}_{\min}(\mathrm{T}(\mathbf{s}_i, \mathbf{g}_j), \mathrm{T}(\mathbf{s}_k, \mathbf{g}_l)) \geq \Delta, \\ &\qquad\qquad \forall (i,j) \in \phi, (k,l) \in \phi, (i,j) \neq (k,l), \end{aligned} \tag{2}$$

where $\mathrm{dist}_{min}(\cdot)$ represents the minimum distance between trajectories given as arguments, Δ stands for the minimum acceptable mutual distance of robots, Φ is the set of all possible assignments from $\mathbb{S}$ to $\mathbb{G}$, and $\mathscr{T}$ is a class of arbitrary trajectory generation functions.

In the following sections, we introduce CAT-ORA, an optimal algorithm for the solution of problem (2) under the following assumptions:

(A1) Both the robots and the goals are unlabeled (any robot can be assigned to an arbitrary goal location).

(A2) The robots are stationary in the initial and goal configurations.

(A3) The motion of the robots between the initial and goal configuration is limited to straight paths with mutually equivalent time parametrization.

(A4) The robots are considered to be spheres with radius R for the collision avoidance resolution.

(A5) The minimum distance between the pairs of initial configurations and the pairs of goal configurations $\delta = \min_{i,j,i\neq j} \min(||\mathbf{s}_i - \mathbf{s}_j||, ||\mathbf{g}_i - \mathbf{g}_j||)$ fulfills the condition $\delta \geq \eta\Delta$ with $\eta \geq \sqrt{2}$ being a constant parameter.

(A6) The convex hull of $\mathbb{S} \cup \mathbb{G}$ is free of obstacles apart from the robots themselves.

The assumptions (A1) – (A6) are necessary to guarantee the optimality of the proposed algorithm to the solution of TOFREP, as defined in (2). However, in Section X, we show that the assumption (A2) is not strict and that the algorithm can also be used for reshaping moving formations, and further that the optimal solution considering assumption (A3) stays close to the theoretical lower bound of the optimal solution not considering (A3). The assumptions (A5) and especially (A6) impose significant limitations, but both (A5) and (A6) may be easily satisfied in most of the real-world scenarios discussed in Section I, making the proposed solution practical for real-world applications.

IV. Overview of the maximum matching in bipartite graphs and the Hungarian method

In this section, we overview key terms and definitions from graph theory applied in a further description of the proposed methodology and briefly describe the Hungarian method employed in the proposed algorithm.

A. Maximum matching in bipartite graphs

Key terms related to maximum matching problem in bipartite graphs used in following sections are:

- *Bipartite graph*: graph $G = \{V, E\} = \{V_x, V_y, E\}$, where the set of vertices V can be partitioned in two disjoint subsets V_x, V_y, such that the set of edges E does not contain any edge connecting vertices from the same partition.
- *Matching*: subset of edges $E_M \subset E$, such that every vertex in V is incident to at most one edge in E_M.
- *Cardinality of the matching*: number of edges in a matching $C_M = |E_M|$. The matching containing the maximum possible number of edges is called *maximum cardinality matching*. If $C_M = |V_x| = |V_y|$, the matching is called *perfect*.
- *Matched edge*: edge e_{ij} is called matched if it is a part of the matching, unmatched otherwise.
- *Matched vertex*: vertex v is matched if it is incident to an edge in matching E_M, and unmatched otherwise.
- *Alternating path*: path in a graph that starts with an unmatched vertex and alternates between edges that do not and do belong to the matching.
- *Augmenting path*: an alternating path that ends with an unmatched vertex.
- *Minimum weight bipartite matching problem*: given bipartite graph $G = \{V_x, V_y, E\}$ and weight function $w : E \rightarrow \mathbb{R}$, find a maximum cardinality matching E_M, such that $\sum_{e_{ij} \in E_M} w(e_{ij})$ is minimum.
- *Dual problem of minimum weight bipartite matching problem*: given bipartite graph $G = \{V, E\} = \{V_x, V_y, E\}$, weight function $w : E \rightarrow \mathbb{R}$, and vertex labeling function $l_f : V \rightarrow \mathbb{R}$, find a feasible labeling of a maximum cost $c(l_f) = \sum_{v_{x,i} \in V_x} l_f(v_{x,i}) + \sum_{v_{y,j} \in V_y} l_f(v_{y,j})$, where *feasible labeling* is a choice of labels such that $l_f(v_{x,i}) + l_f(v_{y,j}) \leq w(e_{ij})$.

For simplicity of description and without loss of generality, we assume that the bipartite graph $G = \{V_x, V_y, E\}$ is complete and balanced, i.e., $|V_x| = |V_y|$ in the remainder of the paper.

B. Hungarian algorithm

The Hungarian algorithm [9], [10] is widely applied for the solution of the assignment problem (which can also be represented as a minimum weight bipartite matching problem) with proven complexity $O(n^3)$, where n is a number of matched entities. The input of the Hungarian algorithm is a square biadjacency matrix $\mathbf{M}_d$ representing a weighted bipartite graph G with weight function $w : E \rightarrow \mathbb{R}$. The algorithm exploits the properties of the dual of minimum weight bipartite matching problem by using dual variables $u_i = l_f(v_{x,i})$, $v_j = l_f(v_{y,j})$, $i, j \in \{0, 1, \cdots, N\}$. These variables are updated during the run of the algorithm and used to determine the admissibility of edge e_{ij} given by condition

$$u_i + v_j = w(e_{ij}). \tag{3}$$

The Hungarian algorithm starts with an empty matching ϕ and repeatedly searches for augmenting paths in an equality subgraph formed by edges fulfilling condition (3). The search for an augmenting path is realized by building so-called Hungarian trees that are rooted in unmatched nodes. If the Hungarian tree formed by alternating paths in a graph G contains an augmenting path, the current matching is updated by flipping the matched and unmatched edges along the found path. This process always increases the cardinality of current matching by one in a single step of the algorithm. If the augmenting path is not found in a current equality subgraph, the values of dual variables are updated such that the dual task remains feasible and new edges are introduced into the equality subgraph. Then, the search for an augmenting path continues. The incremental increase of the cardinality of the matching ensures that the algorithm reaches a perfect matching for $\mathbf{M}_d \in \mathbb{R}^{n \times n}$ in n steps of a successful search for an augmenting path. We refer to [9], [10], [29] for a detailed description of the algorithm and proofs of its properties.

V. Collision-Aware Time-Optimal formation Reshaping Algorithm - overview

The introduced TOFREP (2) consists of two problems: (i) the optimal assignment of initial configurations to goal configurations and (ii) the generation of collision-free minimum-time trajectories. In further description, we assume that these two problems are completely separable, and that

$$\begin{aligned} \max_{(i,j) \in \phi_a} \operatorname{tf}(\mathrm{T}(\mathbf{s}_i, \mathbf{g}_j)) \geq \max_{(i,j) \in \phi_b} \operatorname{tf}(\mathrm{T}(\mathbf{s}_i, \mathbf{g}_j)) \implies \\ \max_{(i,j) \in \phi_a} ||\mathbf{s}_i - \mathbf{g}_j|| \geq \max_{(i,j) \in \phi_b} ||\mathbf{s}_i - \mathbf{g}_j|| \end{aligned} \tag{4}$$

holds for all assignments ϕ_a, ϕ_b from $\mathbb{S}$ to $\mathbb{G}$. This means that the assignment minimizing the makespan (1) corresponds to the assignment minimizing the maximum distance d_{max} between the assigned initial and goal configurations

$$d_{max} = \max_{(i,j) \in \phi} ||\mathbf{s}_i - \mathbf{g}_j||. \tag{5}$$

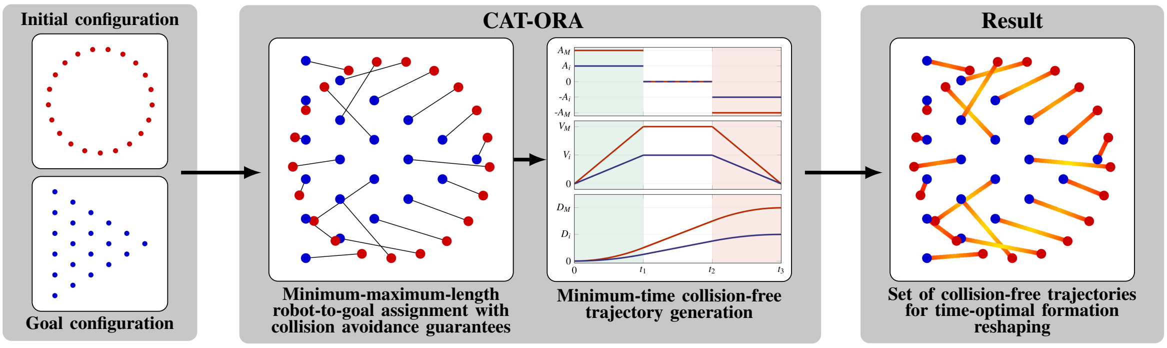

Initial configuration
Goal configuration
CAT-ORA
A_M
A_i
0
$-A_i$
$-A_M$
V_M
V_i
0
D_M
D_i
0
0
t_1
t_2
t_3
Minimum-maximum-length robot-to-goal assignment with collision avoidance guarantees
Minimum-time collision-free trajectory generation
Result
Set of collision-free trajectories for time-optimal formation reshaping

Fig. 2: Block diagram of the proposed Collision-Aware Time-Optimal formation Reshaping Algorithm (CAT-ORA). The colors of the trajectories in the image on the right encode the velocity profile of particular trajectories, with red being equal to zero velocity and yellow to maximum velocity.

This allows us to design Collision-Aware Time-Optimal Formation Reshaping Algorithm (CAT-ORA) such that the robot-to-goal assignment and generation of collision-free minimum-time trajectories are tackled in a decoupled way (see Fig. 2 for block diagram of CAT-ORA). The proof that the proposed decoupled approach does not influence the optimal solution and that (4) is fulfilled within the proposed approach is provided in Section IX-A.

A. *Minimum-weight robot-to-goal assignment*

The task of assigning the goal configurations to particular robots can be defined as an integer linear program

$$\begin{aligned} \text{minimize} & \sum_{i=1}^{n}\sum_{j=1}^{n} w(e_{ij})x_{ij}, \\ \text{subject to} & \sum_{i=1}^{n} x_{ij} = 1 \; \forall j \in \{1,\dots,n\}, \\ & \sum_{j=1}^{n} x_{ij} = 1 \; \forall i \in \{1,\dots,n\}, \\ & x_{ij} \in \{0,1\} \; \forall i,j \in \{1,\dots,n\}, \end{aligned} \tag{6}$$

where $w(e_{ij})$ is the cost of assignment of the goal configuration $\mathbf{g}_j$ to initial configuration $\mathbf{s}_i$, and $x_{ij} = 1$ if $\mathbf{s}_i$ is assigned to $\mathbf{g}_j$, $x_{ij} = 0$ otherwise. The problem (6) is often referred to as Linear Sum Assignment Problem (LSAP) which can be efficiently solved by the Hungarian algorithm [9], [10]. Using the squared Euclidean distances $||\mathbf{s}_i - \mathbf{g}_j||^2$ as costs $w(e_{ij})$, the solution of (6) was proved to guarantee the collision-free property of constant-velocity trajectories when $\delta \geq \sqrt{2}R$ [21], where R is the safety radius of robots.

In compliance with (4), problem (6) must be reformulated to minimize the length of the longest trajectory in the assignment for solving TOFREP (2):

$$\begin{aligned} \text{minimize} & \max_{i,j \in \{0,\dots,n\}} w(e_{ij})x_{ij}, \\ \text{subject to} & \sum_{i=1}^{n} x_{ij} = 1 \; \forall j \in \{1,\dots,n\}, \\ & \sum_{j=1}^{n} x_{ij} = 1 \; \forall i \in \{1,\dots,n\}, \\ & x_{ij} \in \{0,1\} \; \forall i,j \in \{1,\dots,n\}, \end{aligned} \tag{7}$$

known as Linear Bottleneck Assignment Problem (LBAP) [32]. The specificity of the robot-to-goal assignment problem requires augmenting (7) by including constraints on mutually colliding paths

$$\sum_{e=1}^{|C_m|} x_{\text{idx}(C_{m,e})} = 1 \; \forall C_m \in \mathbf{C}, \tag{8}$$

where $\mathbf{C}$ is a set of constraints represented by sets of mutually colliding edges, and $\text{idx}(\cdot)$ represents the indices of the corresponding edge.

Solving (7) augmented by (8) using standard optimization methods would require to compute the whole set of mutual collision constraints prior to the solution of the problem, which would require to check collisions among $\frac{n^2(n-1)^2}{2}$ pairs of edges, making it computationally intractable for large n. In this work, we propose a novel algorithm that combines the Hungarian algorithm [9], [10] and its dynamic variant [33] with fast collision checking. The collision checking is built on the analysis of theoretical guarantees on a minimum mutual distance of trajectories for an assignment provided as a solution of LBAP (detailed in Section VI). A thorough description of the robot-to-goal assignment component of CAT-ORA is provided in Section VII.

B. *Minimum makespan collision-aware trajectory planning*

The generation of collision-free trajectories between pairs of matched initial and goal configurations that minimize the makespan of the formation reshaping process requires considering the generation of individual minimum-time trajectories. In compliance with assumption (A3), we consider a model with single-dimension point-mass dynamics $\ddot{p} = a$, with constraints on acceleration control inputs $-a_{max} \leq a \leq a_{max}$, and limits on velocity $v = \dot{p}$, $0 \leq v \leq v_{max}$. Although the individual minimum-time trajectories using this model would minimize the makespan, they do not preserve the guarantees on mutual collision avoidance. Exploiting the fact that the minimized makespan is influenced only by the length of the longest trajectory, we have proposed an approach for generating mutually collision-free minimum-time trajectories, preserving the theoretical guarantees on minimum mutual distance. The proposed approach, which is based on a closed-form solution of the minimum-time trajectory generation problem, is detailed in Section VIII, along with the proof of theoretical guarantees.

VI. Theoretical guarantees of LBAP solution

The solution of LSAP using squared Euclidean distances as costs has been proved to guarantee minimum distance between trajectories d_{min} equal to

$$d_{min} = \frac{\sqrt{2}}{2}\delta, \tag{9}$$

where δ stands for the minimum distance between any pair of initial and goal configurations [21]. In the following sections, similar properties are derived and proved for the application of LBAP to solve the same problem while minimizing the maximum distance between the assigned initial and goal configurations (5), thus minimizing the makespan (4).

A. Minimum mutual distance of two trajectories

For the analysis of the guarantees on the minimum distance between trajectories, we consider the following scenario. Without loss of generality, we can assume fixed initial and goal positions $\mathbf{s}_i$, $\mathbf{s}_j$, $\mathbf{g}_i$ and an arbitrarily positioned goal position $\mathbf{g}_j$ with $||\mathbf{s}_j - \mathbf{g}_i|| = d$, $||\mathbf{s}_j - \mathbf{g}_i|| \geq ||\mathbf{s}_i - \mathbf{g}_j||$, and $||\mathbf{s}_i - \mathbf{g}_i|| = Md, M \in [0,1)$ (see Fig. 3). For $M \geq 1$, the LBAP solution coincides with the solution to LSAP, thus implicitly providing the same guarantees on minimum mutual distance.

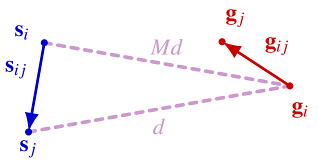


Fig. 3: An example problem consisting of two initial positions $\mathbf{s}_i$, $\mathbf{s}_j$ and two goal locations $\mathbf{g}_i$, $\mathbf{g}_j$. Without loss of generality, the distance $|\mathbf{s}_j - \mathbf{g}_i|$ is assumed to be equal to d and $|\mathbf{s}_i - \mathbf{g}_i| = Md$, where $M \in [0,1)$.

Considering the trajectories with constant velocity, the position of robot $\mathbf{x}_i(t)$ at time t following the trajectory from initial position $\mathbf{s}_i$ to goal position $\mathbf{g}_i$ can be described as

$$\mathbf{x}_i(t) = (1-\alpha)\mathbf{s}_i + \alpha\mathbf{g}_i, \tag{10}$$

with $\alpha = \frac{t}{t_d}$ uniformly sampled on $[0,1]$, where t_d is the duration of the trajectory. Hence, the mutual distance between robots following trajectories T_i and T_j of the same duration t_d from $\mathbf{s}_i$ to $\mathbf{g}_i$ with velocity v_i and $\mathbf{s}_j$ to $\mathbf{g}_j$ with velocity v_j, respectively, can be expressed as

$$\begin{aligned}||\mathbf{x}_j(t) - \mathbf{x}_i(t)|| &= ||(1-\alpha)\mathbf{s}_j + \alpha\mathbf{g}_j - (1-\alpha)\mathbf{s}_i - \alpha\mathbf{g}_i|| \\ &= ||(1-\alpha)(\mathbf{s}_j - \mathbf{s}_i) + \alpha(\mathbf{g}_j - \mathbf{g}_i)||.\end{aligned} \tag{11}$$

Using the notation introduced at the beginning of this section and notations

$$\mathbf{s}_{ij} = \mathbf{s}_j - \mathbf{s}_i, \tag{12}$$

$$\mathbf{g}_{ij} = \mathbf{g}_j - \mathbf{g}_i, \tag{13}$$

the equation (11) can be written in the following form

$$||\mathbf{x}_j(t) - \mathbf{x}_i(t)|| = ||(1-\alpha)\mathbf{s}_{ij} + \alpha\mathbf{g}_{ij}||. \tag{14}$$

Then, the squared distance is given by

$$\begin{aligned}||\mathbf{x}_j(t) - \mathbf{x}_i(t)||^2 &= ||(1-\alpha)\mathbf{s}_{ij} + \alpha\mathbf{g}_{ij}||^2 = ||\mathbf{s}_{ij} + \alpha(\mathbf{g}_{ij} - \mathbf{s}_{ij})||^2 \\ &= (\mathbf{s}_{ij} + \alpha(\mathbf{g}_{ij} - \mathbf{s}_{ij}))^T(\mathbf{s}_{ij} + \alpha(\mathbf{g}_{ij} - \mathbf{s}_{ij})).\end{aligned} \tag{15}$$

Remark: *In the remainder of this manuscript, the monotonicity of the quadratic function in the positive domain is exploited as minimization of maximum distance is equivalent to minimization of the distance squared.*

Following the theorem in [21], for notational convenience, we define:

$$\begin{aligned}a &\equiv \mathbf{s}_{ij}^T\mathbf{s}_{ij}, \\ b &\equiv \mathbf{s}_{ij}^T\mathbf{g}_{ij}, \\ c &\equiv \mathbf{g}_{ij}^T\mathbf{g}_{ij}.\end{aligned} \tag{16}$$

This enables us to simplify (15) to

$$||\mathbf{x}_j(t) - \mathbf{x}_i(t)||^2 = \alpha^2(a - 2b + c) - 2\alpha(a-b) + a. \tag{17}$$

From (17), the value of α minimizing the distance between trajectories of robots with indices i, j, can be found as

$$\alpha_{ij}^* = \frac{a-b}{a-2b+c}. \tag{18}$$

By substituting the value of α_{ij}^* in (17), the minimum squared distance between trajectories T_i, T_j is given by

$$||\mathbf{x}_i - \mathbf{x}_j||_{min}^2 = \begin{cases} \frac{ac-b^2}{a-2b+c} & \text{if } 0 < \alpha_{ij}^* < 1, \\ \delta_{ij}, & \text{otherwise,} \end{cases} \tag{19}$$

where

$$\delta_{ij} = \min(||\mathbf{g}_{ij}||, ||\mathbf{s}_{ij}||)^2. \tag{20}$$

The minimum squared distance (19) was already proved to be greater than $\frac{1}{2}\delta_{ij}$ for $b \geq 0$, which is guaranteed for solutions provided by LSAP [21]. In contrast to the application of LSAP, the LBAP solution does not directly provide any guarantee on values a, b, c, (defined in (16)), and thus the worst-case minimum distance between trajectories is zero. Given the guarantees $||\mathbf{x}_i - \mathbf{x}_j||_{min}^2 \geq \frac{1}{2}\delta_{ij}$ for $b \geq 0$ [21], we further focus on analyzing the guarantees of specific case of the LBAP solutions with $b < 0$.

