CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

MASTER'S THESIS

Extinguishing of Indoor Fires by an Autonomous UAV

Vojtěch Nydrle

Thesis supervisor: **Ing. Pavel Petráček**

Department of Cybernetics

January 2023

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague on ....

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Nydrle Vojtěch** Personal ID number: **474408**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Extinguishing of Indoor Fires by an Autonomous UAV

Master’s thesis title in Czech:

Hašení požárů uvnitř budov pomocí autonomního UAV

Guidelines:

The aim of this thesis is to design and implement algorithms for an autonomous search of fires burning inside a multi-floor building and autonomous detection, localization, and extinguishment of such fires using an unmanned aerial vehicle (UAV) and sensors mounted onboard the UAV. The following tasks will be solved:
1. Design an algorithm and/or state machine for an autonomous 3D search of fires inside a multi-floor building by an outdoor-flying UAV using a thermal camera, and depth camera or 3D laser scanner [1-3].
2. Design and construct an application-tailored prototype of a UAV capable of carrying a heavy load (max. 20 kg) and suppressing dynamic recoil from launching capsules of fire extinguishant.
3. Design an algorithm for a temporal and spatial fusion of fire detections using onboard-UAV sensors.
4. Familiarize yourself with the system [4] developed by the Multi-Robot Systems (MRS) group for control of aerial vehicles and implement and integrate the developed algorithms within the simulation provided by the MRS group [5].
5. Perform a mocap exhibition of an autonomous 3D search of fires inside a multi-floor building in the simulation and analyze the performance of the designed solution.

Bibliography / sources:

[1] Che-Bin Liu and N. Ahuja, "Vision based fire detection," ICPR 2004, Cambridge.
[2] Cowlard, A., Jahn, W., Abecassis-Empis, C. et al. "Sensor Assisted Fire Fighting," Fire Technology, 2010.
[3] J.-H. Kim and B.Y. Lattimer, “Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot,” Fire Safety Journal, 2015.
[4] Tomas Baca, Matej Petrlik, Matous Vrba, Vojtech Spurny, Robert Penicka, Daniel Hert and Martin 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, 2021.
[5] MRS UAV System, online at: https://github.com/ctu-mrs.

Name and workplace of master’s thesis supervisor:

Ing. Pavel Petráček Multi-robot Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **29.08.2022** Deadline for master's thesis submission: **10.01.2023**

Assignment valid until: **19.02.2024**

Ing. Pavel Petráček Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Acknowledgments

I would like to thank my thesis adviser for his advice and improvements which were very helpful for this work and for his patience and goodwill. I would also like to thank all members of the MRS group for their help with experiments and useful advice, and the opportunity to do the project. Special thanks go to Ing. Daniel Hert who developed and assembled a power board used in the proposed UAV platform. Last thanks belong to my family for their support and motivation to finish the thesis.

Sincerely, thank you.

Abstract

This thesis with the deals design and implementation of algorithms for autonomous search, detection, localization, tracking and extinguishing of fires with a custom application-tailored UAV. Primary part of this work focuses on autonomous search of fires burning inside high-rise buildings with a UAV flying outdoors in close proximity to the building. The fires detected and localized during the search mission are tracked over time and reported to the groundyrt of this work focuses on autonomous search of fires burning inside high-rise buildings with a UAV flying outdoors in close proximity to the building. The fires detected and localized during the search mission are tracked over time and reported to the ground. The entire aerial system relies only on on-board sensors, namely thermal and RGB-D cameras and 3D LiDAR. The proposed fire-searching and fire-tracking algorithms are tested and analyzed during a simulated search mission. Apart from the software part, the thesis includes novel design, construction and preliminary testing of an application-tailored sensor-equipped heavy-payload UAV suitable for aerial discharge of a special fire extinguishing ampules. These ampules are launched from board of the UAV through the building window into a fire detected indoors.

Keywords: unmanned aerial vehicle, fire tracing, fire search, autonomous UAV, heavy payload multirotor, high-rise building fire search, semi-autonomous fire extinguishing, horizontal aerial discharge, high-rise building fire extinguishing

Abstrakt

Tato práce se zabývá návrhem a implementací algoritmů pro autonomní vyhledávání, detekci, lokalizaci, sledování a hašení požárů pomocí speciálně navržené a zkonstruované bezpilotní helikoptéry. Hlavní část této práce se soustředí na autonomní plánování průzkumné mise jejíž účelem je zmapování hořící budovy a zaznamenání pozic jednotlivých ložisek požáru s využitím dat z palubních senzorů helikoptéry. Hašení vnitřních ložisek požárů je zajištěno automatickým vystřelením ampule s hasící kapalinou z palubního vystřelovače skrze okno budovy do zdroje požáru. Za tímto účelem byla navržena a sestavena specializovaná bezpilotní helikoptéra vybavena všemi senzory potřebnými pro autonomní misi a optimalizována pro let s těžkým nákladem. Vyvinuté metody jsou analyzovány a jejich chování je testováno v simulaci.

Klíčová slova: bezpilotní helikoptéra, hledání ohně, sledování ohně, semi-automnomní hašení požárů, výstřel v letu, hašení požárů výškových budov, vyhledávání ohně ve výškových budovách, helikoptéra pro velké zatížení

Contents

List of Figures

List of Tables

Chapter 1: Introduction

Contents

This thesis deal with development of autonomous system capable of autonomous search and extinguishment of fires burning inside of multi-floor building. The task is to extinguish these fires from an outside-flying semi-autonomous UAV carrying ampules with a fire extuinguishant. Together with detecting and localizing the indoor fires, launching the ampules through the building windows into the fire are the primary objectives of the task. The key usage of the system lies in quick extinguishment of fires in high floors to reduce fire growing ratio or stop fire growth and provide necessary time for a team of firefighters to access the affected floor. The system requires an application-tailored UAV platform capable of horizontal discharge of fire extinguishing ampules into the fire through windows of the building. The UAV platform has to be equipped with on-board sensors and computational resources to allow autonomous UAV navigation, fire search, detection, and localization.

UAV is an aircraft without human operator onboard. The UAV can be operated remotely or can be self-controlled. A self-controlled UAV relies on onboard sensors and computer operating the UAV flight. Many kind of aircrafts exist, namely a plane, helicopter, multicopter, zeppelin and many other special types. An application of the UAV in the specified task implies operation in close proximity to high-rise buildings in limited space, which precludes usage of majority of aircraft types. The suitable kind for the specific task remains helicopter and multicopter for their ability to land and take-off vertically, hover at spot, and perform full-3D manoeuvres in limited space. A multicopter is used in this application for its robustness, smaller size, and simpler control, design, and construction.

Multicopters are used in countless applications such as filming, photography, monitoring and inspections [3–6], investigation dangerous areas [7, 8], entertainment, cargo transport [9–12], elimination of other aircraft [13], and even human transport. This huge application field creates variety of multicopters varying in size, shape, type of drive, and construction. Every multicopter consists of at least three motors with propellers forcing air under the aircraft, electronics for stabilization, platform for drive and electronics, and landing equipment.

Regarding the fire extuinguishants, an efficient solution is a Bonpet liquid [14]. Bonpet liquid is an aqueous solution of inorganic salts and organic compounds. Contact of the liquid with fire causes endothermic chemical reaction in the liquid producing inert gases (CO_2, N_2) which push oxygen out of the fire, thus removing the fuel for the fire and consequently stopping the process. The inert gases prevents fire to get oxygen to suffocate the fire. This liquid can be used to any class of fires. The plastic ampoules filled by the Bonpet liquid are designed for manual fire extinguish by throwing it into the fire. After impact, the plastic cover breaks and the liquid starts to evaporate and thus negating the reaction. The typical usage of these ampoules makes them ideal for aerial discharge into the fire through a window from the UAV. The primary objective of this thesis is to design and construct a UAV platform capable of carrying and launching six of the Bonpet-filled ampoules and to design and implement algorithms for autonomous search for indoor-burning fires and their detection and localization from on-board-UAV sensors.

1.1 Motivation

Firefighting in multi-floor building is very complicated and dangerous operation where every detail decides about safety of the human personnel. Each second when the fire grows brings more difficulties, damage to the building and other property, and can even result to loses in human lives. Therefore quick suppression of fires is fundamental. Suppression fire in the high floor is very complicated mission where firefighters need to traverse trough lower floors of the building or use ladders to access the fire from the outside. Both of these strategies are time consuming and dangerous when lower parts of the building are already damaged. Burning fire also expands quickly and endangers survivors confined in the building.