Without loss of generality, we assume $c = ka, k \geq 1$, and the size of vector $\mathbf{s}_{ij} = const.$, leading to a also being constant. Considering $b = \sqrt{a}\sqrt{c}\cos\lambda$, where λ is an angle between vectors $\mathbf{s}_{ij}$ and $\mathbf{g}_{ij}$, and the constraints $a \geq 0$, $a \geq b$, $c \geq b$ enforced by constraints on $\alpha \in [0,1]$ and $b < 0$, we can rewrite the first part of (19) to

$$||\mathbf{x}_i - \mathbf{x}_j||_{min}^2 = a\frac{k(1-\cos^2(\lambda))}{k - 2\sqrt{k}\cos(\lambda) + 1}. \tag{21}$$

Since $b < 0 \implies \cos\lambda < 0$ and $a = const.$, the gradient of (21) with respect to k is non-negative for all admissible values of k. Therefore, the squared distance is minimal if $a = c$. Using the substitution (16), this result can be also interpreted as $||\mathbf{s}_{ij}|| = ||\mathbf{g}_{ij}||$. For $a = c$, the equation (19) is simplified to

$$||\mathbf{x}_i - \mathbf{x}_j||^2_{min} = \begin{cases} \frac{a+b}{2} & \text{if } 0 < \alpha^*_{ij} < 1, \\ \min(||\mathbf{s}_{ij}||, ||\mathbf{g}_{ij}||), & \text{otherwise.} \end{cases} \quad (22)$$

Since $a = \text{const.}$, $a > 0$, the minimum distance is achieved for minimum b, such that $a = c$.

Remark: *The distance between two trajectories following line segments is minimal when the trajectories intersect, which means that they lie in the same plane. For each pair of line segments* $(\mathbf{q}, \mathbf{r})$ *in three-dimensional space, it holds that either* $\mathbf{q}||\mathbf{r}$*, and thus* $\mathbf{q}$ *and* $\mathbf{r}$ *lie in the same plane, or we can find a plane* P *such that* $\mathbf{q} \in P$ *and* $\mathbf{r}||P$*. The projection* $\mathbf{r}'$ *of* $\mathbf{r}$ *into a parallel plane preserves the dimension of* $\mathbf{r}$ *and*

$$\begin{aligned} &||(1-\kappa)(\mathbf{q}(t_0) - \mathbf{r}(t_0)) + \kappa(\mathbf{q}(t_f) - \mathbf{r}(t_f))|| \geq \\ &||(1-\kappa)(\mathbf{q}(t_0) - \mathbf{r}'(t_0)) + \kappa(\mathbf{q}(t_f) - \mathbf{r}'(t_f))|| \; \forall \kappa \in [0,1], \end{aligned} \quad (23)$$

where $\mathbf{p}(t_0), \mathbf{r}(t_0)$ *and* $\mathbf{p}(t_f), \mathbf{r}(t_f)$ *stand for the start and end points of the line segments, respectively, and* κ *is an independent variable. This allows us to solve the rest of the problem in two-dimensional space without the loss of generality.*

Based on the definition (16), the value of b is given by

$$b = \mathbf{s}_{ij}^T \mathbf{g}_{ij} = ||\mathbf{s}_{ij}|| ||\mathbf{g}_{ij}|| \cos(\beta + \gamma), \quad (24)$$

where $\beta = \angle \mathbf{g}_i \mathbf{s}_j \mathbf{s}_i$ and $\gamma = \angle \mathbf{g}_j \mathbf{g}_i \mathbf{s}_j$ (see Fig. 4). Considering (24) and the limitations on the values of β and γ coming from $M \in [0, 1)$, the distance is minimized for $\beta + \gamma = \pm\pi$, resulting in the intersection of $\mathbf{s}_i - \mathbf{g}_i$ and $\mathbf{s}_j - \mathbf{g}_j$, leading to a minimum mutual distance equal to zero. Since there are no theoretical guarantees on the limits for the minimum mutual distance of robots following trajectories $\mathrm{T}_i, \mathrm{T}_j$ for a general case, we further analyze the guarantees on minimum mutual distance depending on the value of M. Given the condition

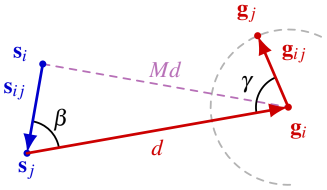


Fig. 4: Illustration of the general case of an assignment problem with fixed points $\mathbf{s}_i, \mathbf{s}_j, \mathbf{g}_i$ and variable point $\mathbf{g}_j$.

$$\frac{\max(||\mathbf{s}_i - \mathbf{g}_i||, ||\mathbf{s}_j - \mathbf{g}_j||)}{\max(||\mathbf{s}_i - \mathbf{g}_j||, ||\mathbf{s}_j - \mathbf{g}_i||)} \leq M, \quad (25)$$

we can state the following theorem:

Theorem 1: *If* $\frac{\max(||\mathbf{s}_i - \mathbf{g}_i||, ||\mathbf{s}_j - \mathbf{g}_j||)}{\max(||\mathbf{s}_i - \mathbf{g}_j||, ||\mathbf{s}_j - \mathbf{g}_i||)} \leq M$, $M \in [0, 1)$, *then minimum mutual distance* $d_{ij,min} \geq \sqrt{1 - M^2} \delta_{ij}$.

Considering the assumption in Theorem 1 and (24), the mutual distance is minimized when

$$||\mathbf{s}_i - \mathbf{g}_i|| = ||\mathbf{s}_j - \mathbf{g}_j|| = M \max(||\mathbf{s}_j - \mathbf{g}_i||, ||\mathbf{s}_i - \mathbf{g}_j||), \quad (26)$$

which maximizes $\beta + \gamma$ in range $(0, \pi)$. This corresponds to a situation in which the positions $\mathbf{s}_i, \mathbf{s}_j, \mathbf{g}_i, \mathbf{g}_j$ form vertices of an isosceles trapezoid. The detailed analysis of minimum mutual distance of robots following trajectories formed by diagonals of an isosceles trapezoid is provided in Appendix A, along with proof of Theorem 1.

As a consequence of Theorem 1, a pair of constant-velocity trajectories for which (25) holds is guaranteed to be collision-free under the condition

$$\delta_{ij} \geq \frac{\sqrt{1 - M^2}}{1 - M^2} \Delta. \quad (27)$$

Since we have analyzed the worst-case scenario, the resulting condition (27) forms a lower bound on the minimum distance between a pair of trajectories that can be applied for an efficient mutual collisions check of robots following constant-velocity trajectories.

VII. Algorithm for solution of LBAP with guarantees on minimum distance and collision-free trajectories

As previously stated in Section V, neither the Hungarian algorithm nor its adaptations for LBAP can be used to directly solve the LBAP with constraints on mutually colliding trajectories (8). Using the results obtained in Section VI, we introduce an optimal algorithm for the solution of (7) with additional constraints (8). The algorithm which forms the first component of the proposed CAT-ORA is outlined in Algorithm 1 and detailed in the following sections.

Algorithm 1: Algorithm for robot-to-goal assignment considering mutual collision constraints

Input: sets of initial and goal configurations $\mathbb{S}$, $\mathbb{G}$
Output: complete, collision-free assignment ϕ from $\mathbb{S}$ to $\mathbb{G}$, minimizing the length of the trajectories

```
$\mathbf{M_d}$, $\mathbf{S_d}$, $\mathbf{G_d}$ := preprocessData($\mathbb{S}, \mathbb{G}$)
$t_{lb}$ := getThresholdLowerBound($\mathbf{M_d}$)
$\mathbf{T}, t_c$ := initializeThresholds($\mathbf{M_d}$, $t_{lb}$)
$\mathbf{B}$ := initializeBoundedMatrix($\mathbf{M_d}$,$t_c$)
$\mathbf{u}, \mathbf{v}$ := initializeDualVariables($\mathbf{M_d}$, $\mathbf{B}$)
$\phi$ := findInitialAssignment($\mathbf{M_d}$, $\mathbf{B}$)
done := false
while not done do
    $\phi$ := internalHungarian($\phi, \mathbf{M_d}, \mathbf{B}, \mathbf{u}, \mathbf{v}$)
    valid := isComplete($\phi$)
    if valid then
        coll_edges := getCollidingEdges($\mathbf{M_d}, \mathbf{S_d}, \mathbf{G_d}, \mathbb{S}, \mathbb{G}$)
        if coll_edges = None then
            done := true
        else
            $\phi$ := branchSolution($\phi, \mathbf{M_d}, \mathbf{B}, \mathbf{u}, \mathbf{v}$)
            valid := isComplete($\phi$)
    if not valid then
        $t_c$ := updateThreshold($\mathbf{T}$)
        $\mathbf{B}, \mathbf{e}_u$ := updateBoundedMatrix($t_c$)
        $\mathbf{u}, \mathbf{v}$ := updateMatchingAndDuals($\phi, \mathbf{M_d}, \mathbf{e}_u, \mathbf{u}, \mathbf{v}$)
```

A. Algorithm for robot-to-goal assignment considering mutual collision constraints

To simplify the description of the proposed robot-to-goal assignment algorithm (Algorithm 1), we assume initial and goal configurations $\mathbb{S}$ and $\mathbb{G}$ to be of the same size $|\mathbb{S}| = |\mathbb{G}| = N$, even though this is not strictly required. The algorithm begins with data preprocessing to get the weighted biadjacency matrix $\mathbf{M}_d \in \mathbb{R}^{N\times N}$, $m_{ij} = ||\mathbf{s}_i - \mathbf{g}_j||^2$ and the distance matrices $\mathbf{S}_d \in \mathbb{R}^{N\times N}$, $s_{ij} = ||\mathbf{s}_i - \mathbf{s}_j||^2$, $\mathbf{G}_d \in \mathbb{R}^{N\times N}$, $g_{ij} = ||\mathbf{g}_i - \mathbf{g}_j||^2$ that store the squared distances of particular start and goal locations for efficient collision checking.

Further steps initialize several variables. First, the lower bound t_{lb} for a threshold of elements in $\mathbf{M}_d$ that determines whether edges e_{ij} can be part of the solution is found as an element of $\mathbf{M}_d$

$$t_{lb} = \max(R_{min}, C_{min}), \tag{28}$$

where

$$R_{min} = \max_{i\in 1,\dots,N} \min_{j\in 1,\dots,N} m_{ij}, \tag{29}$$

$$C_{min} = \max_{j\in 1,\dots,N} \min_{i\in 1,\dots,N} m_{ij}. \tag{30}$$

Next, the list of thresholds $\mathbf{T}$ is formed as a sorted list of elements m_{ij} in $\mathbf{M}_d$ which are greater than t_{lb}. The current threshold $t_c = t_{lb}$ is also applied in initialization of a bounding matrix $\mathbf{B} \in \{0,1\}^{N\times N}$, where

$$b_{ij} = \begin{cases} 0 & \text{if} \quad m_{ij} \le t_c, \\ 1 & \text{if} \quad m_{ij} > t_c. \end{cases} \tag{31}$$

The bounding matrix $\mathbf{B}$ is used and updated throughout the whole algorithm to limit the maximum cost of an admissible edge and also to exclude the restricted edges, being part of the collision, from the assignment.

As the next step, the vectors of row and column dual variables $\mathbf{u} = \{u_1, \dots, u_N\}$ and $\mathbf{v} = \{v_1, \dots, v_N\}$ are initialized according to the following rule:

$$v_j = \min_{i\in\{q|b_{qj}=0,\, q\in\{1,\dots,N\}\}} m_{ij}, \ \forall j \in \{1,\dots,N\}, \tag{32}$$

$$u_i = \min_{j\in\{q|b_{iq}=0,\, q\in\{1,\dots,N\}\}} m_{ij} - v_j, \ \forall i \in \{1,\dots,N\}. \tag{33}$$

This initialization ensures that at least one admissible edge is present in each row and column at the beginning of the algorithm. The final step preceding the main loop of the algorithm finds an initial assignment by a sequential search for an arbitrary admissible edge that lies in a yet unassigned row and column. This step is not necessary since the algorithm can start with a valid matching of arbitrary cardinality (including the empty matching), but it decreases the number of required steps in the initial phase of the algorithm.

With the completed initialization, the main loop of the algorithm begins with the *internalHungarian()* procedure (line 9 of Algorithm 1) detailed in Algorithm 2. This procedure starts by searching for an augmenting path through growing the Hungarian trees rooted at the unmatched nodes in a current equality subgraph. If an augmenting path P is found, the matching at step k, ϕ_k is updated by path P as

$$\phi_{k+1} = (P - \phi_k) \cup (\phi_k - P). \tag{34}$$

Otherwise, the dual variables are updated using the set of nodes encountered in the grown Hungarian trees according to the formula

$$u_i = \begin{cases} u_i - \theta & \text{if} \quad i \in \mathbb{H}_r, \\ u_i + \theta & \text{if} \quad i \notin \mathbb{H}_r, \end{cases} \ \forall i \in \{0,\dots,N\}, \tag{35}$$

$$v_j = \begin{cases} v_j + \theta & \text{if} \quad j \in \mathbb{H}_c, \\ v_j - \theta & \text{if} \quad j \notin \mathbb{H}_c, \end{cases} \ \forall j \in \{0,\dots,N\}, \tag{36}$$

where

$$\theta = \frac{1}{2} \min_{i\notin\mathbb{H}_r, j\in\mathbb{H}_c, b_{ij}=0} m_{ij} - u_i - v_j, \tag{37}$$

and $\mathbb{H}_r, \mathbb{H}_c$ are sets of nodes' indices encountered within the Hungarian trees corresponding to the rows and columns of $\mathbf{M}_d$, respectively.

Up to this part, the *internalHungarian()* procedure (Algorithm 2) matches the internal part of the original Hungarian algorithm with the only difference in (37) which excludes the edges restricted by the bounding matrix $\mathbf{B}$. However, this modification can result in an undefined value of θ, indicating that the assignment problem does not have a solution with the current threshold t_c (line 5 in Algorithm 2). In such a case, the *internalHungarian()* procedure is aborted while keeping the incomplete assignment ϕ and updating the values of dual variables $\mathbf{u}, \mathbf{v}$ for later processing inside the main loop of Algorithm 1.

Algorithm 2: internalHungarian($\phi, \mathbf{M_d}, \mathbf{B}, \mathbf{u}, \mathbf{v}$)

Input: matching ϕ, matrix of squared distances $\mathbf{M_d}$, bounding matrix $\mathbf{B}$ marking the elements exceeding current threshold t_c, row and column dual variables $\mathbf{u}, \mathbf{v}$

Output: updated assignment ϕ with non-decreased cardinality, updated row and column dual variables $\mathbf{u}$, $\mathbf{v}$

```
1  while not isComplete($\phi$) do
2      $T_h$ := growHungarianTrees($\phi, \mathbf{B}, \mathbf{u}, \mathbf{v}$)
3      $P$ := findAugmentingPath($T_h$)
4      if $P$ = None then
5          if isUpdateFeasible($T_h, \mathbf{B}$) then
6              updateDualVariables($T_h, \mathbf{u}, \mathbf{v}$)
7          else
8              break
9      else
10         $\phi$ := augmentPath()
```

Once the matching from the *internalHungarian()* procedure is obtained, its completeness is verified (line 10 of Algorithm 1). If the matching is not complete, meaning that its cardinality card(ϕ) $< N$, the threshold t_c and bounding matrix $\mathbf{B}$ are updated. As a result of the Hungarian algorithm on an incomplete graph, the last found matching ϕ has the maximum cardinality on a graph excluding edges bounded by $\mathbf{B}$. Thus, the matching cannot be completed without adding at least $K = N - \text{card}(\phi)$ new edges. Based on this observation, the current threshold t_c is updated to the lowest value in $\mathbf{T}$ that decreases the number of bounded elements in $\mathbf{B}$ by at least K.

The change of the elements in matrix **B** corresponds to the modifications of values in the original cost matrix $\mathbf{M}_d$, which requires updating the dual variables to maintain the dual task feasible. For this purpose, we have adapted the method for updating dual variables in a dynamic (cost-changing) variant of the task assignment problem proposed in [33]. The *updateMatchingAndDuals()* procedure applied within the proposed algorithm (line 21 of Algorithm 1) is outlined in Algorithm 3. After the adaptation of dual variables, the algorithm proceeds to the next run of the *internalHungarian()* procedure (line 9 of Algorithm 1), starting with the matching of cardinality $\text{card}(\phi_k) \geq \text{card}(\phi_{k-1})$ and a decreased number of bounded elements.

```
Algorithm 3: updateMatchingAndDuals($\phi, \mathbf{M_d}, \mathbf{e}_u, \mathbf{u}, \mathbf{v}$)
Input: matching $\phi$, matrix of squared distances $\mathbf{M_d}$,
       set of updated edges $\mathbf{e}_u$, row and column dual
       variables $\mathbf{u}, \mathbf{v}$
Output: updated assignment $\phi$, updated row and
        column dual variables $\mathbf{u}$, $\mathbf{v}$

for $e_{ij} \in \mathbf{e}_u$ do
    if $m_{ij} < u_i + v_j$ then
        $u_i = \min_{k \in \{1,\dots,N\}} m_{ik} - v_k$
        if $e_{ij} \notin \phi$ then
            $\phi := \phi \setminus \{e_{ik}, k \in \{1,\dots,N\}\}$
    else
        $\phi := \phi \setminus e_{ij}$
```

If the matching found by *internalHungarian()* procedure is perfect, it is tested for the existence of colliding edges using a combination of the results derived in Section VI for evaluation of the majority of the potential collisions, and the precise collision checking using (19). The collision check is done over all pairs of edges in the perfect matching ϕ. The collision check of edges $e_{ij}, e_{kl} \in \phi$ starts with the evaluation of

$$\begin{aligned} \text{colide}(e_{ij}, e_{kl}) =& (m_{ij} + m_{kl}) \geq (m_{il} + m_{kj}) \land \\ & \max(m_{ij}, m_{kl}) > M^2 \max(m_{il}, m_{kj}), \end{aligned} \tag{38}$$

where the value of M is set based on the value of δ and the minimum allowed mutual distance Δ using (27). The first part of the condition rejects the risk of potential collision by detecting the equivalence with the LSAP solution with proven guarantees on the minimum distance of trajectories [21] while the second part eliminates the collisions using Theorem 1.

Since the condition from Theorem 1 represents the lower bound on a minimum mutual distance, we further apply the exact computation of a minimum distance to avoid false positive detections of collisions if $\text{collide}(e_{ij}, e_{kl}) = \text{true}$. Thus, if the condition (27) is not met, equation (19) is applied for an exact computation of the minimum mutual distance of the trajectories being compared to the minimum acceptable distance Δ. In the case that there is no pair of colliding edges in a perfect matching ϕ, the algorithm terminates and returns ϕ as a complete assignment from $\mathbb{S}$ to $\mathbb{G}$, minimizing the maximum length of the trajectory while fulfilling the condition on collision-free assignment with constant-velocity trajectories (line 14 of Algorithm 1).