Exploration of a burning building takes part in beginning of the firefight and is extremely dangerous, the unknown part of building can be unstable and while its exploration can fall and hurt the exploring firefighter. There is a rising trend in using of using unmanned vehicles for an initial survey of a burning building from outside. These aerial vehicles allow for fast assessment of the fire. When equipped with a thermal imaging, the transmission of the video stream to the human personnel allows the fastest decision making than possible ever before. However, in these scenarios, the aerial vehicle is manually controlled and cannot suppress the fire directly. These times a multicopter equipped with thermal camera is frequently used for the exploration purpose but these multicopter is manually controlled and can not affect the fire and prevent it from growing. The manually controlled multicopter also needs specialized human personnel to operate it, which takes important time from the firefighting squad. Introducing assistive autonomy to the aerial surveying allows for speeding the task up by providing effective exploration and direct firefighting mechanisms. The autonomous exploration benefits in its self-control which does not require any human assistance and can provide the same information.The purpose of the UAV is to help firefighters in the first few minutes of their intervention. The idea is that the UAV is transported near the burning building and its mission is manually started, after the start of the mission, the UAV autonomously takes off and starts an autonomous mission.

The UAV mission can vary in the amount of assistance. It can be by fully autonomous, in this case the UAV is just manually started. After the start it automatically takes off and starts searching for fires. When some fire is found, it can be autonomously exhausted by launching Bonpet-filled ampules into it. This kind of mission has advantage in its independence on human assistance which saves important time for firefighters. But there is also disadvantage in the lack of any manual control because the UAV can behave sub-optimally missing some information human have. For example, exhausting the first visible or the largest fire may not be the primary target of exhaustion. In some cases, it might be best to stop spreading of the fire whereas in others it might be beneficial to exhaust fires producing hazardous substances into the air. The decision still remains for the human supervision, which may consider many qualitative factors into the process. These factors may include the danger of the fire based on its location in the building, proximity to hazardeous materials, location of people inside the building, and many others.

More assisted mission can also start with autonomous takes off and fire searching. However, when a fire is found, the UAV stops and notifies the operator who decides whether to extinguish this fire or continue in the search. Such a scenario benefits from the human decision but if the operator is busy by other important tasks, the UAV will keep hovering and wasting important time and energy. For these reasons, a compromise between the approaches is proposed. In this scenario, the UAV autonomously searches for fires and creates 3D map of the environment. In this map, each detected fire is marked and its photo is stored in a database. When the operator needs information about state of the fires and the building, he can just view the map. The operator can also command the UAV to extinguish fires selected in the map or add indices where he wishes the UAV to continue with searching of the fires. This allow the operator to prioritize searching in specified parts of the building. Important property of such an approach is that the UAV can detect and mark other object of interest in the map like survivors [15] during the search.

In both cases, mission initialization (UAV takeoff) and fire searching is autonomous. The autonomous search can proceed to the following scenario. The UAV is manually placed at location suitable for take off, where the building of interest is visible in the onboard sensory data. The building which needs to be explored is identified in the map. Identification of the right building is based on the distance to the building, because it can be assumed (or eventually ensured by the firefighters) that the UAV will start in that place that is closer to the burning building than to the other buildings. After the building of interest is identified, the UAV takes off and approaches the building. After the building is approached, the search for fires begins. Unless specified differently, the search starts from the upper floors. This is because human firefighters start their intervention from the ground floors and follow up. This maximizes the overall coverage during the task. Once the UAV reaches top of the building, it starts generating an optimal search plan. In the proposed solution, this problem is defined as maximum coverage problem in which the entire exterior surface of the perceived building is to be surveyed in minimal time. To do so, the proposed solution defines a set of viewpoints under which the building should be captured with the on-board sensors. This perception-constrained task however requires first to parametrize the building surface. To obtain such a representation, the building is splitted into planar surfaces (walls) which are segmented out from the environment map. Plane-segment segmentation results to a set of walls to be inspected. For each of the wall, a set of viewpoints is generated such that each wall is fully covered if all the viewpoints are visited. Given a set of all viewpoints, the optimal (shortest) path visiting all the viewpoints is computed. This problem is defined as TSP, which in the proposed solution is solved via LKH heuristic. While the UAV is following the plan, new parts of the wall can be explored. Because of that, the entire procedure has to be dynamic and thus be capable of generating VPs, plan path, and follow the new plans online during the flight. After completing inspection of the wall new wall searched for or indices from the operator may be visited.

Simultaneously to the fire search, all the previously detected fires are stored in the map and their tracking over time and space should be performed. During the mission, the UAV shall accept commands from the operator: for extinguishing fires, adding indices or aborting the mission and returning to base. When a command to extinguish a fire is given, the UAV should interrupt the search, navigate to the given fire, position itself optimally such that the launched ampoule will reach the fire over the known ballistic curve, and fire the ampoule into the fire. Once a fire is successfully extinguished, the mission may continue.

1.2 Related work

UAVs are often used in a number of various applications but mainly in monitoring searching and inspections. But in the last years raises the amount of research in the field of application of UAV in firefighting. Most of the research deals with monitoring huge forest fires [16] and searching for and detecting them [17–24], but also UAVs firefighting them are developed [25]. Another fire fighting UAV for exhausting wildfires drops balls filled with fire extinguishant into the fire [26]. Manually controlled UAVs are also often used in firefight missions to provide information about fires and survivors to firefighters [27]. This practice was also upgraded to the system which allows the operation of the monitoring drone from remote dispatching [28] this technique can produce the same information but without busying one of the firefighters.

An autonomous system for planning of whole UAV inspection mission was designed [29], this system brings the ability to plan a path through a known city map to the interesting cubed object. Generate a viewpoint in which is the cube inspected and path through these viewpoints and to the goal destination. Another paper discusses scenarios using UAV for fire fighting [30, 31] describing mostly the strategy of the UAV-fire flight on place and cooperation between firefighters and a team of cooperating UAVs used in this scenario.

Autonomous system for searching various object or survivors by one or group of UAVs were also discussed [7, 8]. Many searching algorithms and mission planning was discussed [32–35] solving its optionally robustness and efficiency.

Also collision-free path planning for heavy Firefighting UAVs was studied and discussed [36]. This is necessary because heavy-loaded UAV has slower dynamics and can not perform aggressive maneuvers without extreme energy consumption which will reduce the time of flight.

Mohamed Bin Zayed International Robotics Challenge (MBZIRC 2020) brings focus to the area of direct UAV-fire flight, which results in several designs capable of fire extinguishment in specific conditions [37–39]. Also another firefighting UAVs were developed for autonomous fire elimination in high-rise buildings [spray,sc,bp] these UAVs are able to autonomously detect fire and position its nozzle towards the found fire to hit it by extinguishant.

Chapter 2: Preliminaries

Contents

This thesis deal with the problem of autonomous extinguishing of indoor fires by UAV flying outside the building. The fires are extinguished by horizontal discharge of a BonPet-filled (fire extinguishant) ampule from the UAV into the fire through the building window. Purpose of the UAV is to found the fires, provide information about its positions to the firefighters, and extinguish some of the fires with minimal human assistance. Intended UAV behavior is generation of map with highlighted fire positions while autonomous search in close proximity of the burning building. Fire marked in the map can be selected as target by human operator and The UAV will autonomously navigate to the selected fire and discharge one or more ampule into it.

The presented problem is quite complex and brings challenges in many domains, therefore it is decoupled into several tasks that can be solved independently. These separated parts can be designed independently, the parts to be designed are namely the UAV platform, fire search procedure, fire tracking system, system for navigation the UAV near the fire, system to position the UAV and launch the ampule into the fire, high-level control of the mission, low-level system for UAV stabilization and localization, mapping and planning system, and some human interface for viewing the generated map of fires and commanding the UAV. As each of this design can be divided into many subtasks which need to be solved it is not possible to design and describe full system in one thesis. Therefore this thesis aims to design just the UAV platform, fire search procedure, and fire tracking system. Even these system designed in this thesis can be divides into subsystems and many of them relays on systems already designed. These already designed subsystem are described bellow in this chapter, following chapters describes systems designed in this thesis.

2.1 MRS system for autonomous UAV control

MRS system for autonomous UAV control [40, 41] takes care of UAV state estimation and control. It localizes the UAV in multiple coordination frames based on available sensors and localization methods. It also provides a fusion of these localization methods based on the credibility of each method. This feature allows to use of the UAV across multiple environments and relies on the localization method that works in the particular environment. For example, it allows a transition from free space outside where only GNSS-based localization gives reasonable position estimation to an indoor environment where no GNSS is available and therefore some LiDAR-based or vision-based localization is used. This multi sensors localization is necessary for the desired application of the UAV due to a GNSS-unstable environment in close proximity to buildings, trees, and other objects in low altitudes contrasting to the top floors of tall buildings with a flat shiny surface which may causes failures of LiDAR-based or vision-based localization methods.

The system also contains a number of sub-modules solving common problems like mutual localization of multiple UAV platforms or their communication. Another module provides mapping and planning support for UAVs equipped with LiDAR or depth camera, this module provides many tools to create and manage octomap-based maps. Octomap is memory friendly way of storing voxel representation of ambient environment closely described in [42]. The voxel-based map represents the world as a grid where each cell can store one of three states. Each cell can be free, occupied, or unknown. The free cell is cells that can be traversed by the UAV without collision, occupied cells represent obstacles that can not be passed and unknown cells are cells out of range of the sensor used for mapping, cells that can not be seen because are blocked by other cells or cells where can not be clearly determined whatever is occupied or free. This representation of the ambient environment can be used for path and mission planning as well as for detection of a building, its walls, and places needed to be inspected by a thermal camera.

Many approaches for planning a path between two or more points in a voxel map exist. Between commonly used approaches belongs the A* algorithm which provides good performance in the 3D grid. The panning module also integrates the A* algorithm to provide a planner which can generate a path from the UAV position to the desired goal. If the goal is out of explored map the planer generates sub-paths that lead the UAV to the edge of the explored map and when the UAV scans a new part of the map the planner generates a new part of the path through this newly scanned map until it reached the goal.

2.2 Fire detection system

The proposed fire searching system relies on a fire detector developed in [43]. This detector uses data from a thermal camera. The camera supplies images containing temperature encoded in each pixel. In these thermal images, fire is highly contrasting with the background. Therefore a thresholding-based method can be applied here in combination with a flood fill algorithm to detect multiple fires in one image. This method provides 2D fire positions in the coordination frame of the thermal images with brief descriptions of the fire, like its maximal temperature and pixel count. Figure 2.1 shows thermal image with highlighted fires.

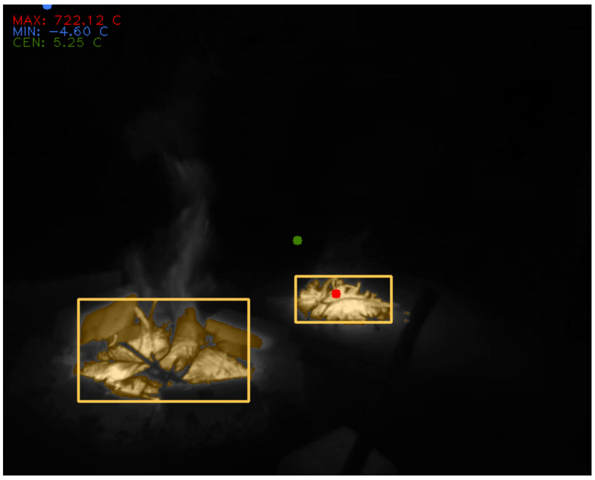


Figure 2.1: Thermal image with highlighted detected fires showing two separate fires and their detected bounding boxes.

2.3 Fire localization system

The localization system developed in [43] is used for the estimation of a 3D fire position from 2D indices given by the fire detector. The 3D fire position is used by the fire tracking system. The localization system relays on data from a depth camera (point cloud) which is used for the determination of the scene depth in the direction defined by the 2D indices. This process assumes the mutual camera transformations are known. The output of this method presents a 3D position of each detected fire in UAV coordination system. Fire located in point cloud is shown in Figure 2.2.

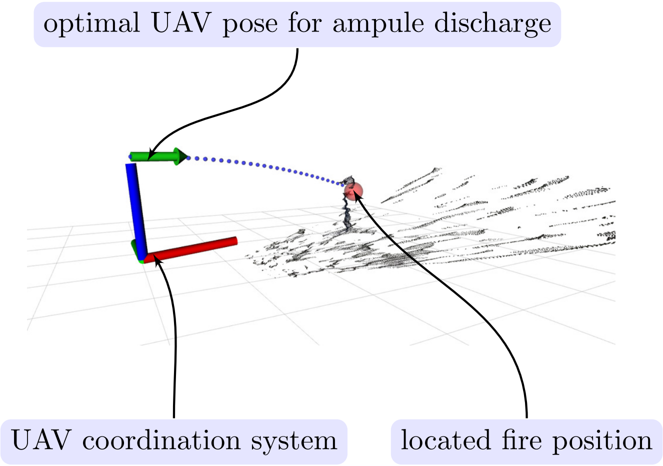


Figure 2.2: Fire location in point cloud and optimal UAV pose for ampule discharge.

2.4 Fire extinguishing ampule launching

In [43] also system for positioning the UAV towards fire in such a way that extinguishing ampule discharged from the UAV falls into the fire was designed. The optimal pose of the UAV for ampule discharge into the fire is shown in Figure 2.2. The designed system assumes the fire is present in the field of view of onboard sensors during the positioning. This cannot be ensured in real world applications. Therefore this system is enchanted by a fire tracking system that keeps track of all seen fires. This fire tracking system stores all fires position with information about the fire, which can be used for positioning the UAV even when direct UAV-fire visibility is lost.

Chapter 3: Hardware Design

Contents

One fire-exhausting capsule carried by the UAV platform previously designed in [43] is not enough to cover a burning surface of a medium-sized fire. This leads to the necessity to supply more capsules to the fire to provide capabilities of extinguishing larger fires. We propose to challenge this issue with a new, larger launcher capable of carrying six fire-exhausting capsules and launch each one independently on the others. This new launcher is capable to carry six fire-exhausting capsules and launch each one independently on the others. This launcher, designed bu an external supplier, weights 20 kg and has dimensions of 37×65×16 cm. The weight and size of the launcher require a new UAV platform capable of carrying this launcher with capsule payload.

The design of the UAV platform can be divided into two parts: the design of mechanical parts, especially a frame and base component, and the electrical design of a drive, sensory equipment, onboard computer, power delivery, and other onboard electronics.

3.1 Platform specification

The applications-specific UAV needs to be capable of autonomous search and extinguishing of fires in close proximity to a tall building (subject to fire extinguishing). This building can be located in free space, in a dense-built city, or nearby a forest. All these environments are expected to contain a variety of different obstacles reducing traversability and blocking transmission of the GNSS signal. Both these issues can lead to difficulties in UAV localization. Moreover, such a building is expected to have simple flat wall which will reduce the usability of sensor-based localization, both laser and vision-based. Therefore the UAV should be equipped with both of these complementary localization systems. The presented hardware design integrates all necessary sensory components allowing to achieve robust localization in these environments, but the localization itself is out of scope of this thesis and is assumed to be provided.

The platform aims to be used in the first part of a firefight intervention before firefighters access the fire. Firefighting intervention has to be fast because the whole apartment can burn down in a few minutes. Therefore the designed UAV must interpose fast but does not need to fly a long time. Five minutes of flight time is sufficient for this purpose. Longer flight time will be an advantage but will bring heavier batteries and all additive mass of the UAV will increase its size.

For safe usage in the outdoor environment near the firefighting intervention, it is necessary to ensure water resistance at least at IP52 level to safely cope with rain or falling reflected water drops coming from other firefighting methods. It is not important to ensure complete water resistance which will bring huge construction difficulties and not protect the UAV in case of a direct hit of water stream from firefight sprinklers which will completely destabilize the UAV and cause a crash anyway.

The extreme weight of the launcher forces huge thrust of the UAV drive to even hover airborne. To enable the UAV to perform faster maneuvers or cope with the kickback from launching the fire-extinguihsing capsules, it is necessary to provide a bigger thrust than the UAV mass. By a common rule of thumb, a UAV having twice bigger thrust than mass is well controllable and able for all necessary maneuvers needed for autonomous flight. The high thrust is ensured by employing propellers with a responding radius, which increase the size of the UAV. Large UAV size reduces its traversability through the environment and the size of the UAV is also limited by the ability to transport it by a human personnel. Therefore it is required to keep the UAV as light as it is possible.