If a collision is detected, the *branchSolution()* procedure (Algorithm 4) is started to ensure that the algorithm explores all possibly collision-free matchings for a current threshold t_c before increasing its value and making new elements of $\mathbf{M}_d$ feasible (line 16 of Algorithm 1). The proposed method is based on the depth-first search algorithm performed on a binary tree graph formed by nodes defined by matching ϕ, pair of colliding edges $\mathbf{e}_c$, and vectors of row and column dual variables $\mathbf{u}, \mathbf{v}$. Note that, the *branchSolution()* method is used to find any collision-free solution with current threshold t_c that defines the optimal value. Thus, the non-optimality of the depth-first search does not influence the optimality of the presented algorithm.

The binary tree, rooted at a node corresponding to initial perfect matching, is iteratively built during the depth-first search by expanding the parent node according to the following expansion rule. The parent node $N_p = \{\phi_p, e_{ij}, \mathbf{u}_p, \mathbf{v}_p\}$ with a maximum matching ϕ_p, restricted edge e_{ij}, and dual variables $\mathbf{u}_p, \mathbf{v}_p$ is, in the case of detected colliding edges $\mathbf{e}_c = \{e_{kl}, e_{op}\}$, expanded into two child nodes derived from the task assignment problem of the parent node by adding a single bounded edge and updating dual variables correspondingly. Thus, the expansion of a node N_p results in new nodes given by

$$\begin{aligned} N_{c1} &= \{\phi_p, e_{kl}, \mathbf{u}_p, \mathbf{v}_p\}, \\ N_{c2} &= \{\phi_p, e_{op}, \mathbf{u}_p, \mathbf{v}_p\}. \end{aligned} \tag{39}$$

Note that, the matching and dual variables of particular nodes are always updated during the *internalHungarian()* procedure before the child nodes are derived from them.

In every iteration of the *branchSolution()* procedure, a node $\mathbf{n}_c = \{\phi_c, e_{ij}, \mathbf{u}_c, \mathbf{v}_c\}$ is dequeued from the Last In First Out (LIFO) queue, the bounding matrix **B** is updated with the newly restricted edge e_{ij} and the corresponding dual variables and matching are updated using *updateMatchingAndDuals()* (Algorithm 3). After that, the *internalHungarian()* is run to find a perfect matching for an updated assignment problem (line 8 of Algorithm 4). Since the newly restricted edge e_{ij} is always part of an initial parent matching, the cardinality of the matching ϕ after an update is always $\text{card}(\phi) = N - 1$. As mentioned earlier, given the matching of cardinality $N - 1$ and the corresponding dual variables, the internal Hungarian algorithm terminates after a single step with either a perfect matching (if an augmenting path exists) or an incomplete matching. This fact is important for keeping the computational complexity of the proposed algorithm low.

If a perfect matching is not found by the *internalHungarian()* method, the solution to an assignment problem with a set of bounded edges given by **B** does not exist. Consequently, it is easy to show that this situation cannot be improved by restricting additional edges. Hence, we cannot get a valid solution by expanding such a node and can proceed to the next iteration. If the computed solution is a perfect matching, it has to be examined whether it is collision-free. In case a pair of colliding edges is not found, the perfect matching is returned as a valid collision-free solution to the main loop of Algorithm 1. Otherwise, the node is expanded according to the expansion rule (39), inserted into the queue, and the algorithm proceeds to the next iteration (line 14 of Algorithm 4).

In case all branches of the tree were explored without finding a valid, complete solution, the procedure returns to the main loop of Algorithm 1, where the value of the current threshold is updated (line 19), and the search for a solution continues. In the main loop, the algorithm repeats the above-described steps until a valid, complete solution is found. The valid solution is guaranteed to exist under the assumptions specified in Section III.

Since the introduced algorithm iteratively increases the threshold on bounded edges, the number of bounded edges decreases and the problem becomes less restricted. In a worst-case scenario, the algorithm reaches a point where none of the edges are bounded, meaning that also none of the edges from solution ϕ_{LSAP} minimizing the sum of squared costs are bounded. Then, ϕ_{LSAP} is an output of the internal Hungarian algorithm. Since the ϕ_{LSAP} solution is guaranteed to be collision-free with given assumptions, a valid collision-free matching is always found.

Algorithm 4: branchSolution($\mathbf{e}_c, \phi, \mathbf{M_d}, \mathbf{B}, \mathbf{u}, \mathbf{v}, \mathbb{S}, \mathbb{G}, \mathbf{S}_d, \mathbf{G}_d$)

```
Input: colliding edges $\mathbf{e}_c$, matching $\phi$, matrix of
       squared distances $\mathbf{M_d}$, bounding matrix $\mathbf{B}$, row
       and column dual variables $\mathbf{u}, \mathbf{v}$, initial and goal
       configurations $\mathbb{S}$, $\mathbb{G}$, matrices with squared
       distances of initial and goal configurations
       $\mathbf{S}_d, \mathbf{G}_d$
Output: complete collision-free assignment $\phi$ if it
        exists, original assignment otherwise
$\mathbf{O}_l := \emptyset$          // Last in first out queue
Node $root := \{\phi, \mathbf{e}_c, \mathbf{u}, \mathbf{v}\}$
$\mathbf{O}_l \leftarrow$ expand($root, \phi, \mathbf{e}_c$)
while $\mathbf{O}_l \neq \emptyset$ do
    $\mathbf{n}_c :=$ dequeue($\mathbf{O}_l$)
    updateRestrictedNodes($\mathbf{M_d}, \mathbf{B}, \mathbf{n}_c$)
    updateMatchingAndDuals($\phi, \mathbf{M_d}, \mathbf{n}_c, \mathbf{u}, \mathbf{v}$)
    $\phi :=$ internalHungarian($\phi, \mathbf{M_d}, \mathbf{B}, \mathbf{u}, \mathbf{v}$)
    if isComplete($\phi$) then
        $\mathbf{e}_c :=$ getCollidingEdges($\mathbf{M_d}, \mathbf{S_d}, \mathbf{G_d}, \mathbb{S}, \mathbb{G}$)
        if $\mathbf{e}_c$ = None then
            return $\phi$           // solution found
        else
            $\mathbf{O}_l \leftarrow$expand($\mathbf{n}_c, \phi, \mathbf{e}_c$)
    else
        continue
return $\phi$                // solution not found
```

VIII. Minimum-makespan trajectory generation

The algorithm designed for the solution of the LBAP introduced in Section VII-A guarantees to solve part of TOFREP by finding an assignment minimizing the length of the longest path with additional guarantees on the collision-free property of the constant-velocity trajectories. In this section, we describe a second component of CAT-ORA that allows us to generate a set of trajectories connecting the pairs of assigned positions, while minimizing the makespan of the reshaping process and preserving the conditions on the collision-free property derived in Section VI.

A. Minimum-time trajectory generation

The time-optimal control of a model with single-dimension point-mass dynamics and constraints on maximum velocity results in a control policy of form

$$a^*(t) = \begin{cases} a_{max} & \text{if} \quad t \leq t_1, \\ 0 & \text{if} \quad t_1 < t \leq t_2, \\ -a_{max} & \text{if} \quad t_2 < t \leq t_3, \end{cases} \tag{40}$$

where $t_3 = t^*$ is the overall minimized time of trajectory following [24]. The control policy (40) leads to trajectories that are described by equations

$$\begin{aligned} p_1 &= p_0 + v_0 t_1 + \frac{1}{2} a_{max} t_1^2, \\ v_1 &= v_0 + a_{max} t_1, \\ p_2 &= p_1 + v_1(t_2 - t_1), \\ v_2 &= v_1, \\ p_3 &= p_2 + v_2(t_3 - t_2) - \frac{1}{2} a_{max}(t_3 - t_2)^2, \\ v_3 &= v_2 - a_{max}(t_3 - t_2), \end{aligned} \tag{41}$$

with velocities v_i and positions $p_i, i \in \{0,1,2,3\}$. With the known initial and final conditions of p_0, v_0, p_3, v_3, the maximum acceleration a_{max}, and the assumption on reachability of the maximum velocity $v_1 = v_{max}$, the number of equations (41) matches the number of unknown variables $(t_1, t_2, t_3, p_1, p_2, v_2)$, and (41) has a closed-form solution. By the addition of an assumption that $v_0 = v_3 = 0$ and its consequence $t_1 = t_3 - t_2$, (41) can be modified to

$$\begin{aligned} p_1 &= p_0 + \frac{1}{2} a_{max} t_1^2, \\ v_1 &= a_{max} t_1, \\ p_2 &= p_1 + v_1(t_2 - t_1), \\ v_2 &= v_1, \\ p_3 &= p_2 + v_2(t_3 - t_2) - \frac{1}{2} a_{max}(t_3 - t_2)^2, \\ v_3 &= v_2 - a_{max}(t_3 - t_2), \\ t_1 &= t_3 - t_2. \end{aligned} \tag{42}$$

With an additional seventh equation, the modified set of equations (42) allows for relaxing the condition $v_1 = v_{max}$ to $v_1 \leq v_{max}$, thus providing a single closed-form solution valid even when maximum velocity cannot be reached, and the optimal control policy reduces to bang-bang control [30].

B. Trajectories for minimum-time formation reshaping

By applying time optimal control policy (40), the time necessary for following the trajectory is directly proportional to the length of the trajectory. Thus, the makespan of the formation reshaping process is equal to the time t_m of following the trajectory T_m corresponding to the longest path P_m obtained in the assignment ϕ

$$|P_m| = D_m = \max_{(i,j) \in \phi} ||\mathbf{s}_i - \mathbf{g}_j||. \tag{43}$$

From the solution of (42) for the longest path P_m, we obtain the duration t_1^m, t_2^m and t_3^m of acceleration, constant speed, and deceleration segments, respectively. Considering the duration of particular segments t_1, t_2, t_3 to be constant and equal to t_1^m, t_2^m, t_3^m for all trajectories, (42) can be applied for the generation of the rest of trajectories with defined t_1, t_2, t_3, but varying $|a_i| \leq a_{max}$ and $v_1 \leq v_{max}$. Such an approach results in trajectories defined by parametrization:

$$\begin{aligned}
&0 \leq s \leq \frac{t_1}{t_3} : \ \mathbf{x}_i(s) = \mathbf{s}_i + \left(\frac{A_m t_3^2}{2D_m} s^2\right)(\mathbf{g}_j - \mathbf{s}_i),\\
&\frac{t_1}{t_3} < s \leq \frac{t_2}{t_3} : \ \mathbf{x}_i(s) = \mathbf{s}_i + \left(\frac{V_m t_3}{D_m} s - \frac{V_m^2}{2A_m D_m}\right)(\mathbf{g}_j - \mathbf{s}_i),\\
&\frac{t_2}{t_3} < s \leq 1 \ : \\
&\mathbf{x}_i(s) = \mathbf{s}_i + \left(\frac{A_m t_3^2}{2D_m}(2s - s^2) - \frac{V_m^4 + A_m^2 D_m^2}{2A_m D_m V_m^2}\right)(\mathbf{g}_j - \mathbf{s}_i),
\end{aligned} \tag{44}$$

where V_m and A_m stand for maximum applied velocity and maximum applied acceleration of T_m, respectively, and $s = \frac{t}{t_m}$ with t being time elapsed from start of the trajectory. Trajectories generated according to parametrization (44) are illustrated in Fig. 5.

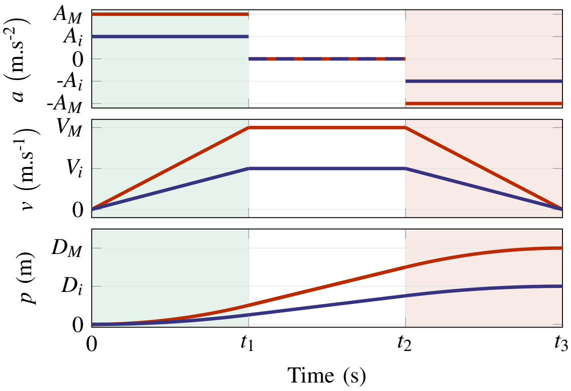


Fig. 5: Acceleration and velocity profiles and position progress along the path for trajectory T_j with length $D_j = D_m$ generated using time-optimal control policy (40) (red), and for trajectory T_i with length $D_i \leq D_m$ generated according to parametrization (44) (blue). The background color distinguishes the acceleration (green), constant speed (white), and deceleration (red) segments of the trajectories.

Let us define the progress ratio at time $t > 0$ for a pair of trajectories T_i, T_j of lengths $D_i > 0$, $D_j > 0$, as

$$PR(T_i, T_j, t) = \frac{p_i(t)}{p_j(t)}, \tag{45}$$

where $p_i(t), p_j(t)$ are the distances traveled along the trajectories T_i, T_j till time t.

Theorem 2: *If the progress ratios $PR_a = PR(T_{i,a}, T_{j,a})$ and $PR_b = PR(T_{i,b}, T_{j,b})$ of two parametrizations a and b of a single pair of paths are constant for all t and $PR_a = PR_b$, the following relation holds:*

$$dist_{min}(T_{i,a}, T_{j,a}) = dist_{min}(T_{i,b}, T_{j,b}). \tag{46}$$

Proof. The assumption $PR_a = PR_b = const.$ in Theorem 2 can be reformulated to equation

$$\frac{p_{i,a}(s_a t_a)}{p_{j,a}(s_a t_a)} = \frac{p_{i,b}(s_b t_b)}{p_{j,b}(s_b t_b)}, \ \forall s_a, s_b \in (0, 1], \tag{47}$$

where t_a and t_b are the duration of trajectories $T_{i,a}, T_{j,a}$ and $T_{i,b}, T_{j,b}$, respectively, and s_a, s_b are independent variables. Equation (47) can be simplified to

$$p_{j,b}(s_b) = p_{j,a}(s_a) \frac{p_{i,a}(s_a)}{p_{i,b}(s_b)}. \tag{48}$$

From (48), it follows that for an arbitrary $p_{i,a}(s_a)$ with associated point $p_{j,a}(s_a)$, it holds that

$$p_{i,b}(s_b) = p_{i,a}(s_a) \Leftrightarrow p_{j,a}(s_a) = p_{j,b}(s_b). \tag{49}$$

Thus, the set of corresponding points on particular trajectories is equal for both pairs of trajectories, and their minimum distances are equal. □

By applying Theorem 2, it can be shown that the progress ratio of trajectories with a constant velocity (10) and minimum-time trajectories defined by (44) are equal and constant for all parts of parametrization (44). It follows that the minimum-time trajectories defined by (44) have the same properties that were derived for trajectories with constant velocity. Thus, the pair of minimum-time trajectories are guaranteed to be collision-free if $\delta \geq \frac{\sqrt{1-M^2}}{1-M^2}\Delta$ and the CAT-ORA consisting of application of Algorithm 1 together with the trajectory generation approach (44) provides a set of minimum-time collision-free trajectories as a solution to TOFREP.

IX. Theoretical and Statistical Analysis

In this section, we provide proof of the optimality of the proposed algorithm and state and prove several theorems that highlight the significant benefits of the proposed approach, and advocate the rightfulness of the stated assumptions.

A. The independence of robot-to-goal assignment on trajectory generation approach

The proposed decoupled solution to TOFREP is optimal under a necessary assumption on the separability of the robot-to-goal assignment problem and minimum-time trajectory generation (4). If (4) holds for all assignments and trajectories generated by the approach described in Section VIII, then the duration of trajectories can be replaced by Euclidean distances in the computation of robot-to-goal assignment without affecting the optimality of the solution.

The proof of (4) comes directly from (43), (44) and properties of applied control policy (40). Based on (43), (44), the duration of all trajectories for a single assignment depends only on the length of the longest path in the assignment. Given the assumptions on stationary initial and goal configurations (A2), the trajectory generation process generates trajectories with the same initial and final velocities. In such a case, the duration of trajectories generated using time-optimal control policy (40) is a monotonic, increasing function of the length of the path, which guarantees legitimacy of the assumption (4).

B. Optimality of the robot-to-goal assignment of CAT-ORA

The proof of the optimality of the proposed algorithm is based on the following observations:

(B1) The Hungarian algorithm, and thus also the *internalHungarian()* procedure, are optimal (proved in [10]).

(B2) The dynamic variant of the Hungarian algorithm is optimal (proved in [33]).

(B3) The bounding of elements $m_{ij} > t_c$ of a cost matrix $\mathbf{M}_d$ by bounding matrix $\mathbf{B}$ is equivalent to substituting constant $Q = \sum_{m_{ij} \in \mathbf{M}_d} [m_{ij} \leq t_c] m_{ij}$ for all elements $m_{ij} > t_c$.

(B4) If the element m_{ij} applied in computation of updates to dual variables (37) is bounded, all unbounded elements of M_d are already admissible, and the cardinality of the current matching cannot be increased without using bounded elements (comes from the properties of the Hungarian algorithm).

(B5) By the addition of k edges, the cardinality of the maximum matching ϕ can be increased by at most k. Thus, the matching ϕ cannot be completed without the addition of at least $k = N - \text{card}(\phi)$ new edges (comes from the properties of the Hungarian algorithm).

The proof of completeness comes straightforwardly from (B1) and (B2). By omitting the procedures that do not change any variables in the main loop of Algorithm 1, the algorithm reduces to a dynamic variant of the Hungarian algorithm, solving the assignment problem with iterative change of costs caused by updates of threshold t_c. Based on (B3), the bounded elements only influence the update of duals once they are smaller than t_c. The threshold is updated until a valid solution is found, eventually ending with $b_{ij} = 0 \, \forall (i,j) \in \{1, \ldots, N\}^2$. If no elements of M_d are bounded, the solution exists according to assumption (A5). Then, in compliance with (B2), the solution is found, proving the algorithm's completeness.

The proof of optimality is built using the fact that the bounded values cannot be part of the solution, and thus the optimal value of the solution is bounded by t_c. Therefore, it is sufficient to show that the threshold t_c is increased only if a valid solution does not exist with the current threshold. According to Algorithm 1, the initial threshold (28) equals the greatest element among minimum elements of particular rows and columns. Since the perfect matching has to contain at least one element from each row and column, the initial lower bound does not exceed the optimal value. Further, we branch the proof to two cases: (i) Algorithm 1 never detects a colliding edge or (ii) Algorithm 1 detects a colliding edge.

In case (i), the algorithm switches between applying the *internalHungarian()* procedure and updating the threshold t_c. According to (B1) and (B4), the *internalHungarian()* procedure always finds a maximum matching with respect to the current bounded matrix. If a solution found with the current threshold is a perfect matching, the optimal solution was found. Otherwise, the threshold is updated. From (B5), we can conclude that an update of t_c to the lowest value that decreases the number of bounded elements in $\mathbf{B}$ by K, cannot increase the threshold above the value of the optimal solution. Then, for case (i), the procedure matches the algorithm [33] with the costs changed by updates of t_c. Thus, the guarantees on optimality for case (i) can be concluded based on (B2) and the validity of threshold updates given by (B5).

In case (ii), we have to further prove that the procedure *branchSolution()* is complete. The constraint on mutually colliding edges represents edges e_r, e_s from which only one can be a part of the solution. The *branchSolution()* procedure exploits this fact by creating a binary search tree where each branch is derived from a parent node by restricting exactly one edge from the colliding pair. As a result, the original instance is split into two, where one of the instances allows us to find any solution to the original problem that does not include e_r, and the second allows us to find any solution to the original problem that does not include e_s. Clearly, this approach cannot miss any valid solution. During the search for the solution, each node represented by the instance of an assignment problem is evaluated using the *internalHungarian()* and possibly *getCollidingEdges()* procedure. For each node, there are three possible outputs of the evaluation with the corresponding set of restricted edges: the found matching is not perfect; the found matching is perfect, but contains colliding edges; and the found matching is perfect and collision-free. If a perfect matching is not found, it does not exist ((B1), (B2)), and thus this branch of the solution does not have to be explored further since the restriction of an additional edge cannot lead to an increase in cardinality. If a perfect matching is found and it contains a pair of colliding edges, the node is split into two and further explored. If the found matching is perfect and does not contain any colliding edge, it is bounded by t_c, and is thus optimal with respect to (7), (8).