The design aims to propose a UAV that compromises between these qualities:

- **small size** — to enable transportation and stowage and to increase traversability through any kind of environment,
- **high thrust** — to ensure the ability of stable and controllable flight, and the capability to handle dynamic recoil caused by the mechanical launching of the capsules,
- **light weight** — to allow longer flight time and smaller size,
- **flight time longer than five minutes** — to provide a reasonable time for assisted operation,
- **water resistant** — to be used in rain or cope with reflected water drops caused by other used fire fighting methods,
- **sensor equipped** — to carry sensors for fire detection and UAV localization and stabilization.

The design of each component with respect to these qualities is described in the following chapter.

3.2 Materials

To ensure robustness and light weight, the entire frame as well as the base component is made from carbon fiber composite. Carbon fiber composite is an ideal material for aerial vehicles for its low density and high toughness. On the other hand it is difficult to create a component from it. Process of creating carbon components include forming of carbon mesh on a form-shaped model. This mesh is then pressed on the model under a high temperature [44]. This process is expensive and difficult therefore our design relies only on parts milled and sawn from semi-products of the carbon fiber composite, which are comercially available, such as plates, pipes, and tubes. The whole frame and the base component is formed by carbon tubes connected together by planar junctions milled from a carbon plate. One of these junctions is shown in Figure 3.1.

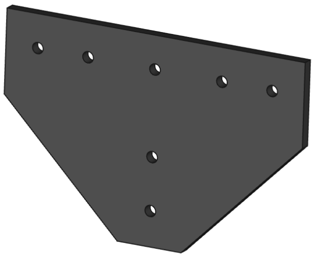


Figure 3.1: CAD model of planar carbon fiber junction, milled from 3 mm thick plate.

Connecting the junctions as well as other components to the carbon tubing in the middle of the tube brings some challenges. Screws and nuts can not be easily used because there is no access to the nut or screw inside the tube. Alternative to the screw and nut connection is the usage of blind rivets, which can be easily inserted even from one accessible side. A drawback of this solution is that this connection can not be disassembled in case of repair or improvement of the UAV prototype. Another problem of the blind rivets is the small contact area with the carbon composite on the blind side of the junction. This small area will lead to big pressure on the carbon composite leading to damage to the surface, which can possibly lead to pulling out the rivet from the hole in the carbon composite. Therefore this solution was rejected and a special nut holder was designed to allow screwing screws to nuts inside the tube with access to only the outer side.

These nut holders were printed on 3D printers from PET-G plastic, equipped with nuts, and pushed to the right place in the tube. Then screw can be screwed to the nut through a previously drilled hole in the tube. This approach allows for assembling the frame but has disadvantage in very complicated assembling.

3.3 Frame

The main purpose of the frame is to provide a connected platform to mount motors, a base component, and landing gear on it. There are many kinds of multi-rotor UAV frames. The most common is a quad-rotor, with 4 rotors placed on extreme points of a square. But for this high-weight UAV application, 4 rotors do not provide sufficient thrust, therefore the designed UAV is based on octocopter X8. This X8 configurations has 8 motors in coaxial configuration placed on extreme points of the rectangle, which provide higher thrust without enlargement of the frame. This selected rotor configuration is shown in Figure 3.2.

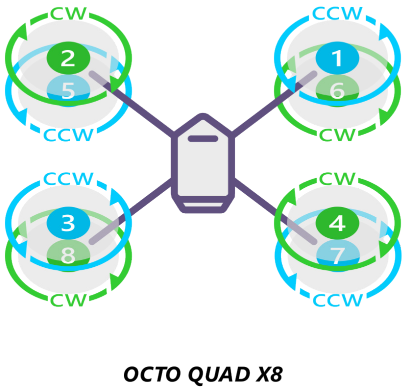


Figure 3.2: Selected rotor placement in X8 configuration [1].

These four extreme points of the rectangle can be connected to the body component by four direct arms or all the corners can be connected together by a rectangular frame holding the base component in the middle of its longer edge. In the proposed design, a rigid rectangular frame holds the base to which a landing gear is attached. This frame is formed by four carbon fiber composite rectangular tubes connected together by planar junctions in corners. These junctions also hold coaxial rotors in their places. The base component is connected to the middle of longer tubes by interlocks and planar junctions. Figure 3.3 shows the whole frame with motors and landing gear.

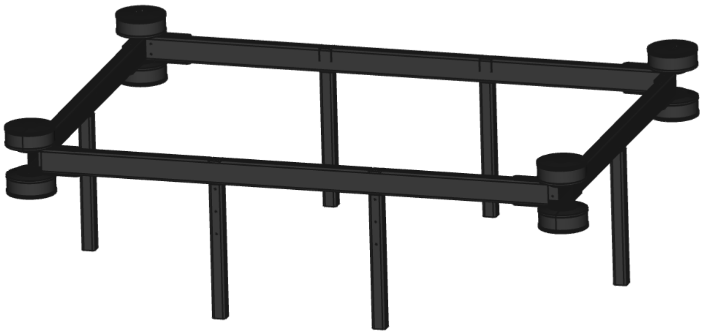


Figure 3.3: Visualization of the main frame with landing gear.

The frame dimensions are 119 × 83 cm and is formed from four 1” by 2” rectangular tubes. Two of these tubes hold the base component which weighs 34 kg (the launcher, batteries, ESCs, and others). The mass of the base component creates bending stress in the tube. The bending stress σ (MPa) can be expressed as

$$\sigma = \frac{M_o}{W_o}, \tag{3.1}$$

where M_o (Nmm) is the maximal moment loading the tube and W_o (mm^3) represents the section modulus of the tube profile. In this case, the moment can be expressed as

$$M_o = Fl, \tag{3.2}$$

where F (N) is force loading the arm and l (mm) is the length of the arm. The section modulus of the profile for a rectangular tubing is given by its dimensions as

$$W_o = \frac{bh^3 - (b-2t)(h-2t)^3}{6h}, \tag{3.3}$$

where h (mm) is profile height, b (mm) its width and t (mm) its wall thickness [45,46]. In the case of the proposed frame, the maximal bending stress in the tube was counted to 16.4 Mpa while the used carbon fiber composite according to the pages of the manufacturer [47] can handle up to 600 Mpa. This brings more than sufficient safety to cope with shocks during possible aggressive maneuvers of the UAV. But it still does not ensure enough rigidity to cope with a free fall of the UAV, in this case, forces to the arms can be many times greater and may come from directions causing greater stresses in the material.

3.4 Base component

The base component forms solid support for sensors, computer, batteries, launcher, and all other electronic parts. The cuboid-shaped base component consists of carbon fiber tubes creating edges of the cuboid and plates supporting electronics components and the launcher. Most of the sensors are mounted on a sensor tower which is part of the base component namely the LiDAR, thermal camera, and stereo camera. The fully equipped base component is shown in Figure 3.4.

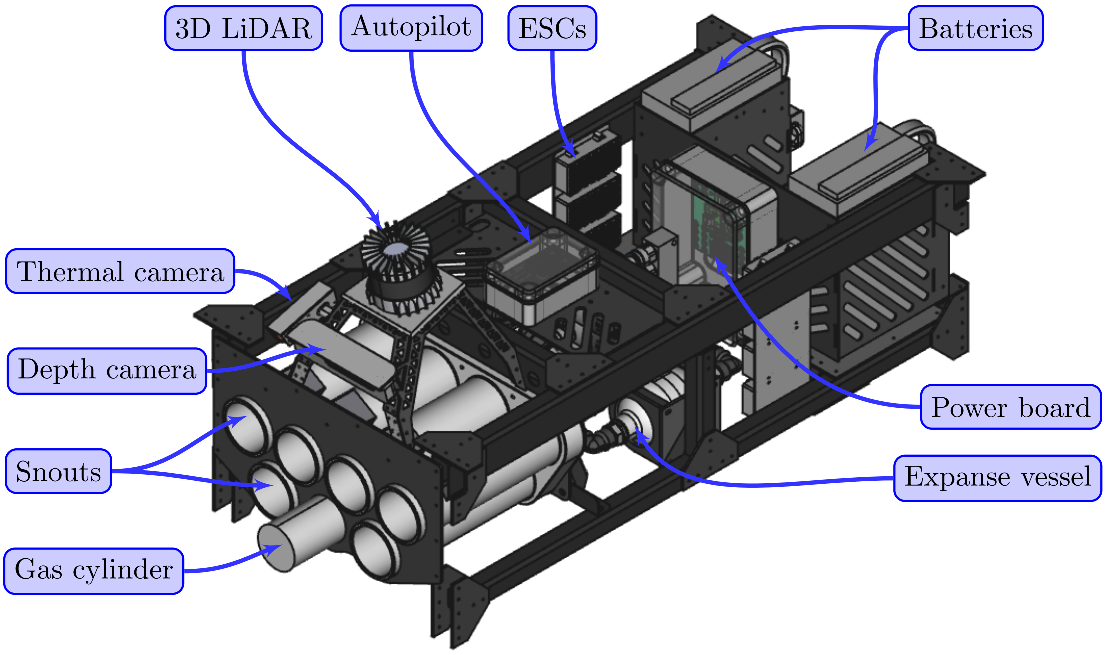


Figure 3.4: Visualization of the entire base component with integrated launcher, sensor tower and onboard electronics.

3.4.1 Sensor tower

The purpose of the sensor tower is to provide a solid junction between sensors and the rest of the UAV platform. Its main objective is to hold these optical sensors in such a position to provide them with the biggest field of view (FOV) while keeping it solid and minimal in size. The shape of the sensor tower was properly designed to fit this objective. This led to tilting the LiDAR and cameras in order to maximize seen area in front of the UAV and prevent reducing the FOV by seeing itself. The idea of positioning the sensor to fulfill these criteria is shown in Figure 3.5 and the final design of the sensor tower is shown in Figure 3.6. Figure 3.6 also shows camera covers which protect the cameras from direct contact with possible water drops.

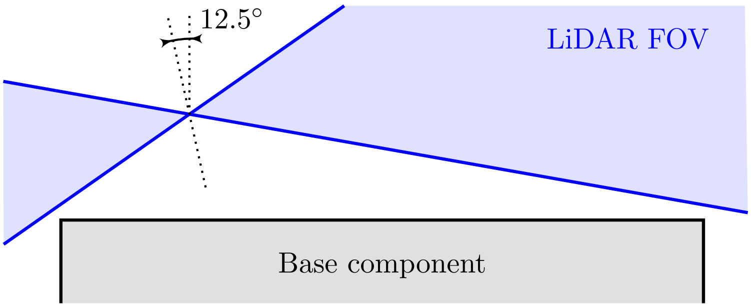


Figure 3.5: Tilting of the LiDAR axis to retrieve full FOV and minimal height of the UAV.

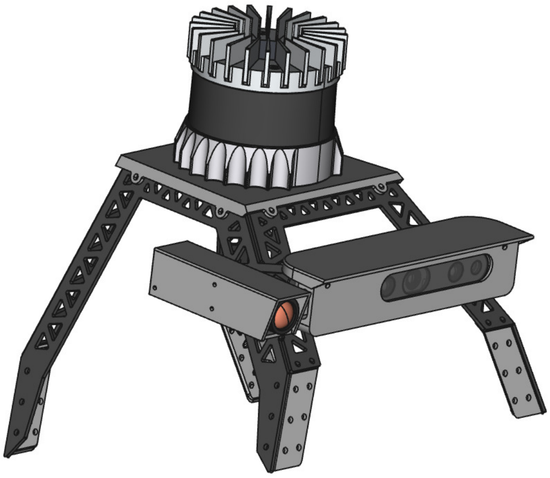


Figure 3.6: Visualization of the sensor tower equipped with LiDAR (Ouster OS1-64), depth camera (RealSense D435), and thermal camera (Teledyne FLIR Boson).

3.5 Ampule launcher

The ampule launcher is capable to launch and carry six ampules each in its own snout. Each snout is equipped with a servo-controlled crank on its estuary, preventing the capsule fall out during aggressive maneuvers and during launching other capsules. These six snouts are mounted to the main body part which keeps them connected to the UAV platform and the remaining launcher parts. The body part also holds a CO_2 gas cylinder which supplies the whole system with compressed gas. The gas cylinder can be easily accessed from the front of the launcher to replace an empty one with a fresh one. The compressed gas is piped through an expansion vessel and six solenoid valves to the end of each snout. When some of the valves are open gas flows from the expansion vessel to the appropriate snout and drives the ampule out of the snout. During this process, 320 ml of the gas (compressed to 1 MPa in the expansion vessel) expands to normal air pressure while rapidly increasing its volume. This expansion forces (launches) the ampule out of its snout.

The whole system filled with gas and all the ampules weights 20 kq, each ampule weights 500 g, and the gas cylinder contains 450 g of the gas. Visualization of the entire system is shown in Figure 3.7.

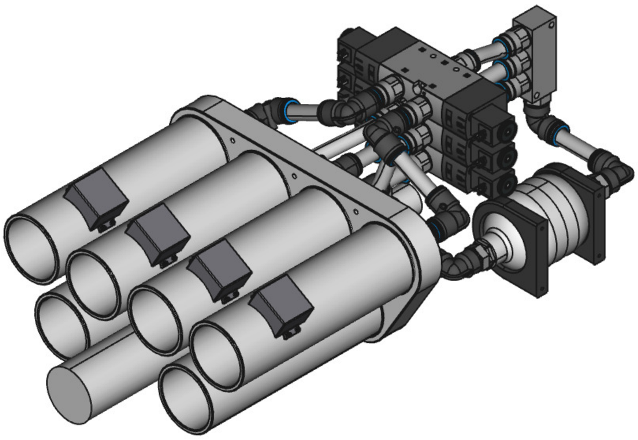


Figure 3.7: CAD model of the launcher capable of carrying and independently launching of six fire exhausting ampules.

3.6 Drive

Drive provides the necessary thrust for the UAV and therefore allows it to fly. It consists of batteries, electronic speed controllers (ESC), motors, and propellers. Selection of these components affects the flight characteristic of the UAV, namely weight, size, generated thrust, and flight time. As the commercial market is extremely huge, the choice of components must compromise between these characteristics and likewise account for the mutual compatibility between motor propellers, motors, and battery voltage.

There is an online tool eCalc [2] suitable for finding optional drive parts and their parameters. This tool provides a calculation of the flight characteristic according to the selected drive component and UAV weight. Selected components and their parameters are summarized in Table 3.1 and further described in detail in Section 3.6.1. These components compromise between the UAV size and high-weight payload, and ensure mutual compatibility. Flight characteristics of the UAV with this designed drive are described in Figure 3.8.

Segment	Component	Specifications	Weight [g]
Drive	Motors	T-Motor U12II	8 × 778
	Electronic Speed Controller	T-Motor FLAME 100A 14S	8 × 139
	Propellers	T-Motor Prop 30x10, 5	8 × 97
	Accumulator	Tattu 6S LiPo	4 × 1974
Frame	Base component	Carbon and 3D printed parts	2621
	Core with landing gear	Carbon parts	1653
Electronics	Central processing unit	Intel NUC 11TNKi7	230
	Autopilot	Pixhawk 4	16
	Power board	Group made PCB	123
	Radio control receiver	Radiomaster R81 Receiver	2
Sensors	Optical rangefinders	Garmin Lidar Lite v3	22
	Front-facing depth camera	RealSense D435	72
	Thermal camera	Teledyne FLIR Boson	18
	GPS/GLONASS receiver	PixHawk ublox Neo-M8N	23
	3D LiDAR	Ouster OS1-64	447
Payload	Extinguishing capsule launcher	Launcher	16638
	Extinguishing capsule	Bonpet Grenade	6 × 557
			41215