Since we prove the optimality of the algorithm for the solution of LBAP with mutual collision constraints, as well as the independence of robot-to-goal assignment on minimum-time trajectory generation approach, it can be concluded that the CAT-ORA is an optimal algorithm for the solution of TOFREP.

□

C. Comparison of LBAP and LSAP in terms of maximum path length

The superior performance of the LBAP-based solution to robot-to-goal assignment in terms of the length of the longest path is evident from the LSAP and LBAP problem formulation. With Theorem 3 introduced and proven in this section, we provide an insight into the significance of this phenomenon, and thus also the benefit of solving robot-to-goal assignment as LBAP instead of LSAP. The theorem shows that the LSAP solution can produce an up to 1.7-times longer longest path compared to the LBAP solution already for small instances of 3 robots. This ratio further grows with the squared root of a number of robots, where for 100 robots, it already reaches a ratio of 10.

Theorem 3: *The upper bound on the ratio between the maximum length of the path in LSAP assignment ϕ_s and LBAP assignment ϕ_b is $\sqrt{N}$, where N is the number of goals. The lower bound on this ratio equals 1. Thus, the following equation holds*

$$1 \leq \frac{\max_{(i,j)\in\phi_s} ||\mathbf{s}_i - \mathbf{g}_j||}{\max_{(i,j)\in\phi_b} ||\mathbf{s}_i - \mathbf{g}_j||} \leq \sqrt{N}. \tag{50}$$

Proof. The proof of the first inequality comes directly from the *LBAP* algorithm. If no edge could be removed from the LSAP solution unless the problem becomes unfeasible, the maximum lengths of the paths of LSAP and LBAP solutions coincide.

The second inequality can be proved by considering a general set of paths resulting from assignment ϕ_s and a set of paths resulting from assignment ϕ_b. We would like to prove that

$$\frac{\max_{(i,j)\in\phi_s}||\mathbf{s}_i-\mathbf{g}_j||}{\max_{(i,j)\in\phi_b}||\mathbf{s}_i-\mathbf{g}_j||}\leq\sqrt{N}, \tag{51}$$

$$\text{if}\sum_{(i,j)\in\phi_s}||\mathbf{s}_i-\mathbf{g}_j||^2\leq\sum_{(i,j)\in\phi_b}||\mathbf{s}_i-\mathbf{g}_j||^2. \tag{52}$$

Considering the independence of the expressions in the numerator and denominator, the maximum of this expression can be found by independent maximization of numerator and minimization of denominator while conforming to constraint (52). The task can be simplified using the following observations:

i. The paths shorter than $\max_{(i,j)\in\phi_s}||\mathbf{s}_i-\mathbf{g}_j||$ do not influence the value of numerator, but increase the value of the left side of (52). In the extreme case, this leads to a set of paths with zero length except for a single path in the set.

ii. The paths shorter than $\max_{(i,j)\in\phi_b}||\mathbf{s}_i-\mathbf{g}_j||$ do not influence the value of the denominator, but increase the value of the right side of (52). In the extreme case, this leads to a set of paths with equal lengths.

Based on these observations, (52) can be rewritten into the least constrained form

$$\left(\max_{(i,j)\in\phi_s}||\mathbf{s}_i-\mathbf{g}_j||\right)^2\leq N\left(\max_{(i,j)\in\phi_b}||\mathbf{s}_i-\mathbf{g}_j||\right)^2, \tag{53}$$

which can be further reformulated to

$$\frac{\max_{(i,j)\in\phi_s}||\mathbf{s}_i-\mathbf{g}_j||}{\max_{(i,j)\in\phi_b}||\mathbf{s}_i-\mathbf{g}_j||}\leq\sqrt{N}. \tag{54}$$

□

D. Study on suboptimality of the solution to TOFREP neglecting assumption on straight paths

The optimality of the CAT-ORA for the solution of TOFREP is proved in Section IX-B. However, the assumption (A3) discriminates the use of collision resolution techniques, such as time delays and geometric modifications of paths [24], [38]. Although the use of such techniques mostly leads to a significant increase in the computational complexity of the algorithm, they can resolve some collisions that are unsolvable by the proposed algorithm without increasing the threshold t_c. Thus, neglecting the assumption (A3) can change the optimum value of TOFREP.

We compare the achieved optimum value of the proposed algorithm with a theoretical lower bound of TOFREP (see Theorem 4) to analyze the gap between the optimum value of the solution while both considering and not considering assumption (A3). The results presented in Fig. 6 show that the CAT-ORA yields an optimal solution equal to the theoretical lower bound (neglecting assumption (A3)) in more than 95% of instances in dense environments with an average suboptimality $s_{avg} = 1.0008$ and maximum suboptimality $s_{max} = 1.16$.

Theorem 4: *The lower bound on the solution of TOFREP without limitations imposed by assumption (A3) is given by the duration of the minimum-time trajectory that corresponds to the longest path in the robot-to-goal assignment, as obtained by the solution of LBAP without considering mutual collisions.*

Proof. Considering an assumption on static initial and goal configuration, the optimum value of TOFREP equals the duration of minimum-time trajectory along the longest path among all reshaping paths. Thus, minimizing the length of the longest path among all reshaping paths optimizes the original problem. By applying any technique to resolve the collisions among trajectories, the optimum value remains the same or increases. Thus, the solution of LBAP together with the minimum-time trajectory generation forms a lower bound to TOFREP. □

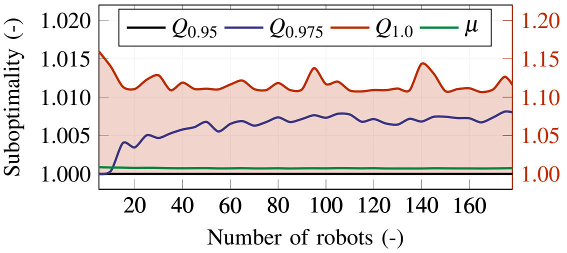


Fig. 6: Quantitative analysis of the suboptimality of the CAT-ORA to TOFREP omitting assumption (A3). The presented results are generated using 10^5 instances with a density of the environment $d_r = 0.1$ for every number of robots in $[5, 175]$. Q_x stands for corresponding quantiles and μ stands for the mean value. The curve of $Q_{1.0}$ is associated with the values on the right axis.

X. Numerical and Experimental Results

In this section, numerical and experimental results are presented to demonstrate the performance indicators of the proposed approach. All results were evaluated in scenarios with varying numbers and densities of the robots randomly generated in a 3D environment. The density of robots in the environment of volume V_e is defined as

$$d_r = \frac{\sum_{r=1}^{N} V_r}{V_e}, \tag{55}$$

where V_r stands for the volume occupied by particular robots. All evaluations were performed on a computer with a 4-core Intel$^{(R)}$ Core(TM) i7-10510U CPU @ 1.80GHz.

A. The effect on length of the path

Although the presented approach is focused on minimizing the makespan of the formation reshaping process, the comparison based on the duration of the trajectories would depend on the choice of dynamic constraints. Thus, it would not yield fair results. Therefore, we compare the solutions provided by our algorithm for robot-to-goal assignment in terms of maximum length of the path with the solutions of LSAP used by several state-of-the-art works [21], [34]–[36]. The results show that, on average, the CAT-ORA produces a set of paths with a maximum length 10% shorter than the LSAP approach. This highlights the significant benefit of using CAT-ORA instead of LSAP-based approaches, especially for battery-constrained robots or time-constrained applications. The detailed results for various numbers of robots and densities of the environment are presented in Fig. 7. In compliance with Theorem 3, a more significant effect is observed for instances with more robots.

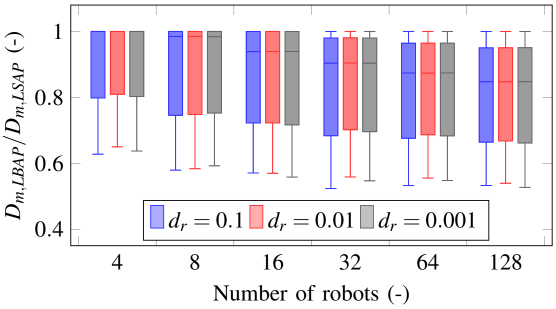


Fig. 7: The ratio between the maximum length of the set of paths produced by LBAP approach $D_{m,LBAP}$ and by the LSAP approach $D_{m,LSAP}$ for a varying number of robots and density of robots in the environment d_r. The results are generated using 10^5 instances for each presented number of robots.

B. Computational time

The introduced procedures and checks guaranteeing the optimality of the CAT-ORA come at the cost of higher computational times in comparison to the original Hungarian algorithm. A major increase in the computational burden may potentially come from the *branchSolution()* method. However, reaching its theoretical asymptotic complexity would mean that all LBAP solution edges mutually collide. The probability of this situation is limited by the assumption (A5) and by solving the LBAP as an LSAP with restrictions on certain edges. This brings the advantage that any pair of edges $e_{i,j}, e_{k,l}$ is guaranteed to be collision-free if $e_{i,l} \leq t_c$ and $e_{k,j} \leq t_c$. Thus, in practice, the *branchSolution()* method is responsible for 4.7% of the total computational time on average among 10^5 randomly generated instances with high density.

A detailed analysis has shown that the main part of the additional time required by CAT-ORA is not consumed by the collision resolution part, but by the search for a correct threshold for the feasible solution. Since some algorithms for the solution of LBAP have lower theoretical complexity than those for the solution of LSAP, they can be used to increase the efficiency of a search for the threshold t_c. However, their application in Algorithm 1 is limited by the crucial role of dual variables that would require running the algorithm from its initial phase after the threshold is found. Therefore, such an approach is efficient only for instances with a high number of robots and an inaccurate initial estimate of threshold t_{lb}.

The detailed comparison of computational times of the algorithm is shown in Fig. 8. Although the ratio between the maximum computational times of the Hungarian algorithm, applied for the solution of LSAP and CAT-ORA is significant, the absolute maximum difference in times does not exceed a few milliseconds for the instances with up to 32 robots. This keeps the computational demands sufficiently low for using CAT-ORA in applications that require real-time computations. The ratio between computational times decreases with an increasing number of robots since it mitigates the effect of the more demanding initialization phase.

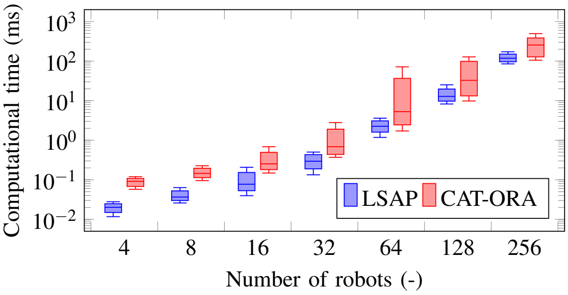


Fig. 8: Comparison of computational demands of the LSAP approach and CAT-ORA approach for varying numbers of robots in an environment. The presented results are generated using 10^5 instances with varying densities of the environment.

C. Formation reshaping

We benchmark the CAT-ORA by comparing the achieved results with the LSAP and LBAP algorithms coupled with minimum-time trajectories. Similarly to the detailed results in the previous section, the algorithms were evaluated on a set of 10^5 instances representing formation reshaping tasks with various numbers and densities of robots in an environment. While the comparison results can be easily inferred from the characteristics of the individual algorithms, the presented results, as detailed in Table I, quantitatively demonstrate the expected outcomes. The LBAP-based consistently yields solutions with shorter maximum path lengths compared to other methods, resulting in a reduced makespan. However, the generated trajectories lead to collisions in more than 6% of instances. The CAT-ORA and LSAP-based approach provide collision-free trajectories for all instances. Yet, while the LSAP-based solution leads to an average increase of 12% in makespan and 15% in maximum path length compared to LBAP-based approach, the CAT-ORA only marginally extends the duration of reshaping process by an average of 0.06% compared to LBAP-based approach.

The advantage of the CAT-ORA over the LSAP-based algorithm is showcased in a scenario requiring 200 robots initially arranged in a rectangular formation to sequentially adapt the formation shape to represent the letters C, T, and U. Guided by the CAT-ORA, the entire formation reshaping task is completed in 36.6 s, which is 5.4 s faster than the solution provided by the LSAP-based approach. A detailed presentation of a specific formation reshaping instance is provided in Fig. 9.

D. Real-world experiment

In the real-world experiment, the CAT-ORA was applied in a scenario simulating a small-scale drone performance. The scenario requires a set of robots to perform 19 transitions between

TABLE I: Comparison of different approaches for the solution of formation reshaping task. The values were computed from results obtained for 10^5 instances of varying density $d_r \in [0.0001, 0.1]$, varying number of robots $N \in [10, 210]$, $v_{max} = 4\,\mathrm{m\,s^{-1}}$, and $a_{max} = 2\,\mathrm{m\,s^{-2}}$. The *PDB* value for statistics x and method m is computed as $PDB(x,m) = (x_m - x_{best})/x_{best}$, where x_{best} is the best value of x among all methods for a particular instance.

Approach	Success rate (%)	Makespan PDB (%)		Max. length PDB (%)		Total length PDB (%)		Comp. time PDB (%)	
		mean	std. dev.	mean	std.dev	mean	std.dev	mean	std.dev
LSAP, min. time	**100.0**	12.17	8.54	15.82	11.18	**0.00**	0.00	**0.06**	1.69
LBAP, min. time	93.19	**0.00**	0.00	**0.00**	0.00	3.11	2.82	244.52	346.24
CAT-ORA	**100.0**	0.06	0.48	0.07	0.63	3.11	2.83	266.23	372.82

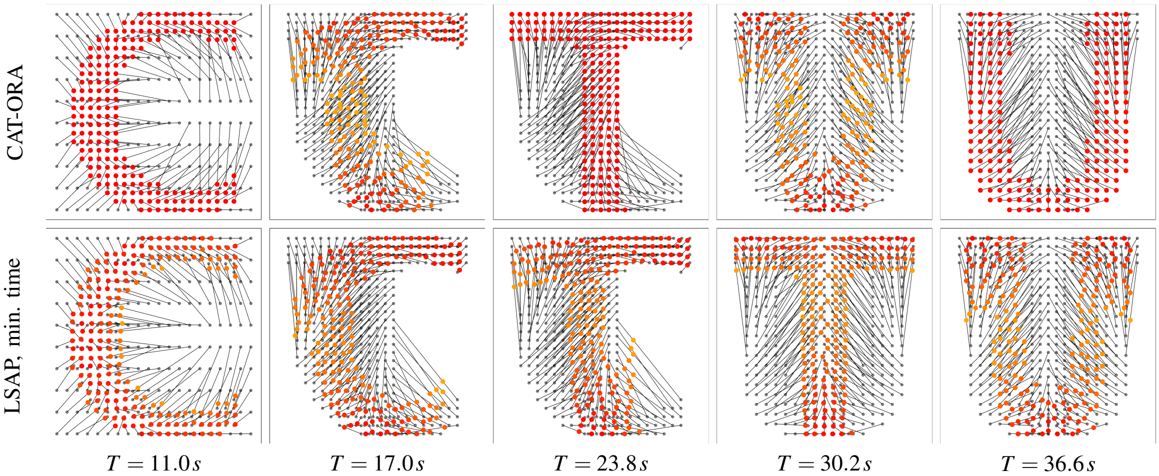


Fig. 9: A qualitative comparison of the formation reshaping process applying CAT-ORA and approach applying LSAP solution coupled with minimum-time trajectory generation. The formation consists of 200 robots that are initially organized in a rectangular formation and are consequently required to adapt the shape of the formation to represent letters C, T, and U. The applied dynamic constraints are $v_{max} = 4.0\,\mathrm{m\,s^{-1}}$, and $a_{max} = 2.0\,\mathrm{m\,s^{-2}}$. The height of each letter is 100 m and the scale of the axis is equivalent. The gray lines represent the reshaping paths, and the colored points represent positions of robots at corresponding times. The color encodes the velocity of particular robots, with red being equal to zero velocity and yellow to v_{max}.

formations of diverse shapes (both 2D and 3D) and sizes, while the center of the formation continuously moves through the environment. Each formation $\mathbb{F}_i = \{\mathbf{r}_1, \dots, \mathbf{r}_N\}$ is defined by a set of desired relative positions to the center of the formation $\mathbf{r}_j \in \mathbb{R}^3$ defined in the orthogonal coordinate system H that coincides with the position and orientation of the center of the formation. Since the requirement on continuous movement contradicts the assumption (A2) on robots being stationary in the initial and goal configurations, CAT-ORA cannot be directly applied to compute trajectories between the robots' configurations defined in the world coordinate frame W.

However, the definition of relative positions in an orthogonal coordinate system ensures independence of mutual distances between desired relative positions on the motion of the formation. Therefore, we apply CAT-ORA to compute the trajectories in the space of relative positions considering consecutive formations $\mathbb{F}_i, \mathbb{F}_{i+1}$ as initial and goal configurations, respectively. The generated trajectories then define the time evolution of $\mathbf{r}_j$, leading to continuous adaptation of the formation shape. The trajectories in the world coordinate frame are then defined by $\mathbf{p}_j(t) = \mathbf{T}_{H,W}\mathbf{r}_j(t), t \in (t_0, t_f)$, where $\mathbf{T}_{H,W}$ is a transformation matrix from formation frame H to world coordinate frame W. This approach shows that the assumption (A2) is not a strict requirement for the applicability of CAT-ORA to solve TOFREP. However, the superposition of the generated trajectories to the trajectory of the center of the formation requires adapting the dynamic constraints for the generation of formation reshaping trajectories, such that the resulting trajectories $\mathbf{p}_j(t), j \in \{1, 2, \dots, N\}, t \in (t_0, t_f)$ do not violate the dynamic constraints. Thus, (A3) is a necessary assumption for guaranteeing the optimality of the solution.

The real-world experiment was performed with 19 multirotor helicopters [39], [40] using the MRS UAV system for low-level control and trajectory tracking [41]. The time required for the whole performance was 294.80 s with the total time of reshaping 214.8 s and an overall computational time of 60 ms. Snapshots from the experiment are shown in Fig. 10 and Fig. 1.

XI. Conclusion

This paper introduces an algorithm named CAT-ORA (Collision-Aware Time-Optimal formation Reshaping Algorithm) to address the time-optimal formation reshaping problem while considering mutual collision avoidance among robots. It showcases superior performance in terms of the makespan of the formation reshaping process, while maintaining computational demands at a level suitable for real-time deployment, even in formations comprising up to one hundred robots. The properties of the proposed algorithm have

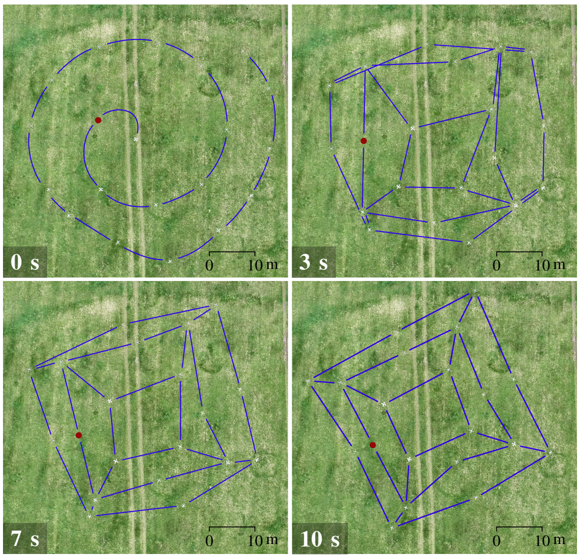


Fig. 10: Snapshots from a real-world experiment showing the transition between a 3D spiral and a pyramid shape. The transition was completed within 10 seconds during continual rotation of the formation. The red point represents a missing UAV that failed to start due to a HW failure. Blue lines highlight the shape of the formation. Since the images show a 3D formation, the measuring scale is approximate.

been evaluated by thorough numerical and theoretical analysis, including the proof of optimality, and the applicability of the algorithm in practical scenarios was demonstrated through simulations and real-world experiments.