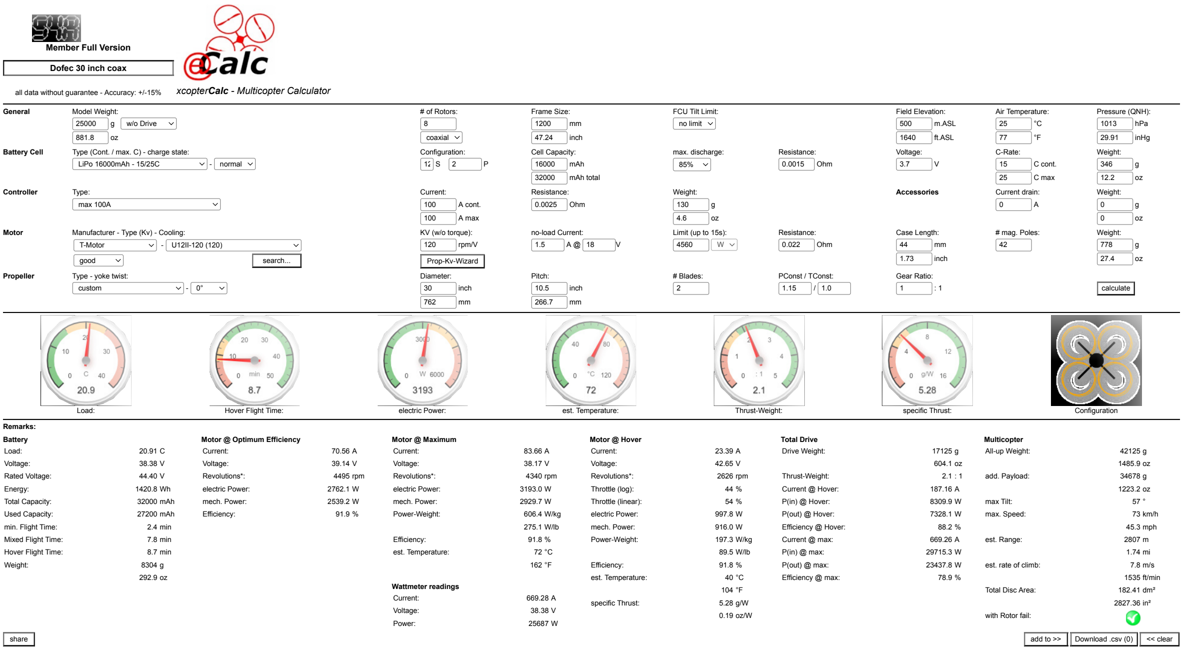
Table 3.1: List of the used components with their weight.

3.6.1 Drive components

Battery

The battery pack powers all electronic components on the UAV, including motors, ESC, sensors, autopilot, onboard computer, and even the solenoid valves and servos on the launcher. The main parameters of batteries are capacity, weight, maximum discharging rate, and the number of serial and parallel cells. The number of cells in a series designates battery output voltage that should be accepted by ESCs and motors. The capacity of the battery determines how much energy can be stored in the battery and the maximum discharging rate denotes the maximum current that can be drawn from the battery.

To efficiently power motors applied in this design, 12 LiPo cells in series with nominal voltage 44.4 V are employed. To increase the capacity of the battery pack, two of these 12 cells batteries are connected in parallel. Low market and shipping issues with 12 cells battery packs lead to the splitting of these packs to the serial connection of two 6-cell battery packs commonly available. These splitted battery packs also reduced the time needed for their maintenance, because charging them and discharging them before storage can be done in parallel. The final battery pact contains four 6-cell batteries connected to two parallel couples of



Member Full Version

Dofec 30 inch coax

all data without guarantee - Accuracy: +/-15%

xcopterCalc - Multicopter Calculator

General — Model Weight: 25000 g, w/o Drive; 881.8 oz — # of Rotors: 8, coaxial — Frame Size: 1200 mm; 47.24 inch — FCU Tilt Limit: no limit — Field Elevation: 500 m.ASL; 1640 ft.ASL — Air Temperature: 25 °C; 77 °F — Pressure (QNH): 1013 hPa; 29.91 inHg

Battery Cell — Type (Cont. / max. C) - charge state: LiPo 16000mAh - 15/25C - normal — Configuration: 12 S 2 P — Cell Capacity: 16000 mAh; 32000 mAh total — max. discharge: 85% — Resistance: 0.0015 Ohm — Voltage: 3.7 V — C-Rate: 15 C cont.; 25 C max — Weight: 346 g; 12.2 oz

Controller — Type: max 100A — Current: 100 A cont.; 100 A max — Resistance: 0.0025 Ohm — Weight: 130 g; 4.6 oz — **Accessories** — Current drain: 0 A — Weight: 0 g; 0 oz

Motor — Manufacturer - Type (Kv) - Cooling: T-Motor - U12II-120 (120); good — KV (w/o torque): 120 rpm/V — no-load Current: 1.5 A @ 18 V — Limit (up to 15s): 4560 W — Resistance: 0.022 Ohm — Case Length: 44 mm; 1.73 inch — # mag. Poles: 42 — Weight: 778 g; 27.4 oz

Propeller — Type - yoke twist: custom - 0° — Diameter: 30 inch; 762 mm — Pitch: 10.5 inch; 266.7 mm — # Blades: 2 — PConst / TConst: 1.15 / 1.0 — Gear Ratio: 1 : 1

Remarks:

Battery	
Load:	20.91 C
Voltage:	38.38 V
Rated Voltage:	44.40 V
Energy:	1420.8 Wh
Total Capacity:	32000 mAh
Used Capacity:	27200 mAh
min. Flight Time:	2.4 min
Mixed Flight Time:	7.8 min
Hover Flight Time:	8.7 min
Weight:	8304 g
	292.9 oz

Motor @ Optimum Efficiency	
Current:	70.56 A
Voltage:	39.14 V
Revolutions*:	4495 rpm
electric Power:	2762.1 W
mech. Power:	2539.2 W
Efficiency:	91.9 %

Motor @ Maximum	
Current:	83.66 A
Voltage:	38.17 V
Revolutions*:	4340 rpm
electric Power:	3193.0 W
mech. Power:	2929.7 W
Power-Weight:	606.4 W/kg
	275.1 W/lb
Efficiency:	91.8 %
est. Temperature:	72 °C
	162 °F

Wattmeter readings	
Current:	669.28 A
Voltage:	38.38 V
Power:	25687 W

Motor @ Hover	
Current:	23.39 A
Voltage:	42.65 V
Revolutions*:	2626 rpm
Throttle (log):	44 %
Throttle (linear):	54 %
electric Power:	997.8 W
mech. Power:	916.0 W
Power-Weight:	197.3 W/kg
	89.5 W/lb
Efficiency:	91.8 %
est. Temperature:	40 °C
	104 °F
specific Thrust:	5.28 g/W
	0.19 oz/W

Total Drive	
Drive Weight:	17125 g
	604.1 oz
Thrust-Weight:	2.1 : 1
Current @ Hover:	187.16 A
P(in) @ Hover:	8309.9 W
P(out) @ Hover:	7328.1 W
Efficiency @ Hover:	88.2 %
Current @ max:	669.26 A
P(in) @ max:	29715.3 W
P(out) @ max:	23437.8 W
Efficiency @ max:	78.9 %

Multicopter	
All-up Weight:	42125 g
	1485.9 oz
add. Payload:	34678 g
	1223.2 oz
max Tilt:	57 °
max. Speed:	73 km/h
	45.3 mph
est. Range:	2807 m
	1.74 mi
est. rate of climb:	7.8 m/s
	1535 ft/min
Total Disc Area:	182.41 dm²
	2827.36 in²
with Rotor fail:	✓

Figure 3.8: UAV design characteristic using the eCalc web tool [2].

serially connected batteries. This configuration provides maximum continuous current 480 A. Its total capacity reaches 32 Ah with a total weight of 7896 g.

Electronic Speed Controller

Electronic speed controllers provide a bridge between a logic signal from autopilot and three-phase high-current power to the motors. They form this three-phase sinusoidal voltage from direct current voltage input from batteries, the frequency of the output voltage responds to low voltage pulse-width modulated input signal from the autopilot. The frequency of the output voltage affects the speed of rotation of each motor which then affects the thrust provided by it. Autopilot commands the ESC in order to stabilize and control the UAV.

The key parameters of ESCs are continuous current, maximum peak current, and input voltage range. In the proposed design, the ESCs with 100 A continuous current and nominal input voltage of 44.4 V are used. These ESCs ensure IP55 certification which allows their utilization without other water shields or cases.

Motor

It is common to use a brushless direct current motor in a UAV system for its quick reactions and easy speed control. The motor is determined by several electrical parameters, namely its power limit, resistance, current, and voltage limit as well as physical parameters especially its geometrical shape dimensions, and weight. Also, geometrical compatibility between the motor and propeller should be ensured.

The parameters most affecting the design of the drive are the power limit, the motor velocity constant K_v (revolution per minute per volt), motor resistance, and its weight. The power limit defines the maximal mechanical power that could be generated by the motor. The resistance defines the direct current resistance of the motor coils, it defines the dependency between the input voltage and current flowing through the motor. The K_v constants come from a simple electrical motor model described in [48]. This model defines an electrical motor as a serial connection of a resistor with the same resistance as the motor and back electromotive voltage source, where the amplitude of back electromotive voltage V_{peak} depends on the motor speed ω. K_v is the number of motor revolutions per minute divided by back electromotive voltage (revolution per minute per volt)

$$K_v = \frac{\omega}{V_{peak}}. \tag{3.4}$$

Since the motor resistance is typically small (e.g., 22 mΩ), it is a good approximation that the speed of an unloaded motor equals K_v multiplied by the motor input voltage.

This UAV design employs T-MOTOR U12II motor with maximal power 4560 W, 120 Kv and 22 mΩ resistance. This motor also offers IP55 certification which allows using it on UAV without additional covers.

Propeller

The special shape of the propeller generates air pressure difference which causes lift, this principle is well-known in the field of aeronautics and it is henceforth referred to as thrust in our case. The amount of generated thrust depends on the shape of the propeller as well as on its diameter and pitch. Diameter defines the size of the propeller and pitch defines the propeller's vertical shift after one revolution in a solid environment.

This UAV design employs a carbon fiber composite propeller with 30" diameter and 10.5" pitch. The manufacturer recommends these propellers with the selected motor series and guarantees 5.5 kg of thrust per rotor at 60% throttle [] which is ideal for application in the proposed design.

3.7 Electronics

The onboard electronics consist of an autopilot, onboard computer, power delivery, drive parts discussed before, and additional sensory equipment. The components not discussed before are discussed further in this section.

Each of these onboard electronics components requires a power supply and a communication interface. The power from batteries is delivered to a power board designed to redistribute it for ESCs, autopilot, onboard computer, and sensors. High-level control algorithms are executed on the onboard computer therefore the computer must communicate with all the sensors and actuators. Some of the components are connected to the computer through autopilot due to its real-time software support whereas others are connected through a microcontroller on the power board and some can be connected directly to the computer. Power and signal connections are shown in Figure 3.9.

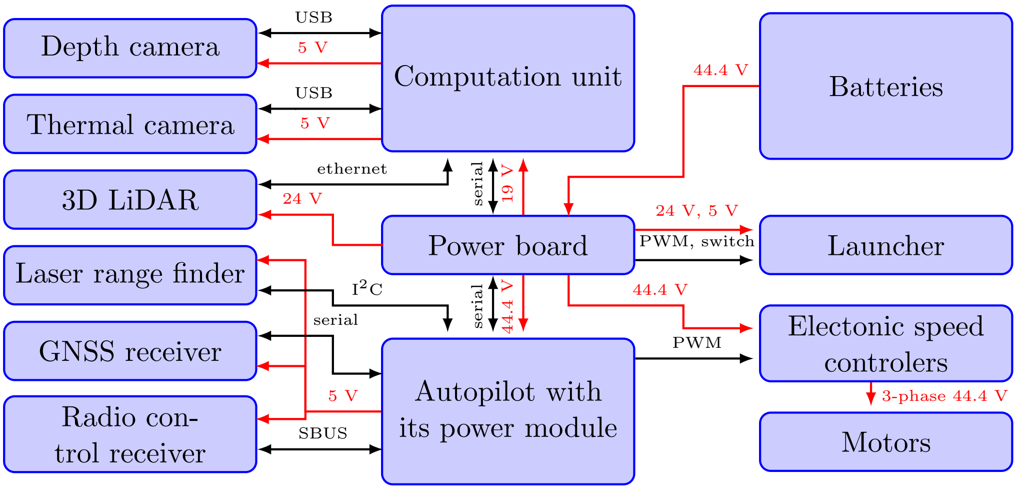


Figure 3.9: High-level connection diagram. Red highlights the power lines and black highlights the signal lines.

3.7.1 Power distribution board

The main purpose of the power distribution board is to the battery power to all other components with suitable voltage the particular components expect. Other purposes of the board are to control the launcher and ensure a link between the autopilot and the onboard computer.

To provide necessary thrust, the motors and ESC must be powered by 12 cells battery with nominal voltage 44.4 V and maximal 50.4 V. This battery voltage is not suitable for almost any other component therefore the power distribution board contains three step-down converters providing voltages of 24 V, 19 V and 5 V. The voltage 24 V is used to power LiDAR and switch solenoid valves of the launcher. The 19 V power supply powers only the onboard computer and it ensures protection from disturbing computer power by other devices. The 5 V power is used for launcher servos as well for the communication part of this board.

The communication part of the board consists of an STM microcontroller and FT4232H chip. The microcontroller STM32F103C8T controls the launcher servos and transistors switching the solenoid valves. The FT4232H chip provides translation between the USB connection to the computer and four channels of the serial interface. One of these channels is connected to the microcontroller to enable launcher control from high-level computer application and reporting of the board status including voltages of all power supplies to the master computer.

The power board is mounted in an electrical box with IP65 certification to protect it from the environment.. The power distribution board was designed, assembled, and programmed just for these purposes.

3.7.2 Onboard computer

The autonomous mission control, fire detection, fire localization, and other necessary algorithms including the robot operating system are executed on a common x86-64 computer. To run these algorithms onboard, the platform is equipped with a powerful but small computer

Intel NUC 11TNKi7 with a 4-core processor with 4.7 GHz frequency. Reasons for choosing this computer as onboard are the processing power, compact design, a sufficient number of peripherals, input voltage of 12 V to 24 V and 230 g weight. The computer board is mounted in a specially designed water-protecting box.

3.7.3 Autopilot

An autopilot takes care of low-level UAV stabilization, it controls the thrust of each motor in such a way the UAV stays in a fixed position or moves around a trajectory specified by the onboard computer. This stabilization process mostly relays on fast inertial measurement unit (IMU) providing linear acceleration and angular velocities. It can also apply data from other sensors like magnetometer, gyroscope, and barometer or from localization methods like GNSS or feature matching methods.

The presented design employs PixHawk 4 [49] autopilot based on the PX4 stack, as required by the system for the stabilization of a UAV developed by the Multi-Robot Systems group. This device comes with an internal IMU containing three accelerometers and gyroscopes, a barometer for altitude measurement, a magnetometer for azimuth measurement, and the ability to connect many other sensors and peripherals. It serves a fully redundant design, which eliminates the failure of any of the internal sensors or processors.

The autopilot is placed on the top of the base component above the center of gravity of the UAV in an electrical box with IP65 certification together with its power module, which is also fully redundant and can be powered directly from the battery voltage.

3.7.4 Sensory equipment

The UAV platform needs to be well-sensor-equipped to provide enough data for UAV stabilization, localization, mission planning, fire detection, and localization. The UAV will also operate in various environments where it needs efficient localization systems and obstacle detection. For this purpose LiDAR, laser rangefinder, and GNSS are employed on the desired platform. The depth and thermal camera then take care of supplying sufficient data for fire detection and localization. All these sensors are described below.

Laser range finder

A laser rangefinder is employed as an absolute altimeter measuring distance to the ground. In contrast with the barometer integrated within the autopilot, the rangefinder is not affected by changes in air pressure causing drifts in altitude estimation but can be affected by flying over some obstacle causing disturbance of its altitude estimate. Therefore both of these altitude estimation methods are fused together. The presented design uses Garmin LiDAR-Lite v3 laser rangefinder, which is lightweight and has specifications of 40 m range, up to 1 cm resolution and ± 2.5 cm accuracy.

GNSS

PixHawk ublox Neo-M8N GPS/GLONASS receiver is employed to provide UAV absolute localization in open space environments. The selected module offers integrated magnetometer IST8310, the accuracy of 2.5 m, and compatibility with the selected autopilot [50].

3D LiDAR

LiDAR stands for light detection and ranging which covers any light-based distance measurements. This distance measurement is typically based on the time of flight method where time t (s) between emitted and returned reflected light beam is measured and distance d (m) is counted based on a constant speed of light c (m/s)

$$d = \frac{ct}{2}. \tag{3.5}$$

3D LiDAR employs this principle in multi-directorial distance measurement which results in a local map of surfaces in FOV of the LiDAR. These stamped local maps can be compared together to determine UAV movement between the times when the maps were scanned and improve the localization of the UAV. It can be also concatenated to build a complete map of the ambient environment to plan the mission on that map.

The proposed design uses Ouster OS1-64 LiDAR which provides a 45° vertical field of view, 360-degree horizontal FOV, 64 laser beams scanning in parallel in the vertical FOV, 90 m range, and 2.5 cm maximal range deviation. These qualities with industrial design ensuring IP68 certificate and robustness make it an unrivaled choice.