Notably, the results highlight a significant advantage of the robot-to-goal assignment aspect within CAT-ORA. It reduces the maximum length of the assigned path by up to 48% compared to the LSAP-based methods utilized by state-of-the-art approaches in cooperative motion planning and formation control. This finding holds particular significance for aerial vehicles with constrained operational time, as it enhances their performance during a real-world deployment. Moreover, this outcome has potential implications for future research on formation reshaping focused on the deployment of autonomous robots in general environments.

References

[1] M. Tranzatto *et al.*, "CERBERUS in the DARPA Subterranean Challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[2] M. Petrlík *et al.*, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," *Field Robotics*, vol. 3, no. 1, pp. 1–68, 2023.

[3] K. Ji *et al.*, "A Virtual Force Interaction Scheme for Multi-Robot Environment Monitoring," *Robotics and Autonomous Systems*, vol. 149, p. 103967, 2022.

[4] K. Azadeh *et al.*, "Robotized and Automated Warehouse Systems: Review and Recent Developments," *Transportation Science*, vol. 53, no. 4, pp. 917–945, 2019.

[5] P. Petracek *et al.*, "New Era in Cultural Heritage Preservation: Cooperative Aerial Autonomy: Supervised Autonomy for Fast Digitalization of Difficult-to-Access Interiors of Historical Monuments," *IEEE Robotics & Automation Magazine*, pp. 2–19, 2023.

[6] F. Afghah *et al.*, "Wildfire Monitoring in Remote Areas Using Autonomous Unmanned Aerial Vehicles," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops*, 2019, pp. 835–840.

[7] M. Waibel *et al.*, "Drone Shows: Creative Potential and Best Practices," ETH Zurich, Tech. Rep., 2017.

[8] S. J. Kim *et al.*, "A Survey of Drone Use for Entertainment and AVR (Augmented and Virtual Reality)," *Augmented Reality and Virtual Reality: Empowering Human, Place and Business*, pp. 339–352, 2018.

[9] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[10] J. R. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of The Society for Industrial and Applied Mathematics*, vol. 10, pp. 196–210, 1957.

[11] L. Antonyshyn *et al.*, "Multiple Mobile Robot Task and Motion Planning: A Survey," *ACM Comput. Surv.*, vol. 55, no. 10, 2023.

[12] S. Poudel and S. Moh, "Task Assignment Algorithms for Unmanned Aerial Vehicle Networks: A Comprehensive Survey," *Vehicular Communications*, vol. 35, p. 100469, 2022.

[13] G. M. Skaltsis *et al.*, "A Survey of Task Allocation Techniques in MAS," in *2021 International Conference on Unmanned Aircraft Systems*, 2021, pp. 488–497.

[14] F. Quinton *et al.*, "Market Approaches to the Multi-Robot Task Allocation Problem: a Survey," *Journal of Intelligent & Robotic Systems*, vol. 107, no. 2, p. 29, 2023.

[15] S. Chopra *et al.*, "A Distributed Version of the Hungarian Method for Multirobot Assignment," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 932–947, 2017.

[16] R. E. Burkard *et al.*, "Assignment Problems," in *IFIP Congress: Fundamentals - Foundations of Computer Science*, 1998.

[17] G. Carpaneto and P. Toth, "Algorithm for the Solution of the Bottleneck Assignment Problem," *Computing*, vol. 27, no. 2, pp. 179–187, 1981.

[18] I. Jang *et al.*, "Anonymous Hedonic Game for Task Allocation in a Large-Scale Multiple Agent System," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1534–1548, 2018.

[19] G. Oh *et al.*, "PSO-based Optimal Task Allocation for Cooperative Timing Missions," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 314–319, 2016, 20th IFAC Symposium on Automatic Control in Aerospace 2016.

[20] H.-L. Choi *et al.*, "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[21] M. Turpin *et al.*, "CAPT: Concurrent Assignment and Planning of Trajectories for Multiple Robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014, publisher: SAGE Publications Ltd STM.

[22] S. Agarwal and S. Akella, "Simultaneous Optimization of Assignments and Goal Formations for Multiple Robots," in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 6708–6715.

[23] S. Kloder and S. Hutchinson, "Path Planning for Permutation-Invariant Multirobot Formations," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, 2006.

[24] B. Gravell and T. Summers, "Centralized Collision-free Polynomial Trajectories and Goal Assignment for Aerial Swarms," *Control Engineering Practice*, vol. 109, p. 104753, 2021.

[25] S. Akella, "Assignment Algorithms for Variable Robot Formations," in *Algorithmic Foundations of Robotics XII*, K. Goldberg *et al.*, Eds. Cham: Springer International Publishing, 2020, vol. 13, pp. 912–927.

[26] P. MacAlpine *et al.*, "SCRAM: Scalable Collision-avoiding Role Assignment with Minimal-Makespan for Formational Positioning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[27] H. Wang and M. Rubenstein, "Shape Formation in Homogeneous Swarms Using Local Task Swapping," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597–612, 2020.

[28] A. Khan *et al.*, "Learning Safe Unlabeled Multi-Robot Planning with Motion Constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7558–7565.

[29] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, 1982, vol. 32.

[30] R. Penicka and D. Scaramuzza, "Minimum-Time Quadrotor Waypoint Flight in Cluttered Environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5719–5726, 2022.

[31] X. Zhao *et al.*, "Fast Task Allocation for Heterogeneous Unmanned Aerial Vehicles through Reinforcement Learning," *Aerospace Science and Technology*, vol. 92, pp. 588–594, 2019.

[32] D. R. Fulkerson *et al.*, "A Production-Line Assignment Problem," RAND Corporation, Tech. Rep., 1953.

[33] G. A. Mills-Tettey *et al.*, "The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs," Robotics Insistute, Pittsburgh, Tech. Rep., 2007.

[34] P. C. Lusk *et al.*, "A Distributed Pipeline for Scalable, Deconflicted Formation Flying," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5213–5220, 2020.

[35] J. Alonso-Mora *et al.*, "Distributed Multi-Robot Formation Control in Dynamic Environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.

[36] L. Quan *et al.*, "Robust and Efficient Trajectory Planning for Formation Flight in Dense Environments," *IEEE Transactions on Robotics*, pp. 1–20, 2023, Early Access.

[37] M. Bürger *et al.*, "A Distributed Simplex Algorithm for Degenerate Linear Programs and Multi-agent Assignments," *Automatica*, vol. 48, no. 9, pp. 2298–2304, 2012.

[38] W. Honig *et al.*, "Trajectory Planning for Quadrotor Swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.

[39] D. Hert *et al.*, "MRS Drone: A Modular Platform for Real-world Deployment of Aerial Multi-robot Systems," *Journal of Intelligent & Robotic Systems*, vol. 108, pp. 1–34, 2023.

[40] D. Hert *et al.*, "MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments," in *2022 International Conference on Unmanned Aircraft Systems*, 2022, pp. 1264–1273.

[41] T. Baca *et al.*, "The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 2021.

Appendix A
Minimum distance on diagonals of an isosceles trapezoid

For finding the minimum mutual distance of robots following constant-velocity trajectories on the diagonals of an isosceles trapezoid, the relation for a minimum value of α^*_{ij} derived for a general case in (18) can be used. Given that $||\mathbf{s}_{ij}|| = ||\mathbf{g}_{ij}|| \Leftrightarrow a = c$, (18) can be simplified to

$$\alpha^*_{ij} = \frac{a-b}{2a-2b} = \frac{1}{2}. \tag{56}$$

As a consequence, the minimum distance between two trajectories following the diagonals of an isosceles trapezoid equals to distance between points cutting the diagonals in two line segments of equal length (see Fig. 11).

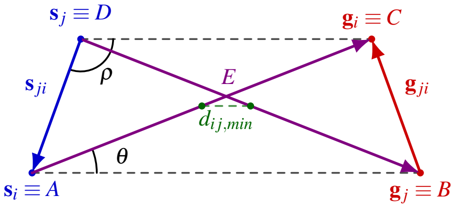


Fig. 11: Graphical illustration of the minimum distance $d_{ij,min}$ of constant-velocity trajectories lying on the diagonals of an isosceles trapezoid.

The theoretical guarantees on the minimum distance can be determined by the relation between minimum distance $d_{ij,min}$ and the value of δ_{ij} (20). Using substitutions

$$\mathbf{s}_i \equiv A, \mathbf{g}_j \equiv B, \mathbf{g}_i \equiv C, \mathbf{s}_j \equiv D, \tag{57}$$

the following relations hold for an isosceles trapezoid

$$K = \frac{|CD|}{|AB|} = \frac{|CE|}{|AE|} = \frac{|DE|}{|BE|}. \tag{58}$$

Applying the concept of triangle similarity, the distance $d_{ij,min}$ in an isosceles trapezoid can be expressed as

$$d_{ij,min} = \frac{|AE||AB| - \frac{1}{2}|AB||AC|}{|AE|}. \tag{59}$$

Based on the assumption from Theorem 1 and (58), we can determine relations

$$\begin{aligned} |AB| &= \frac{1}{M}|AC|, \\ |AE| &= \frac{1}{1+K}|AC|, \end{aligned} \tag{60}$$

that can be applied to simplify (59) to

$$d_{ij,min} = \frac{(1-K)|AC|}{2M}. \tag{61}$$

Furthermore, we can utilize the properties of the right triangle and leverage the Law of Cosines to find a system of equations

$$\begin{aligned} \cos\theta &= \frac{|AB| - \frac{|AB| - K|AB|}{2}}{M|AB|} = \frac{1+K}{2M}, \\ \cos\theta &= \frac{|AB|^2 + M^2|AB|^2 - \delta_{ij}^2}{2M|AB|^2}. \end{aligned} \tag{62}$$

From the system of equations (62), the relation for K can be derived as

$$K = \frac{M^2(|AC|^2 - \delta_{ij}^2)}{|AC|^2}. \tag{63}$$

By substituting this result into (61), we get the required relation for a minimum distance of two trajectories depending only on the minimum mutual distance of starts and goals δ_{ij}, the ratio between the length of the trajectories M, and the length of the trajectory $|AC|$

$$d_{ij,min} = \frac{|AC|^2(1-M^2) + M^2\delta_{ij}^2}{2M|AC|}. \tag{64}$$

To find a minimum distance of trajectories independently on the length of $|AC|$, we find the derivative of (64) with respect to $|AC|$ and set it equal to zero

$$\frac{\partial d_{ij,min}}{\partial |AC|} = \frac{(1-M^2)|AC|^2 - M^2\delta_{ij}^2}{2M|AC|^2} = 0. \tag{65}$$

As a solution of (65) with $\frac{\partial^2 d_{ij,min}}{\partial^2 |AC|} > 0$, we get the value of $|AC|$ minimizing the distance $d_{ij,min}$

$$|AC|^*_{min} = \frac{\sqrt{1-M^2}M\delta_{ij}}{1-M^2}. \tag{66}$$

By substituting $|AC|^*_{min}$ into (64) and simplifying this equation, we get a final relation for $d_{ij,min}$:

$$d_{ij,min} = \sqrt{1-M^2}\delta_{ij}, \tag{67}$$

that proves Theorem 1. □

Chapter 6

Results and Discussion

This chapter summarizes the achieved results and contributions presented in the core publications included in this thesis and other relevant author's publications. The summary is supplied by the discussion on achieved results and an outline of the potential future work. The chapter is organized into three sections corresponding to thematic sections established in the introduction of the thesis.

6.1 Autonomous navigation and exploration of GNSS-denied environments

The results achieved in single- and multi-UAV navigation and exploration of GNSS-denied environments represent significant advancement, establishing a new state of the art in the field. Together with solutions of the other teams participating in the SubT Challenge, the complete system presented in [1c], [2c], [6c] outperforms the existing solutions in terms of reliability, robustness, resiliency, and level of autonomy. These advances were achieved through designing a computationally-efficient software stack, including novel planning and exploration algorithms, and extensive field testing of the proposed system in a variety of GNSS-denied structures, each representing unique challenges such as extreme humidity, areas of water, extreme dust concentration, varying lighting conditions, and dynamic settings (see Figure 6.1). Such focus on the thus-far rather omitted aspects of GNSS-denied navigation and exploration has pushed the entire field to a new level, where the designed systems are close to achieving efficient deployment in real-world scenarios, such as post-disaster site reconnaissance or assistance to emergency services in search and rescue missions.

As an integral part of the heterogeneous multi-robot system of team CTU-CRAS-NORLAB [14a], the designed UAV-based system significantly contributed to the team's notable achievements in the DARPA SubT Challenge. In the systems track of the Urban Circuit, the system helped the team to secure first place among self-funded teams and third place overall. Further, within the systems track of the final round of the SubT Challenge, the heterogeneous CTU-CRAS-NORLAB robotic team achieved sixth place. In both these events, our UAV system was one of the top-performing UAV systems among all competing teams from leading universities and research institutions in the mobile robotics field. Within the virtual track of SubT Challenge the designed UAV-based exploration pipeline was a key component of the multi-robot team achieving second place in the final round of the competition. This makes the proposed system the top-performing system among those deployed in both the systems and virtual track of the competition.

The particular planning, mapping, and exploration algorithms developed as part of the designed navigation and exploration pipeline were integrated into the open source MRS UAV system[1] [83]. As such, the individual modules utilized during the competition are available for use in education and research activities of the MRS group and other research institutions. Together with the experience gained during the development of this software, these modules serve as the base for the most of the research projects of the MRS group[2] related to GNSS-denied navigation, including automated remote sensing in indoor environments [4c], [8a], exploration of multi-level buildings [134], and cooperative guidance of heterogeneous UAV teams in cluttered environments [19a], [135].

Despite recent advancements in S&R robotics, the effectiveness of single-UAV systems in general S&R scenarios is constrained by their limited payload capacity and operational time. Enhancing either of these aspects typically results in a trade-off, such as a reduction in sensory equipment or an increase in platform size, which in turn limits the operational space of the aerial platforms. In contrast to UGVs, the coupling between dimensions of the platform, operational time, and effective payload of multi-rotor UAVs is significant. Consequently, achieving compactness and collision tolerance often means sacrificing flight time and sensory equipment capabilities, and vice versa. This complex interplay of design factors highlights that there are still significant challenges to overcome in optimizing the balance between size, payload, and operational duration of UAVs in S&R robotics.

Two promising research directions are emerging to tackle these challenges. The first involves advancements in hardware technology and algorithms that could maintain high performance in localization, mapping, and navigation modules while relying on lighter sensors, such as RGBD cameras or radars, instead of heavy 3D LiDAR systems. The second approach resides in developing coordination algorithms for efficient cooperation within heterogeneous robotic teams featuring diverse specialized sensory setups, dimensions, endurance, and collision-tolerance characteristics to maximize the utility of individual robots for a specific role in the team. Although this coordination problem can be tackled as a distributed dynamic task assignment problem, it shows increased complexity compared to general problems of this type. This complexity arises from the need to consider factors like limited operational time and the range, reliability, and bandwidth of communication, all of which are affected by the robots' decisions. Moreover, in a heterogeneous robotic setup, the data required for effective coordination extend beyond merely sharing intentions and plans among individual robots. This makes the development of coordination algorithms for heterogeneous S&R robotic teams an intriguing area of research, with the potential to enhance the capabilities and applicability of UAVs in S&R operations.

6.2 Cooperative sensing by robotic teams

The research performed in the field of cooperative remote sensing by robotic teams [3c], [5c], [16a], [18a] represents a significant advancement beyond the current state of the art. While several research works tackle the remote sensing applications where a team of robots is required to simultaneously visit certain locations [31], [136]–[138], our work uniquely focuses on the cooperative enhancement of the captured scene — a concept that remains largely unexplored in existing literature. Thus, the results achieved in the cooperative sensing field outperform the state-of-the-art methods with respect to the level of cooperation and provide an important tool for monitoring the state of historical buildings and dynamic target tracking with additional lighting enhancing the captured scene.

[1] https://github.com/ctu-mrs/mrs_uav_system

[2] https://mrs.felk.cvut.cz/

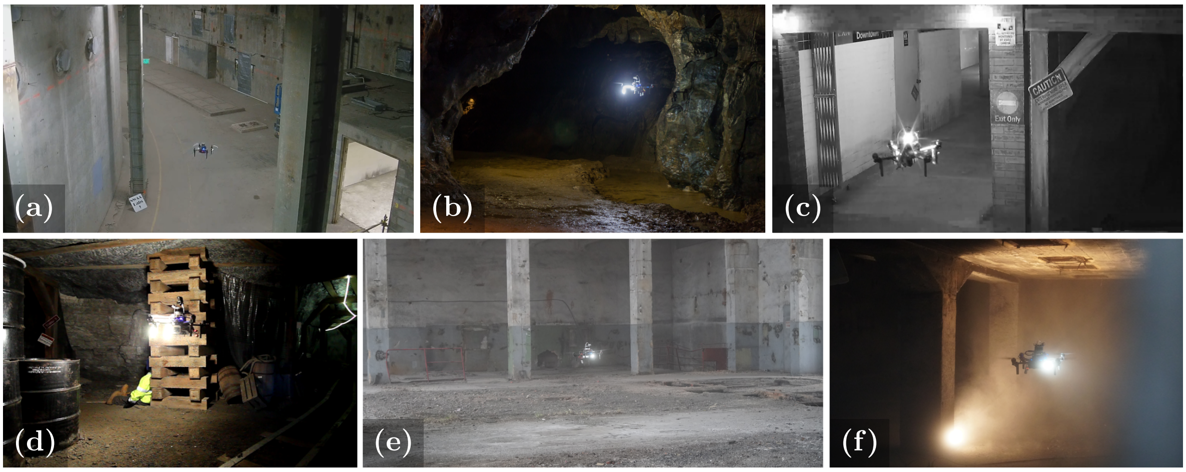


Figure 6.1: Deployment of the developed system for autonomous navigation and exploration of GNSS-denied environments, including an unfinished nuclear power plant during SubT Challenge Urban Circuit (a), abandoned buildings (e-f), large cave systems (b), and made-man structures in the DARPA SubT Finals (c-d).

The author's publications [4c], [8a] present a worldwide unique UAV-based technology for documentation of historical buildings that allows safe, repeatable, and frequent monitoring of historical monuments and particular valuable artifacts. The developed technology is based on application-tailored multi-rotor UAV equipped with a software stack integrating state-of-the-art methods of autonomous stabilization, localization, and navigation of cooperative UAVs inside large historical buildings. The presented framework, comprising additional software, tackles the whole documentation process, from obtaining an initial approximate model of the environment, specifying the required data to be captured, planning the missions, and automated data capturing, up to processing the data and providing outputs to the end-users. An integral part of the framework is the developed coordination algorithms for UAV-based cooperative sensing [3c], [16a] tailored for the realization of advanced documentation techniques in difficult-to-access areas of historical buildings. The high level of autonomy, the ability to fly beyond the visual line of sight between the UAV and a human operator, and the deployability in low lighting conditions (using a unique method of distributed illumination by a cooperating UAV team [5c], [16a]) enable gathering crucial data for heritage protection and documentation. Based on the high added value for heritage protection, the entire system has been approved by the Czech Ministry of Culture for indoor usage and accompanied by an official methodology describing its proper deployment in historical monuments. As the first methodology for indoor UAV use in historical buildings, it sets a new standard for such applications, offering a groundbreaking tool for cultural heritage documentation and preservation.

In cooperation with the Czech National Heritage Institute, the proposed technology for the realization of advanced documentation techniques was applied in several historical monuments, helping in the process of digitization of cultural heritage, collecting data for planned restoration works, and assessment of the state of individual artifacts to prevent their irreversible damage. So far, the system has been deployed in the documentation mission in 18 historical objects of varying dimensions and characteristics, including two historical structures listed in UNESCO World Heritage (Chateau Kroměříž and Chateau Telč), and six national heritage sites. During these documentation missions, the system was employed to perform a range of specific tasks tailored to the requirements of each site (images from several deployments are shown in Figure 6.2). This included photography in the visual, infrared, and ultraviolet spectra, enhanced by additional lighting provided by cooperating UAVs. Additionally, the Reflectance Transformation Imaging (RTI) method was used in these campaigns, further demonstrating the system's capability to accommodate various documentation techniques.

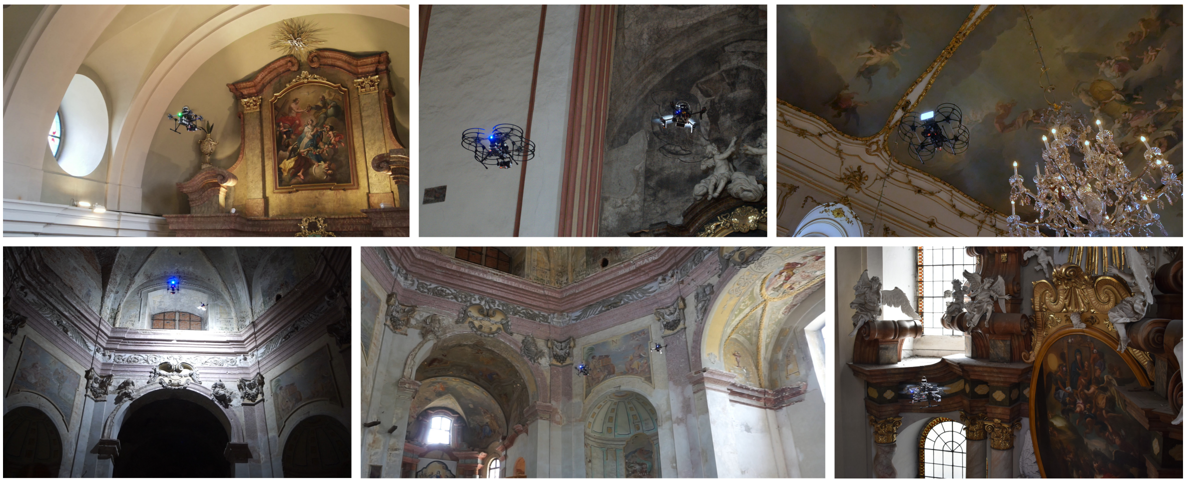


Figure 6.2: Example deployment of the developed framework for realization of documentation techniques, including cooperative sensing tasks, in interiors of historical buildings.

The cooperative motion planning approach was further improved in the core publication [5c], building upon the author's preliminary work on cooperative motion planning in cooperative sensing scenarios using Model Predictive Control (MPC) [16a]. This advancement was achieved by introducing an efficient multi-stage planning methodology based on convex decomposition of free space [139], enabling UAV formations to operate in more cluttered, complex, and dynamic environments. The proposed approach is further extended by including onboard detection and pose estimation of the human worker, as well as interaction with the human through onboard detected and processed gestures, as detailed in [18a]. To the best of our knowledge, this work represents the first instance of a contactless human-multi-robot interaction approach involving a human and a team of multi-rotor UAVs, incorporating onboard human state estimation and gesture recognition. The entire framework [5c], [18a] is part of a technology developed in the AERIAL-CORE[3] project, which is dedicated to the design of technology integrating aerial cognitive robotic systems with unprecedented capabilities in the operational range and safety. The project is particularly focused on the inspection and maintenance of large power transmission infrastructures, where interaction with people and aerial co-workers is highly beneficial.

The field of cooperative sensing offers several promising directions for future research. One key area involves miniaturizing hardware to enable reducing safe distances between robots and increasing the range of environments suitable for the deployment of developed cooperative sensing methods. Apart from miniaturizing hardware components, the potential improvements lie in implementing methods that decrease the requirements on the payload and heavy sensory equipment. For instance, a potential improvement is the direct application of the light-weight relative localization approach. This would further increase the reliability of the system through an additional means of localization [17a] and also provide an option to decrease the requirements on the sensory equipment of individual deployed UAVs [19a]. Advancements in this field

[3] https://aerial-core.eu/

can further increase the applicability of the developed framework thanks to lowering costs of the introduced technology.

A second promising research direction is the large-scale, long-term cooperative sensing missions that could greatly benefit from efficient solutions to task allocation and coalition formation problems. While these problems have been extensively studied and a variety of algorithms have been proposed [140]–[142], to our knowledge, there is no existing method that integrates path planning in general 3D environments directly into the task allocation process for multi-robot scenarios. The absence of such a method limits the practical application of existing algorithms in real-world scenarios, as decoupled sequential methods can lead to less efficient or even infeasible solutions due to the uncertain feasibility of paths between initial states and allocated goals. Addressing this problem would not only enhance the operational capabilities and efficiency of multi-robot systems, but also expand their potential use cases.

6.3 Formation shape adaptation

The core publication [7c] presents a centralized deterministic algorithm for time-optimal formation reshaping with collision avoidance guarantees, with it being the first publication presenting a solution to this problem. Despite the importance of time-efficient operation of robots in real-world applications, none of the approaches dealing with formation reshaping consider minimizing the makespan of the formation reshaping process while considering inter-agent collision avoidance in 3D environments [111]–[116], [118], [119]. While initially designed for mobile robot shape adaptation, the algorithm's potential applications extend beyond the realm of robotics. It can be applied to any situation requiring structural changes among entities at risk of colliding, making it a versatile tool for a variety of fields, such as analytical chemistry and biomedical diagnostics.

The low computational time of the proposed algorithm, together with the guarantees on the completeness and optimality of the solution, makes the algorithm suitable for real-world applications, including frequent onboard computation. We consider the efficient formation reshaping algorithm to be the fundamental element for extending cooperative sensing systems to large-scale scenarios comprising multiple cooperating formations with exchangeable formation members operating in shared environments. To pursue this goal, we plan to evolve the formation reshaping algorithm for use in environments with obstacles and to develop an efficient algorithm for formation splitting and merging, building upon the foundation provided by the proposed formation reshaping algorithm. These advancements have potential to further enhance the capabilities of cooperative robotic systems, enabling more complex, dynamic operations and opening up new possibilities for their deployment in a wide range of practical applications.

Chapter 7

Conclusion

This thesis addressed various aspects of cooperative sensing by a team of Unmanned Aerial Vehicles in complex, real-world environments. In the first part of the thesis, we focused on the development of a single-UAV system capable of reliable autonomous navigation in cluttered indoor environments comprising dynamic and challenging harsh real-world conditions. The developed UAV-based system, featuring novel planning and exploration algorithms, was extensively tested and successfully deployed in the DARPA SubT Challenge, where the developed system achieved one of the best performances among all UAV systems participating in the competition. The second part of the thesis is dedicated to the development of cooperative motion planning and multi-robot coordination algorithms for the realization of advanced documentation techniques and cooperative aerial filming tasks. These algorithms, integrated with an application-tailored hardware and software stack for autonomous indoor cooperative flight, are combined into a worldwide unique UAV-based technology for documentation of historical buildings, enabling safe, repeatable, and frequent monitoring of valuable historical structures. The entire framework has already been used in numerous documentation missions in historical monuments belonging to the most valuable historical structures in the Czech Republic, including UNESCO World Heritage sites. Lastly, the thesis addressed the problem of efficient structural reorganization of a group of mobile robots by presenting a time-optimal formation reshaping algorithm with completeness and inter-agent collision avoidance guarantees. The proposed algorithm, CAT-ORA, showcases superior performance in terms of minimizing the makespan of the formation reshaping process while maintaining computational demands at a level suitable for real-time deployment. Throughout the development of the technologies and algorithms presented in this thesis, a strong emphasis is placed on their practical application in real-world environments, underlining the goal of advancing the field of robotics in ways that are beneficial and meaningful for human society.

Chapter A

References

References to the author's work are listed first, followed by other references cited within this work. The authored references contain his contribution, the number of citations based on Web of Science (WoS), Scopus, and Google Scholar (GS), and the latest known impact factor of the journal at the time of initial submission of particular publications. The citation counts were gathered on January 9th, 2024.

A.1 Thesis core publications

Core articles in peer-reviewed journals with impact factor

[1c] **V. Krátký**, P. Petráček, T. Báča, and M. Saska, "An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments," *Journal of Field Robotics*, vol. 38, no. 8, pp. 1036–1058, 2021, **Contributions: VK: 55%, PP: 35%, TB: 5%, MS: 5%, IF 3.7 (Q2 in Robotics), citations: 17 in WoS, 15 in Scopus, 35 in GS**.

[2c] P. Petráček, **V. Krátký**, M. Petrlík, T. Báča, R. Kratochvíl, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021, **Contributions: PP: 42%, VK: 20%, MP: 20%, BT: 6%, RK: 6%, MS: 6%, IF 3.7 (Q2 in Robotics), citations: 24 in WoS, 26 in Scopus, 46 in GS**.

[3c] **V. Krátký**, P. Petráček, V. Spurný, and M. Saska, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020, **Contributions: VK: 60%, PP: 20%, VS: 10%, MS: 10%, IF 3.6 (Q1 in Robotics), citations: 15 in WoS, 18 in Scopus, 33 in GS**.

[4c] P. Petráček, **V. Krátký**, T. Báča, M. Petrlík, and M. Saska, "New era in cultural heritage preservation: Cooperative aerial autonomy for fast digitalization of difficult-to-access interiors of historical monuments," *IEEE Robotics & Automation Magazine*, pp. 2–19, 2023, **co-first author, Contributions: PP: 41%, VK: 41%, TB: 6%, MP: 6%, MS: 6%, IF 5.7 (Q1 in Automation and Control Systems), citations: 1 in WoS, 1 in Scopus, 6 in GS**.

[5c] **V. Krátký**, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, "Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021, **Contributions: VK: 38%, AA: 38%, JC: 7%, PS: 7%, MS: 5%, AO: 5%, IF 3.7 (Q2 in Robotics), citations: 16 in WoS, 23 in Scopus, 30 in GS**.

Core articles in peer-reviewed journals

[6c] M. Petrlík, P. Petráček, **V. Krátký**, T. Musil, Y. Stasinchuk, M. Vrba, T. Báča, D. Heřt, M. Pecka, T. Svoboda, and M. Saska, "UAVs beneath the surface: Cooperative autonomy for subterranean search and rescue in DARPA SubT," *Field Robotics*, vol. 3, no. 1, pp. 1–68, 2023, **Contributions: MP: 30%, PP: 15%, VK: 15%, TM: 10%, YS: 6%, MV: 6%, TB: 6%, DH: 6%, MP: 2%, TS: 2%, MS: 2%, citations: 0 in WoS, 0 in Scopus, 13 in GS**.

Core articles - submitted

[7c] **V. Krátký**, R. Pěnička, J. Horyna, P. Štibinger, T. Báča, M. Petrlík, P. Štěpán, and M. Saska, "CAT-ORA: Collision-aware time-optimal formation reshaping for efficient robot coordination in 3D environments," *submitted to Transactions on Robotics*, 2023, **Contributions: VK: 70%, RP: 12%, JH: 6%, PS: 4%, MP: 2%, TB: 2%, PS: 2%, MS: 2%, IF 7.8 (Q1 in Robotics), citations: 0 in WoS, 0 in Scopus, 0 in GS**.

A.2 Other thesis-related author's publications

Thesis-related articles in peer-reviewed journals with impact factor

[8a] **V. Krátký**, P. Petráček, T. Nascimento, M. Čadilová, M. Škobrtal, P. Stoudek, and M. Saska, "Safe documentation of historical monuments by an autonomous unmanned aerial vehicle," *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, p. 738, 2021, **30% contribution, IF 2.9 (Q2 in Computer Science, Information Systems), citations: 5 in WoS, 4 in Scopus, 10 in GS**.

[9a] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrlík, A. Reinke, **V. Krátký**, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, "Present and future of SLAM in extreme environments: The DARPA SubT Challenge," *IEEE Transactions on Robotics*, pp. 1–20, 2023, early access, **2% contribution, IF 7.8 (Q1 in Robotics), citations: 0 in WoS, 0 in Scopus, 48 in GS**.

[10a] D. Heřt, T. Báča, P. Petráček, **V. Krátký**, R. Pěnička, V. Spurný, M. Petrlík, M. Vrba, D. Žaitlík, P. Stoudek, V. Walter, P. Štěpán, J. Horyna, V. Pritzl, M. Šrámek, A. Ahmad, G. Silano, D. B. Licea, P. Štibinger, T. Nascimento, and M. Saska, "MRS drone: A modular platform for real-world deployment of aerial multi-robot systems," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 4, p. 64, 2023, **5% contribution, IF 3.1 (Q3 in Robotics), citations: 1 in WoS, 2 in Scopus, 6 in GS**.

[11a] P. Petráček, **V. Krátký**, and M. Saska, "Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2078–2085, 2020, **20% contribution, IF 3.6 (Q1 in Robotics), citations: 22 in WoS, 22 in Scopus, 43 in GS**.

[12a] J. Deckerová, J. Faigl, and **V. Krátký**, "Traveling salesman problem with neighborhoods on a sphere in reflectance transformation imaging scenarios," *Expert Systems with Applications*, vol. 198, p. 116 814, 2022, **20% contribution, IF 7.0 (Q1 in Computer Science, Artificial Intelligence), citations: 3 in WoS, 4 in Scopus, 6 in GS**.

[13a] M. Saska, D. Heřt, T. Báča, **V. Krátký**, and T. Nascimento, "Formation control of unmanned micro aerial vehicles for straitened environments," *Autonomous Robots*, vol. 44, no. 6, pp. 991–1008, 2020, **20% contribution, IF 3.6 (Q1 in Computer Science, Artificial Intelligence), citations: 9 in WoS, 15 in Scopus, 23 in GS**.

Thesis-related articles in peer-reviewed journals

[14a] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, T. Azayev, D. Heřt, M. Petrlík, T. Báča, V. Spurný, **V. Krátký**, P. Petráček, D. Baril, M. Vaidis, V. Kubelka, F. Pomerleau, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajník, "System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA Subterranean Challenge," *Field Robotics*, vol. 2, no. 1, pp. 1779–1818, 2022, **3% contribution, citations: 0 in WoS, 0 in Scopus, 36 in GS**.

Thesis-related conference proceedings in WoS

[15a] D. Heřt, T. Báča, P. Petráček, **V. Krátký**, V. Spurný, M. Petrlík, M. Vrba, D. Žaitlík, P. Stoudek, V. Walter, P. Štěpán, J. Horyna, V. Pritzl, G. Silano, D. Bonilla Licea, P. Štibinger, R. Pěnička, T. Nascimento, and M. Saska, "MRS modular UAV hardware platforms for supporting research in real-world outdoor and indoor environments," in *2022 International Conference on Unmanned Aircraft Systems*, 2022, pp. 1264–1273, **5% contribution, citations: 5 in WoS, 6 in Scopus, 25 in GS**.

[16a] M. Saska, **V. Krátký**, V. Spurný, and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation*, 2017, pp. 1–8, **25% contribution, citations: 9 in WoS, 21 in Scopus, 74 in GS, best paper award**.

[17a] J. Horyna, **V. Krátký**, E. Ferrante, and M. Saska, "Decentralized multi-robot velocity estimation for UAVs enhancing onboard camera-based velocity measurements," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 11 570–11 577, **10% contribution, citations: 1 in WoS, 1 in Scopus, 1 in GS**.

Thesis-related publications - submitted

[18a] **V. Krátký**, G. Silano, M. Vrba, C. Papaioannidis, I. Mademlis, R. Pěnička, I. Pitas, and M. Saska, "Gesture-controlled aerial robot formation for human-swarm interaction in safety monitoring applications," in *submitted to 2024 IEEE International Conference on Robotics and Automation*, 2024. [Online]. Available: `https://mrs.felk.cvut.cz/data/papers/icra2024gestures.pdf`, **40% contribution, citations: 0 in WoS, 0 in Scopus, 0 in GS**.

[19a] V. Pritzl, M. Vrba, Y. Stasinchuk, **V. Krátký**, J. Horyna, P. Štěpán, and M. Saska, *Drones guiding drones: Cooperative navigation of a less-equipped micro aerial vehicle in cluttered environments*, submitted to 2024 IEEE International Conference on Robotics and Automation, 2023. arXiv: `2312.09786 [cs.RO]`, **5% contribution, citations: 0 in WoS, 0 in Scopus, 0 in GS**.

[20a] A. Ollero, A. Suarez, C. Papaioannidis, I. Pitas, J. M. Marredo, V. Duong, E. Ebeid, **V. Krátký**, M. Saska, C. Hanoune, A. Afifi, A. Franchi, C. Vourtsis, D. Floreano, G. Vasiljevic, S. Bogdan, A. Caballero, F. Ruggiero, V. Lippiello, C. Matilla, G. Cioffi, D. Scaramuzza, J. R. Martinez-de-Dios, B. C. Arrue, C. Martin, K. Zurad, C. Gaitan, J. Rodriguez, A. Munoz, and A. Viguria, *AERIAL-CORE: AI-powered aerial robots for inspection and maintenance of electrical power infrastructures*, 2024. arXiv: `2401.02343 [cs.RO]`, **3% contribution, IF 5.7 (Q1 in Automation, and Control Systems), citations: 0 in WoS, 0 in Scopus, 0 in GS**.

A.3 Unrelated author's publications

Articles in peer-reviewed journals

[21a] M. Škobrtal, A. Škobrtal Zlámalová, **V. Krátký**, and P. Petráček, "Průzkum malby Zvěstování Panny Marie v Poutním chrámu ve Staré Vodě u Libavé pomocí bezpilotní helikoptéry," *Památky*, vol. 1, no. 4, pp. 16–33, 2022, **10% contribution, citations: 0 in WoS, 0 in Scopus, 0 in GS**.

Other submitted publications

[22a] J. Horyna, **V. Krátký**, V. Pritzl, T. Báča, E. Ferrante, and M. Saska, "Agile swarming of unmanned aerial vehicles in GNSS-denied plain environments without explicit communication," *submitted to IEEE Robotics and Automation Letters*, 2023. [Online]. Available: `https://mrs.felk.cvut.cz/data/papers/ral2023mrse_swarming.pdf`, **8% contribution, IF 5.2 (Q2 in Robotics), citations: 0 in WoS, 0 in Scopus, 0 in GS**.

A.4 Cited references

[23] M. Roberts, S. Shah, D. Dey, A. Truong, S. Sinha, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3D scanning," in *2017 IEEE International Conference on Computer Vision*, 2017, pp. 5334–5343.

[24] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1537–1548, 2020.

[25] Z. Zhang, Y. Zhong, J. Guo, Q. Wang, C. Xu, and F. Gao, "Auto filmer: Autonomous aerial videography under human interaction," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 784–791, 2023.

[26] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.

[27] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, "CERBERUS in the DARPA Subterranean Challenge," *Science Robotics*, vol. 7, no. 66, eabp9742, 2022.

[28] M. Petrlík, T. Báča, D. Heřt, M. Vrba, T. Krajník, and M. Saska, "A robust UAV system for operations in a constrained environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2169–2176, 2020.

[29] M. Vrba, D. Heřt, and M. Saska, "Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3402–3409, 2019.

[30] M. Vrba, Y. Stasinchuk, T. Báča, V. Spurný, M. Petrlík, D. Heřt, D. Žaitlík, and M. Saska, "Autonomous capture of agile flying objects using UAVs: The MBZIRC 2020 challenge," *Robotics and Autonomous Systems*, vol. 149, p. 103 970, 2022.

[31] P. Štibinger, T. Báča, and M. Saska, "Localization of ionizing radiation sources by cooperating micro aerial vehicles with pixel detectors in real-time," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3634–3641, 2020.

[32] P. Odonkor, Z. Ball, and S. Chowdhury, "Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping," *Swarm and Evolutionary Computation*, vol. 46, pp. 52–68, 2019.

[33] Y.-C. Lin and A. Habib, "Quality control and crop characterization framework for multi-temporal UAV LiDAR data over mechanized agricultural fields," *Remote Sensing of Environment*, vol. 256, p. 112 299, 2021.

[34] J. Xing, G. Cioffi, J. Hidalgo-Carrió, and D. Scaramuzza, "Autonomous power line inspection with drones via perception-aware MPC," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 1086–1093.

[35] C. Kanellakis, E. Fresk, S. S. Mansouri, D. Kominiak, and G. Nikolakopoulos, "Towards visual inspection of wind turbines: A case of visual data acquisition using autonomous aerial robots," *IEEE Access*, vol. 8, pp. 181 650–181 661, 2020.

[36] J. Horyna, T. Báča, and M. Saska, "Autonomous collaborative transport of a beam-type payload by a pair of multi-rotor helicopters," in *2021 International Conference on Unmanned Aircraft Systems*, 2021, pp. 1139–1147.

[37] H. Nguyen and K. Alexis, "Forceful aerial manipulation based on an aerial robotic chain: Hybrid modeling and control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3711–3719, 2021.

[38] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.

[39] M. Saska, V. Spurný, and V. Vonásek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, pp. 379–397, 2016.

[40] V. Hoang, M. Phung, T. Dinh, and Q. Ha, "Angle-encoded swarm optimization for UAV formation path planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 5239–5244.

[41] Y. Chen, J. Yu, X. Su, and G. Luo, "Path planning for multi-UAV formation," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 1, pp. 229–246, 2015.

[42] D. Zhou and M. Schwager, "Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles," in *2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 1737–1742.

[43] H. Mytum and J. R. Peterson, "The application of reflectance transformation imaging (RTI) in historical archaeology," *Historical Archaeology*, vol. 52, no. 2, pp. 489–503, 2018.

[44] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops*, 2019, pp. 835–840.

[45] M. Waibel, B. Keays, and F. Augugliaro, "Drone shows: Creative potential and best practices," ETH Zurich, Tech. Rep., 2017.

[46] S. J. Kim, Y. Jeong, S. Park, K. Ryu, and G. Oh, "A survey of drone use for entertainment and AVR (augmented and virtual reality)," *Augmented Reality and Virtual Reality: Empowering Human, Place and Business*, pp. 339–352, 2018.

[47] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.

[48] L. Quan, L. Yin, T. Zhang, M. Wang, R. Wang, S. Zhong, X. Zhou, Y. Cao, C. Xu, and F. Gao, "Robust and efficient trajectory planning for formation flight in dense environments," *IEEE Transactions on Robotics*, pp. 1–20, 2023, Early Access.

[49] I. Lopez-Sanchez and J. Moreno-Valenzuela, "PID control of quadrotor UAVs: A survey," *Annual Reviews in Control*, vol. 56, p. 100 900, 2023.

[50] R. Amin, L. Aijun, and S. Shamshirband, "A review of quadrotor UAV: Control methodologies and performance evaluation," *International Journal of Automation and Control*, vol. 10, no. 2, pp. 87–103, 2016.

[51] X. Ye, F. Song, Z. Zhang, and Q. Zeng, "A review of small UAV navigation system based on multisource sensor fusion," *IEEE Sensors Journal*, vol. 23, no. 17, pp. 18 926–18 948, 2023.

[52] L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of UAV motion planning," *IET Cyber-Systems and Robotics*, vol. 2, no. 1, pp. 14–21, 2020.

[53] A. Ait Saadi, A. Soukane, Y. Meraihi, A. Benmessaoud Gabis, S. Mirjalili, and A. Ramdane-Cherif, "UAV path planning using optimization approaches: A survey," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 4233–4284, 2022.

[54] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3D path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, pp. 1–22, 2016.

[55] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.

[56] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser], "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[57] H. Azpúrua, M. Saboia, G. M. Freitas, L. Clark, A.-a. Agha-mohammadi, G. Pessin, M. F. M. Campos, and D. G. Macharet, "A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments," *Robotics and Autonomous Systems*, vol. 160, p. 104 304, 2023.

[58] A. Sharma, P. Vanjani, N. Paliwal, C. M. Basnayaka, D. N. K. Jayakody, H.-C. Wang, and P. Muthuchidambaranathan, "Communication and networking technologies for UAVs: A survey," *Journal of Network and Computer Applications*, vol. 168, p. 102 739, 2020, ISSN: 1084-8045.

[59] N. Gyagenda, J. V. Hatilima, H. Roth, and V. Zhmud, "A review of GNSS-independent UAV navigation techniques," *Robotics and Autonomous Systems*, vol. 152, p. 104 069, 2022, ISSN: 0921-8890.

[60] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 9–15.

[61] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3D exploration," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4528–4535, 2021.

[62] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2135–2142.

[63] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[64] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

[65] L. Campos-Macías, R. Aldana-López, R. de la Guardia, J. I. Parra-Vilchis, and D. Gómez-Gutiérrez, "Autonomous navigation of MAVs in unknown cluttered environments," *Journal of Field Robotics*, vol. 38, no. 2, pp. 307–326, 2021.

[66] D. Duberg and P. Jensfelt, "UFOExplorer: Fast and scalable sampling-based exploration with a graph-based planning structure," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2487–2494, 2022.

[67] V. Orekhov and T. Chung, "The DARPA Subterranean Challenge: A synopsis of the circuits stage," *Field Robotics*, vol. 2, no. 1, pp. 735–747, 2022.

[68] T. H. Chung, V. Orekhov, and A. Maio, "Into the robotic depths: Analysis and insights from the DARPA Subterranean Challenge," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 477–502, 2023.

[69] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, S. Zimmermann, H. Nguyen, M. Fehr, L. Solanka, R. Buchanan, M. Bjelonic, N. Khedekar, M. Valceschini, F. Jenelten, M. Dharmadhikari, T. Homberger, P. D. Petris, L. Wellhausen, M. Kulkarni, T. Miki, S. Hirsch, M. Montenegro, C. Papachristos, F. Tresoldi, J. Carius, G. Valsecchi, J. Lee, K. Meyer, X. Wu, J. Nieto, A. Smith, M. Hutter, R. Siegwart, M. Mueller, M. Fallon, and K. Alexis, "CERBERUS: Autonomous legged and aerial robotic exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge," *Field Robotics*, vol. 2, no. 1, pp. 274–324, 2022.

[70] M. Tranzatto, M. Dharmadhikari, L. Bernreiter, M. Camurri, S. Khattak, F. Mascarich, P. Pfreundschuh, D. Wisth, S. Zimmermann, M. Kulkarni, V. Reijgwart, B. Casseau, T. Homberger, P. D. Petris, L. Ott, W. Tubby, G. Waibel, H. Nguyen, C. Cadena, R. Buchanan, L. Wellhausen, N. Khedekar, O. Andersson, L. Zhang, T. Miki, T. Dang, M. Mattamala, M. Montenegro, K. Meyer, X. Wu, A. Briod, M. Mueller, M. Fallon, R. Siegwart, M. Hutter, and K. Alexis, *Team CERBERUS wins the DARPA Subterranean Challenge: Technical overview and lessons learned*, 2022. arXiv: `2207.04914 [cs.RO]`.

[71] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, and K. Alexis, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation*, 2022, pp. 3306–3313.

[72] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems*, 2020, pp. 1024–1029.

[73] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. Lo Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernandez, G. Catt, W. Docherty, B. Tidd, B. Tam, S. Murrell, M. Bessell, L. Hanson, L. Tychsen-Smith, H. Suzuki, L. Overs, F. Kendoul, G. Wagner, D. Palmer, P. Milani, M. O'Brien, S. Jiang, S. Chen, and R. Arkin, "Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO Data61's approach to the DARPA Subterranean Challenge," *Field Robotics*, vol. 2, no. 1, pp. 595–636, 2022.

[74] J. Williams, S. Jiang, M. O'Brien, G. Wagner, E. Hernandez, M. Cox, A. Pitt, R. Arkin, and N. Hudson, "Online 3D frontier-based UGV and UAV exploration using direct point cloud visibility," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2020, pp. 263–270.

[75] M. T. Ohradzansky, E. R. Rush, D. G. Riley, A. B. Mills, S. Ahmad, S. McGuire, H. Biggie, K. Harlow, M. J. Miles, E. W. Frew, C. Heckman, and J. S. Humbert, "Multi-agent autonomy: Advancements and challenges in subterranean exploration," *Field Robotics*, vol. 2, no. 1, pp. 1068–1104, 2022.

[76] H. Biggie, E. Rush, D. Riley, S. Ahmad, M. Ohradzansky, K. Harlow, M. Miles, D. Torres, S. McGuire, E. Frew, C. Heckman, and J. Humbert, "Flexible supervised autonomy for exploration in subterranean environments," *Field Robotics*, vol. 3, no. 1, pp. 125–189, 2023.

[77] C. Cao, L. Nogueira, H. Zhu, J. Keller, G. Best, R. Garg, D. Kohanbash, J. Maier, S. Zhao, F. Yang, K. Cujic, R. Darnley, R. DeBortoli, B. Drozd, P. Sun, I. Higgins, S. Willits, G. Armstrong, J. Zhang, and S. Scherer, "Exploring the most sectors at the DARPA Subterranean Challenge Finals," *Field Robotics*, vol. 3, pp. 801–836, 2023.

[78] S. Scherer, V. Agrawal, G. Best, C. Cao, K. Cujic, R. Darnley, R. DeBortoli, E. Dexheimer, B. Drozd, R. Garg, I. Higgins, J. Keller, D. Kohanbash, L. Nogueira, R. Pradhan, M. Tatum, V. K. Viswanathan, S. Willits, S. Zhao, H. Zhu, D. Abad, T. Angert, G. Armstrong, R. Boirum, A. Dongare, M. Dworman, S. Hu, J. Jaekel, R. Ji, A. Lai, Y. H. Lee, A. Luong, J. Mangelson, J. Maier, J. Picard, K. Pluckter, A. Saba, M. Saroya, E. Scheide, N. Shoemaker-Trejo, J. Spisak, J. Teza, F. Yang, A. Wilson, H. Zhang, H. Choset, M. Kaess, A. Rowe, S. Singh, J. Zhang, G. A. Hollinger, and M. Travers, "Resilient and modular subterranean exploration with a team of roving and flying robots," *Field Robotics*, vol. 2, no. 1, pp. 678–734, 2022.

[79] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, “Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy,” *The International Journal of Robotics Research*, 2023, online first.

[80] J. Kuffner and S. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 995–1001.

[81] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund, M. F. Ginting, K. Ebadi, M. Anderson, T. Pailevanian, E. Terry, M. Wolf, A. Tagliabue, T. S. Vaquero, M. Palieri, S. Tepsuporn, Y. Chang, A. Kalantari, F. Chavez, B. Lopez, N. Funabiki, G. Miles, T. Touma, A. Buscicchio, J. Tordesillas, N. Alatur, J. Nash, W. Walsh, S. Jung, H. Lee, C. Kanellakis, J. Mayo, S. Harper, M. Kaufmann, A. Dixit, G. Correa, C. Lee, J. Gao, G. Merewether, J. Maldonado-Contreras, G. Salhotra, M. S. D. Silva, B. Ramtoula, Y. Kubo, S. Fakoorian, A. Hatteland, T. Kim, T. Bartlett, A. Stephens, L. Kim, C. Bergh, E. Heiden, T. Lew, A. Cauligi, T. Heywood, A. Kramer, H. A. Leopold, C. Choi, S. Daftry, O. Toupet, I. Wee, A. Thakur, M. Feras, G. Beltrame, G. Nikolakopoulos, D. Shim, L. Carlone, and J. Burdick, “NeBula: Quest for robotic autonomy in challenging environments; Team CoSTAR at the DARPA Subterranean Challenge,” *Field Robotics*, vol. 2, no. 1, pp. 1432–1506, 2022.

[82] A. Patel, B. Lindqvist, C. Kanellakis, A.-a. Agha-mohammadi, and G. Nikolakopoulos, “REF: A rapid exploration framework for deploying autonomous MAVs in unknown environments,” *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, p. 35, 2023.

[83] T. Báča, M. Petrlík, M. Vrba, V. Spurný, R. Pěnička, D. Heřt, and M. Saska, “The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 2021.

[84] T. Musil, M. Petrlik, and M. Saska, “SphereMap: Dynamic multi-layer graph structure for rapid safety-aware UAV planning,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 007–11 014, 2022.

[85] G. Pajares, “Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs),” *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–329, 2015.

[86] T.-Z. Xiang, G.-S. Xia, and L. Zhang, “Mini-unmanned aerial vehicle-based remote sensing: Techniques, applications, and prospects,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 3, pp. 29–63, 2019.

[87] P. Ramon-Soria, M. Perez-Jimenez, B. C. Arrue, and A. Ollero, “Planning system for integrated autonomous infrastructure inspection using UAVs,” in *2019 International Conference on Unmanned Aircraft Systems*, 2019, pp. 313–320.

[88] S. Zhuge, X. Xu, L. Zhong, S. Gan, B. Lin, X. Yang, and X. Zhang, “Noncontact deflection measurement for bridge through a multi-UAVs system,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 746–761, 2022.

[89] B. Yang, E. Yang, L. Yu, and C. Niu, “Ultrasonic- and IMU-based high-precision UAV localization for the low-cost autonomous inspection in oil and gas pressure vessels,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 523–10 534, 2023.

[90] G. Morgenthal and N. Hallermann, “Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures,” *Advances in Structural Engineering*, vol. 17, no. 3, pp. 289–302, 2014.

[91] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-Range Photogrammetry and 3D Imaging.* Berlin, Boston: De Gruyter, 2020.

[92] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri, and L. Seneviratne, “Coverage path planning with adaptive viewpoint sampling to construct 3D models of complex structures for the purpose of inspection,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 7047–7054.

[93] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 6423–6430.

[94] F. Nekovář, J. Faigl, and M. Saska, "Multi-tour set traveling salesman problem in planning power transmission line inspection," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6196–6203, 2021.

[95] D. Morilla-Cabello, L. Bartolomei, L. Teixeira, E. Montijano, and M. Chli, "Sweep-your-map: Efficient coverage planning for aerial teams in large-scale environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 810–10 817, 2022.

[96] N. Bolourian and A. Hammad, "LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection," *Automation in Construction*, vol. 117, p. 103 250, 2020.

[97] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. P. Ha, "System architecture for real-time surface inspection using multiple UAVs," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2925–2936, 2020.

[98] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A UAV system for inspection of industrial facilities," in *2013 IEEE Aerospace Conference*, 2013, pp. 1–8.

[99] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017.

[100] L. Yu, E. Yang, B. Yang, A. Loeliger, and Z. Fei, "Stereo vision-based autonomous navigation for oil and gas pressure vessel inspection using a low-cost UAV," in *2021 IEEE International Conference on Real-time Computing and Robotics*, 2021, pp. 1052–1057.

[101] M. Nieuwenhuisen, J. Quenzel, M. Beul, D. Droeschel, S. Houben, and S. Behnke, "ChimneySpector: Autonomous MAV-based indoor chimney inspection employing 3D laser localization and textured surface reconstruction," in *2017 International Conference on Unmanned Aircraft Systems*, 2017, pp. 278–285.

[102] X. Zheng, F. Wang, and Z. Li, "A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 483–494, 2018.

[103] M. Hu, W. Liu, J. Lu, R. Fu, K. Peng, X. Ma, and J. Liu, "On the joint design of routing and scheduling for vehicle-assisted multi-UAV inspection," *Future Generation Computer Systems*, vol. 94, pp. 214–223, 2019.

[104] E. Petitprez, F. Georges, N. Raballand, and S. Bertrand, "Deployment optimization of a fleet of drones for routine inspection of networks of linear infrastructures," in *2021 International Conference on Unmanned Aircraft Systems*, 2021, pp. 303–310.

[105] Y. H. Jo and S. Hong, "Three-dimensional digital documentation of cultural heritage site based on the convergence of terrestrial laser scanning and unmanned aerial vehicle photogrammetry," *ISPRS International Journal of Geo-Information*, vol. 8, no. 2, 2019.

[106] T. Bakirman, B. Bayram, B. Akpinar, M. F. Karabulut, O. C. Bayrak, A. Yigitoglu, and D. Z. Seker, "Implementation of ultra-light UAV systems for cultural heritage documentation," *Journal of Cultural Heritage*, vol. 44, pp. 174–184, 2020.

[107] N. Hallermann, G. Morgenthal, and V Rodehorst, "Vision-based monitoring of heritage monuments – unmanned aerial systems (UAS) for detailed inspection and high-accurate survey of structures," *WIT Transactions on The Built Environment*, vol. 153, pp. 621–632, 2015.

[108] M. Ceccarelli, D. Cafolla, G. Carbone, M. Russo, M. Cigola, L. J. Senatore, A. Gallozzi, R. Di Maccio, F. Ferrante, F. Bolici, S. Supino, N. Colella, M. Bianchi, C. Intrisano, G. Recinto, A. Micheli, D. Vistocco, M. R. Nuccio, and M. Porcelli, "Heritagebot service robot assisting in cultural heritage," in *2017 First IEEE International Conference on Robotic Computing*, 2017, pp. 440–445.

[109] B. P. Gerkey and M. J. Matarić, “A formal analysis and taxonomy of task allocation in multi-robot systems,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[110] S. Poudel and S. Moh, “Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey,” *Vehicular Communications*, vol. 35, p. 100 469, 2022.

[111] M. Turpin, N. Michael, and V. Kumar, “Capt: Concurrent assignment and planning of trajectories for multiple robots,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[112] S. Agarwal and S. Akella, “Simultaneous optimization of assignments and goal formations for multiple robots,” in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 6708–6715.

[113] S. Kloder and S. Hutchinson, “Path planning for permutation-invariant multirobot formations,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, 2006.

[114] S. Akella, “Assignment algorithms for variable robot formations,” in *Algorithmic Foundations of Robotics XII*, K. Goldberg, P. Abbeel, K. Bekris, and L. Miller, Eds., vol. 13, Springer International Publishing, 2020, pp. 912–927.

[115] P. MacAlpine, E. Price, and P. Stone, “SCRAM: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning,” *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, pp. 2096–2102, 2015.

[116] B. Gravell and T. Summers, “Centralized collision-free polynomial trajectories and goal assignment for aerial swarms,” *Control Engineering Practice*, vol. 109, p. 104 753, 2021.

[117] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982, ISBN: 0-13-152462-3.

[118] H. Wang and M. Rubenstein, “Shape formation in homogeneous swarms using local task swapping,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597–612, 2020.

[119] A. Khan, C. Zhang, S. Li, J. Wu, B. Schlotfeldt, S. Y. Tang, A. Ribeiro, O. Bastani, and V. Kumar, “Learning safe unlabeled multi-robot planning with motion constraints,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7558–7565.

[120] P. C. Lusk, X. Cai, S. Wadhwania, A. Paris, K. Fathian, and J. P. How, “A distributed pipeline for scalable, deconflicted formation flying,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5213–5220, 2020.

[121] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, “A distributed simplex algorithm for degenerate linear programs and multi-agent assignments,” *Automatica*, vol. 48, no. 9, pp. 2298–2304, 2012.

[122] H.-L. Choi, L. Brunet, and J. P. How, “Consensus-based decentralized auctions for robust task allocation,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[123] C. Ju, J. Kim, J. Seol, and H. I. Son, “A review on multirobot systems in agriculture,” *Computers and Electronics in Agriculture*, vol. 202, p. 107 336, 2022.

[124] G. Gardi, S. Ceron, W. Wang, K. Petersen, and M. Sitti, “Microrobot collectives with reconfigurable morphologies, behaviors, and functions,” *Nature Communications*, vol. 13, no. 1, p. 2239, 2022.

[125] D. Liu, T. Wang, and Y. Lu, “Untethered microrobots for active drug delivery: From rational design to clinical settings,” *Advanced Healthcare Materials*, vol. 11, no. 3, p. 2 102 253, 2022.

[126] V. Shekar, M. Campbell, and S. Akella, “Towards automated optoelectrowetting on dielectric devices for multi-axis droplet manipulation,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1439–1445.

[127] Y. Yu, J. Chen, and J. Zhou, “Parallel-plate lab-on-a-chip based on digital microfluidics for on-chip electrochemical analysis,” *Journal of Micromechanics and Microengineering*, vol. 24, no. 1, p. 015 020, 2013.

[128] E. Liu, C. Wang, H. Zheng, S. Song, A. Riaud, and J. Zhou, "Two-dimensional manipulation of droplets on a single-sided continuous optoelectrowetting digital microfluidic chip," *Sensors and Actuators B: Chemical*, vol. 368, p. 132 231, 2022.

[129] A Cosentino, "Macro photography for reflectance transformation imaging: A practical guide to the highlights method," *e-conservation Journal*, no. 1, pp. 70–85, 2013.

[130] J. Valcarcel Andrés and M. Osca Pons, "Applications of reflectance transformation imaging for documentation and surface analysis in conservation," *Internetional Journal of Conservation Science*, no. 4, pp. 535–548, 2013.

[131] Y. H. Kim, J. Choi, Y. Y. Lee, B. Ahmed, and K. H. Lee, "Reflectance transformation imaging method for large-scale objects," in *2016 13th International Conference on Computer Graphics, Imaging and Visualization*, 2016, pp. 84–87.

[132] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[133] J. R. Munkres, "Algorithms for the assignment and transportation problems," *Journal of The Society for Industrial and Applied Mathematics*, vol. 10, pp. 196–210, 1957.

[134] V. Pritzl, M. Vrba, C. Tortorici, R. Ashour, and M. Saska, "Adaptive estimation of UAV altitude in complex indoor environments using degraded and time-delayed measurements with time-varying uncertainties," *Robotics and Autonomous Systems*, vol. 160, p. 104 315, 2023.

[135] V. Pritzl, M. Vrba, P. Štěpán, and M. Saska, "Cooperative navigation and guidance of a micro-scale aerial vehicle by an accompanying UAV using 3D LiDAR relative localization," in *2022 International Conference on Unmanned Aircraft Systems*, 2022, pp. 526–535.

[136] X. Xu, G. Shi, P. Tokekar, and Y. Diaz-Mercado, "Interactive multi-robot aerial cinematography through hemispherical manifold coverage," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 11 528–11 534.

[137] A. Alcántara, J. Capitán, R. Cunha, and A. Ollero, "Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 140, p. 103 778, 2021.

[138] A. Rauniyar, J. Li, and S. Scherer, *Enhancing multi-drone coordination for filming group behaviours in dynamic environments*, 2023. arXiv: 2310.13184 [cs.RO].

[139] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[140] A. Rauniyar and P. K. Muhuri, "Multi-robot coalition formation problem: Task allocation with adaptive immigrants based genetic algorithms," in *2016 IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 137–142.

[141] H. Zhang, H. Luo, Z. Wang, Y. Liu, and Y. Liu, "Multi-robot cooperative task allocation with definite path-conflict-free handling," *IEEE Access*, vol. 7, pp. 138 495–138 511, 2019.

[142] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, "Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art," *Robotics and Autonomous Systems*, vol. 168, p. 104 492, 2023.

Chapter B

Citations of Author's Publications

Citations of the author's work were extracted from the Web of Science. First- and second-order self-citations are excluded. The data were gathered on January 9th, 2024.

[1c] **V. Krátký**, P. Petráček, T. Báča, and M. Saska, "An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments," *Journal of Field Robotics*, vol. 38, no. 8, pp. 1036–1058, 2021

- G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, "Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy," *International Journal of Robotics Research*, 2023.
- P. Trybala, J. Szrek, F. Remondino, P. Kujawa, J. Wodecki, J. Blachowski, and R. Zimroz, "MIN3D dataset: MultI-seNsor 3D mapping with an unmanned ground vehicle," *PFG-Journal of Photogrammetry Remote Sensing and Geoinformation Science*, 2023.
- J. Lee, J. Lim, S. Pyo, and J. Lee, "Aerial online mapping on-board system by real-time object detection for UGV path generation in unstructured outdoor environments," *Journal of Field Robotics*, vol. 40, no. 7, pp. 1754–1765, 2023.
- H. Azpurua, M. Saboia, G. M. Freitas, L. Clark, A.-a. Agha-mohammadi, G. Pessin, M. F. M. Campos, and D. G. Macharet, "A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments," *Robotics and Autonomous Systems*, vol. 160, 2023.
- Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, A.-a. Agha-mohammadi, and L. Carlone, "LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9175–9182, 2022.
- B. Lindqvist, C. Kanellakis, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "COMPRA: A compact reactive autonomy framework for subterranean MAV based search-and-rescue operations," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 3, 2022.
- R. Anand, H. M. B. Kumar, A. Raghavan, R. Maddara, and P. Anand, "Automated UAV to survey and monitor ionising radiation levels in a closed environment," *Power Electronics and Drives*, vol. 7, no. 1, pp. 134–145, 2022.
- G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, "Resilient multi-sensor exploration of multifarious environments with a team of aerial robots," in *Robotics: Science and Systems XVIII*.
- D. Zhu, T. Zhou, J. Lin, Y. Fang, and M. Q.-H. Meng, "Online state-time trajectory planning using timed-ESDF in highly dynamic environments," in *2022 IEEE International Conference on Robotics and Automation*, 2022, pp. 3949–3955.

[2c] P. Petráček, **V. Krátký**, M. Petrlík, T. Báča, R. Kratochvíl, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021

- H. Nguyen, R. Andersen, E. Boukas, and K. Alexis, "Uncertainty-aware visually-attentive navigation using deep neural networks," *International Journal of Robotics Research*, 2023.

- B. Zhou, W. Liu, and H. Yang, "Unmanned aerial vehicle service network design for urban," *Transportation Research Part C - Emerging Technologies*, vol. 157, 2023.
- P. Trybala, J. Szrek, F. Remondino, P. Kujawa, J. Wodecki, J. Blachowski, and R. Zimroz, "MIN3D dataset: MultI-seNsor 3D mapping with an unmanned ground vehicle," *PFG-Journal of Photogrammetry Remote Sensing and Geoinformation Science*, 2023.
- M. Chen, W. Yan, Y. Feng, S. Wang, and Q. Liang, "Large-scale underground mine positioning and mapping with LiDAR-based semantic intersection detection," *Mining Metallurgy & Exploration*, 2023.
- J. Yu, H. Shen, J. Xu, and T. Zhang, "ECHO: An efficient heuristic viewpoint determination method on frontier-based autonomous exploration for quadrotors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5047–5054, 2023.
- M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned aerial vehicles for search and rescue: A survey," *Remote Sensing*, vol. 15, no. 13, 2023.
- J. Yan, X. Lin, Z. Ren, S. Zhao, J. Yu, C. Cao, P. Yin, J. Zhang, and S. Scherer, "MUI-TARE: Cooperative multi-agent exploration with unknown initial position," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4299–4306, 2023.
- L. Long, G. Peiheng, Z. Yunlong, P. Qi, Y. Dequan, and T. Jiaying, "Research on UAV transmission line tracking and obstacle crossing technology based on RTK high-precision positioning," *Reviews of Adhesion and Adhesives*, vol. 11, no. 2, pp. 173–198, 2023.
- T. Gao, R. Weng, T. Wu, R. Zhang, C. Han, X. Ji, and M. Liu, "Near-ground trajectory planning for UAVs via multi-resolution hybrid voxel-surfel map," *Science China - Technological Sciences*, vol. 66, no. 5, SI, pp. 1245–1254, 2023.
- B. Zhou, H. Xu, and S. Shen, "RACER: Rapid collaborative exploration with a decentralized multi-UAV system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.
- L. Chen, Y. Liu, P. Dong, J. Liang, and A. Wang, "An intelligent navigation control approach for autonomous unmanned vehicles via deep learning-enhanced visual SLAM framework," *IEEE Access*, vol. 11, pp. 119 067–119 077, 2023.
- H. Azpurua, M. Saboia, G. M. Freitas, L. Clark, A.-a. Agha-mohammadi, G. Pessin, M. F. M. Campos, and D. G. Macharet, "A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments," *Robotics and Autonomous Systems*, vol. 160, 2023.
- T. Yu, B. Deng, J. Gui, X. Zhu, and W. Yao, "Efficient informative path planning via normalized utility in unknown environments exploration," *Sensors*, vol. 22, no. 21, 2022.
- M. Zhao, H. Lu, S. Cheng, S. Yang, and Y. Shi, "A multi-robot cooperative exploration algorithm considering working efficiency and working load," *Applied Soft Computing*, vol. 128, 2022.
- H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual-inertial-UWB state estimation system for aerial swarms," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.
- B. Lindqvist, S. Karlsson, A. Koval, I. Tevetzidis, J. Haluska, C. Kanellakis, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue," *Robotics and Autonomous Systems*, vol. 154, 2022.
- B. Bendris and J. Cayero Becerra, "Design and experimental evaluation of an aerial solution for visual inspection of tunnel-like infrastructures," *Remote Sensing*, vol. 14, no. 1, 2022.

[3c] **V. Krátký**, P. Petráček, V. Spurný, and M. Saska, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020

- Y. Castro, G. Le Goic, H. Chatoux, L. De Luca, and A. Mansouri, "A new pixel-wise data processing method for reflectance transformation imaging," *Visual Computer*, 2023.
- J. Deckerova, K. Kucerova, and J. Faigl, "On improvement heuristic to solutions of the close enough traveling salesman problem in environments with obstacles," in *2023 European Conference on Mobile Robots*, L Marques and I Markovic, Eds., 2023, pp. 180–185.

- A. Jaspe-Villanueva, M. Ahsan, R. Pintus, A. Giachetti, F. Marton, and E. Gobbetti, “Web-based exploration of annotated multi-layered relightable image models,” *ACM Journal on Computing and Cultural Heritage*, vol. 14, no. 2, SI, 2021.
- A. Alcantara, J. Capitan, R. Cunha, and A. Ollero, “Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles,” *Robotics and Autonomous Systems*, vol. 140, 2021.
- Y. Tang, L. Huang, J. Dai, and H. Huang, “Edge detection of reflectance transformation imaging for colorless traces,” in *AOPC 2021: Optical Sensing and Imaging Technology*, Y Jiang, Q Lv, D Liu, D Zhang, and B Xue, Eds., vol. 12065, 2021.

[5c] **V. Krátký**, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, “Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021

- W. Yu, B. Zhu, X. Wang, P. Yi, H. Liu, and T. Hu, “Enhanced affine formation maneuver control using historical velocity command (HVC),” *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7186–7193, 2023.
- G. J. Paz-Delgado, C. J. Perez-del Pulgar, M. Azkarate, F. Kirchner, and A. Garcia-Cerezo, “Multi-stage warm started optimal motion planning for over-actuated mobile platforms,” *Intelligent Service Robotics*, vol. 16, no. 3, pp. 247–263, 2023.
- S. Garrido, J. Munoz, B. Lopez, F. Quevedo, C. A. Monje, and L. Moreno, “Fast marching techniques for teaming UAV's applications in complex terrain,” *Drones*, vol. 7, no. 2, 2023.
- Z. Zhang, Y. Zhong, J. Guo, Q. Wang, C. Xu, and F. Gao, “Auto filmer: Autonomous aerial videography under human interaction,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 784–791, 2023.
- G. Borowik, M. Kozdon-Debecka, and S. Strzelecki, “Mutable observation used by television drone pilots: Efficiency of aerial filming regarding the quality of completed shots,” *Electronics*, vol. 11, no. 23, 2022.
- P. Pueyo, E. Montijano, A. C. Murillo, and M. Schwager, “CineMPC: Controlling camera intrinsics and extrinsics for autonomous cinematography,” in *2022 IEEE International Conference on Robotics and Automation*, 2022, pp. 4058–4064.
- X. Xu, G. Shi, P. Tokekar, and Y. Diaz-Mercado, “Interactive multi-robot aerial cinematography through hemispherical manifold coverage,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 11 528–11 535.
- M. Niarchos, M. E. Stamatiadou, C. Dimoulas, A. Veglis, and A. Symeonidis, “A semantic preprocessing framework for breaking news detection to support future drone journalism services,” *Future Internet*, vol. 14, no. 1, 2022.
- L. Sun, J. Chen, D. Feng, and M. Xing, “Parallel ensemble deep learning for real-time remote sensing video multi-target detection,” *Remote Sensing*, vol. 13, no. 21, 2021.

[11a] P. Petráček, **V. Krátký**, and M. Saska, “Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2078–2085, 2020

- B. Yang, E. Yang, L. Yu, and C. Niu, “Ultrasonic- and IMU-based high-precision UAV localization for the low-cost autonomous inspection in oil and gas pressure vessels,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 523–10 534, 2023.
- L. Kovanic, B. Topitzer, P. Petovsky, P. Blistan, M. B. Gergelova, and M. Blistanova, “Review of photogrammetric and lidar applications of UAV,” *Applied Sciences*, vol. 13, no. 11, 2023.
- N. Bu, J. Ge, J. Yang, and H. Ru, “Emergency landing system of rotor UAV in complex ground environment,” in *Proceedings of 2022 International Conference on Autonomous Unmanned Systems*, W Fu, M Gu, and Y Niu, Eds., ser. Lecture Notes in Electrical Engineering, vol. 1010, 2023, pp. 2954–2964.

- B. Yang, E. Yang, L. Yu, and C. Niu, "Adaptive extended kalman filter-based fusion approach for high-precision UAV positioning in extremely confined environments," *IEEE-ASME Transactions on Mechatronics*, vol. 28, no. 1, pp. 543–554, 2023.
- M. Jacquet, M. Kivits, H. Das, and A. Franchi, "Motor-level N-MPC for cooperative active perception with multiple heterogeneous UAVs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2063–2070, 2022.
- M. Jacquet and A. Franchi, "Enforcing vision-based localization using perception constrained N-MPC for multi-rotor aerial vehicles," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 1818–1824.
- G. Corsini, M. Jacquet, H. Das, A. Afifi, D. Sidobre, and A. Franchi, "Nonlinear model predictive control for human-robot handover with application to the aerial case," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 7597–7604.
- M. Niarchos, M. E. Stamatiadou, C. Dimoulas, A. Veglis, and A. Symeonidis, "A semantic preprocessing framework for breaking news detection to support future drone journalism services," *Future Internet*, vol. 14, no. 1, 2022.
- A. Alcantara, J. Capitan, A. Torres-Gonzalez, R. Cunha, and A. Ollero, "Autonomous execution of cinematographic shots with multiple drones," *IEEE Access*, vol. 8, pp. 201 300–201 316, 2020.

[8a] **V. Krátký**, P. Petráček, T. Nascimento, M. Čadilová, M. Škobrtal, P. Stoudek, and M. Saska, "Safe documentation of historical monuments by an autonomous unmanned aerial vehicle," *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, p. 738, 2021

- L. Kovanic, B. Topitzer, P. Petovsky, P. Blistan, M. B. Gergelova, and M. Blistanova, "Review of photogrammetric and lidar applications of UAV," *Applied Sciences*, vol. 13, no. 11, 2023.
- A. Alsayed and M. R. A. Nabawy, "Indoor stockpile reconstruction using drone-borne actuated single-point LiDARs," *Drones*, vol. 6, no. 12, 2022.

[12a] J. Deckerová, J. Faigl, and **V. Krátký**, "Traveling salesman problem with neighborhoods on a sphere in reflectance transformation imaging scenarios," *Expert Systems with Applications*, vol. 198, p. 116 814, 2022

- H. Huang, Y. Wei, Y. Zhou, and Q. Luo, "Spherical vector-based artificial gorilla troops optimization for spherical asymmetric multiple traveling salesman problem," *Evolving Systems*, 2023.
- P. Pirozmand, A. A. R. Hosseinabadi, M. J. Chari, F. Pahlavan, S. Mirkamali, G.-W. Weber, S. Nosheen, and A. Abraham, "D-PFA: A discrete metaheuristic method for solving traveling salesman problem using pathfinder algorithm," *IEEE Access*, vol. 11, pp. 106 544–106 566, 2023.
- M. Mondal and D. Srivastava, "A genetic algorithm-based approach to solve a new time-limited travelling salesman problem," *International Journal of Distributed Systems and Technologies*, vol. 14, no. 2, p. 15, 2023.

[13a] M. Saska, D. Heřt, T. Báča, **V. Krátký**, and T. Nascimento, "Formation control of unmanned micro aerial vehicles for straitened environments," *Autonomous Robots*, vol. 44, no. 6, pp. 991–1008, 2020

- Y. T. d. Passos, X. Duquesne, and L. S. Marcolino, "Congestion control algorithms for robotic swarms with a common target based on the throughput of the target area," *Robotics and Autonomous Systems*, vol. 159, 2023.
- H. Chehardoli and A. Ghasemi, "Adaptive size-independent control of uncertain leader following systems with only relative displacement information," *Physica A - Statistical Mechanics and Its Applications*, vol. 597, 2022.

- W. Li, X. Zhang, B. Huang, Y. Chen, R. Zhang, and S. BalaMurugan, "Research on the control method of unmanned helicopter under the background of artificial intelligence," *Journal of Interconnection Networks*, vol. 22, no. SUPP02, 2, 2022.
- S. Vargas, H. M. Becerra, and J.-B. Hayet, "MPC-based distributed formation control of multiple quadcopters with obstacle avoidance and connectivity maintenance," *Control Engineering Practice*, vol. 121, 2022.
- M. Nasir, M. F. Hayat, A. Jamal, and Z. Ahmed, "Frequency domain consensus control analysis of the networked multi-agent system with controller area network bus-induced delay," *Journal of Vibration and Control*, vol. 28, no. 19-20, pp. 2900–2912, 2022.

[16a] M. Saska, **V. Krátký**, V. Spurný, and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation*, 2017, pp. 1–8

- R. Maskeliunas, A. Katkevicius, D. Plonis, T. Sledevic, A. Meskenas, and R. Damasevicius, "Building facade style classification from UAV imagery using a pareto-optimized deep learning network," *Electronics*, vol. 11, no. 21, 2022.
- B. Sabetghadam, A. Alcantara, J. Capitan, R. Cunha, A. Ollero, and A. Pascoal, "Optimal trajectory planning for autonomous drone cinematography," in *2019 European Conference on Mobile Robots*, 2019.