Thermal camera

A thermal camera is employed in the fire detection and localization task, it provides a stream of thermal images describing the surface temperature of the seen scene. The principle of thermal imaging is closely described in [43] where the fire detection procedure was designed and described in detail. The thresholding-based fire detection procedure obtains a 2D fire position in the thermal image coordination system which can be combined with data from other sensors to obtain a 3D fire position.

The proposed design uses Teledyne FLIR Boson thermal camera. This camera offers sufficient image resolution, is small in size, and is lightweight. It provides lower accuracy at temperature measurement than previously used Wiris Pro 640 thermal camera but the accuracy of measurement is not needed for fire detection due to its high-temperature contrast. On the other hand, the Boson thermal camera provides its image stream via a USB interface which acts faster than the high-level real-time streaming protocol (RTSP) used by the Wiris camera. The unpredictable latency of RTSP brings big fire localization issues caused by the desynchronization of thermal images with data from others sensors.

Depth camera

A depth camera serves information about the 3D scene in its FOV in form of a 2D image stream, where each pixel of the image codes the depth of the scene — a distance between the camera and a surface seen by the pixel.

In the proposed design, a depth camera is used to estimate the 3D position of a detected fire by the algorithm presented in [43]. Although fire position estimate can be done from LiDAR scans, the depth camera is used for backward compatibility with fire localization software developed before and for the future fusion of LiDAR and depth camera data to improve the accuracy of the fire localization.

In the proposed design Intel Realsense D435i [51] depth camera is used. This depth camera offers a credible solution due to its long-term usage by the robotic community. The depth camera consists of two infrared cameras finding laser markers projected by an infrared projector, when the marker is recognized in both cameras its spatial position can be estimated.

3.7.5 Final multicopter

The final multicopter design is shown in Figure 3.10 and Figure 3.11 shows photos of the real constructed prototype of the UAV. This fully loaded prototype weights 41.2 kg and reaches size of 202 × 165 cm. The UAV is capable of autonomous flight longer than 12 min and horizontal discharge of six fire-exhausting ampules (each weighting 557 g) independently on each other. The platform contains all necessary equipment for the autonomous mission including localization, mapping, planning and the detection of a fire and its localization.

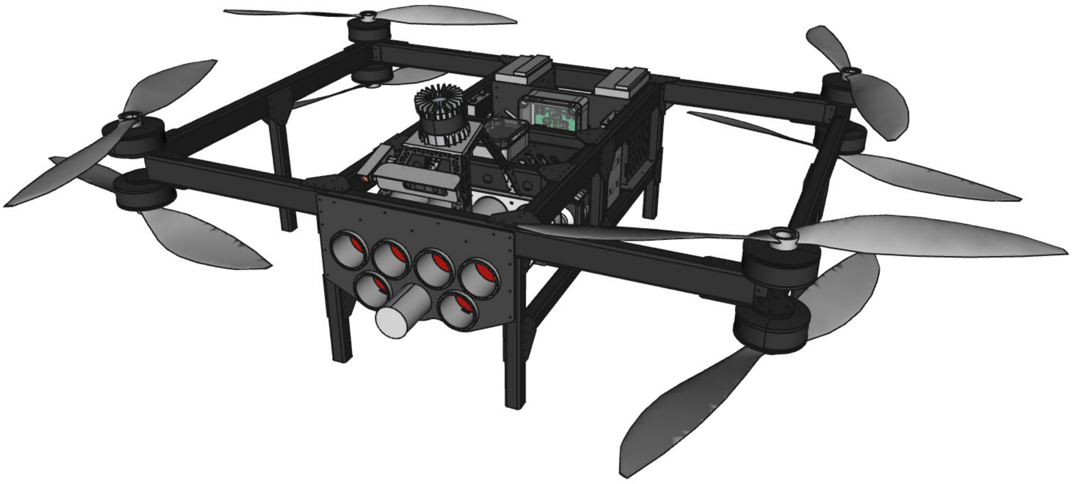
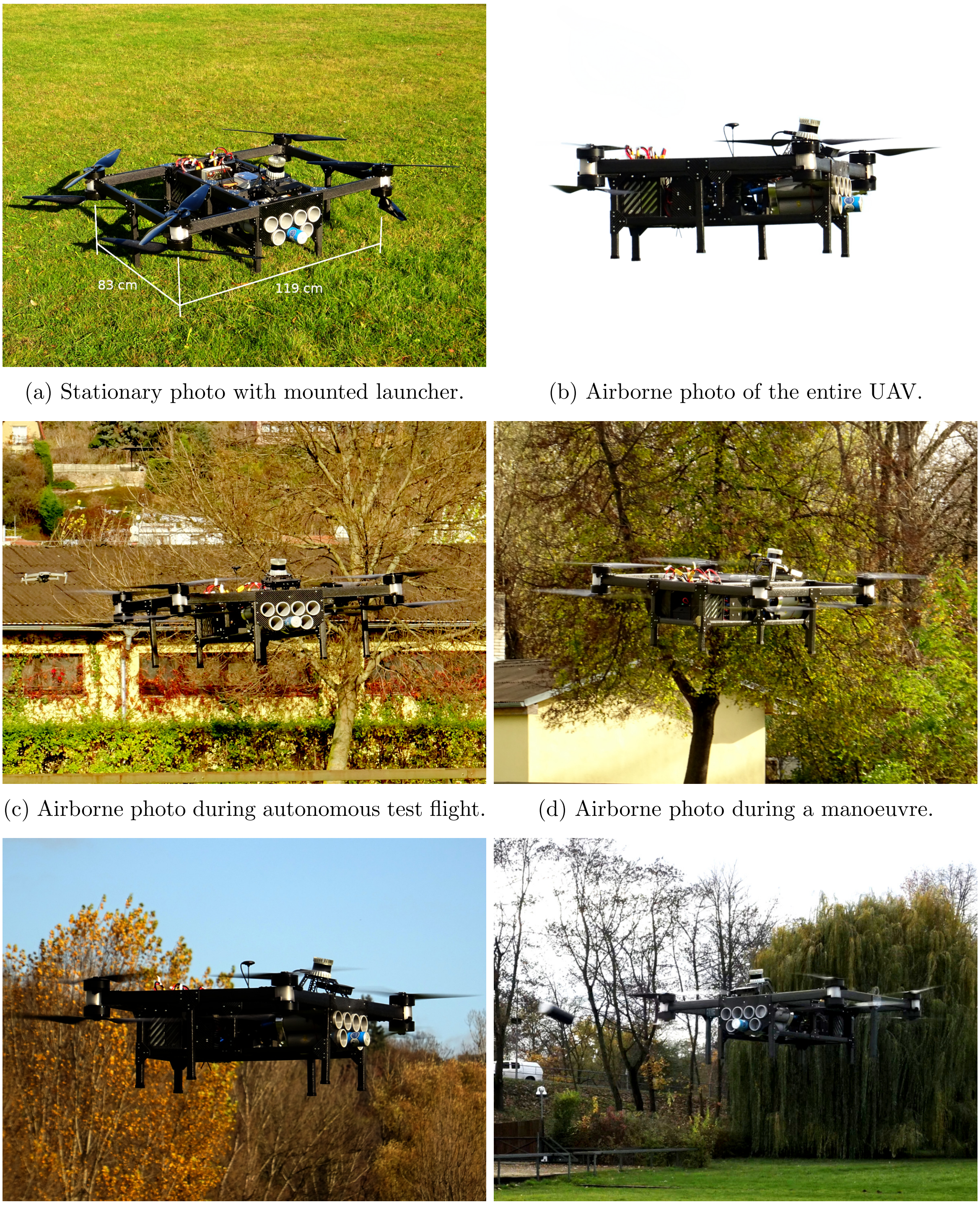


Figure 3.10: Final design of the entire multicopter.



(a) Stationary photo with mounted launcher.

(b) Airborne photo of the entire UAV.

(c) Airborne photo during autonomous test flight.

(d) Airborne photo during a manoeuvre.

(e) Airborne photo of the complete system.

(f) Airborne photo during ampule discharge.

Figure 3.11: Realized prototype of the UAV.

Chapter 4: Fire Tracking

Contents

Since fire-searching mission requires fast and autonomous fire location and classification, it is necessary keep track of all seen fires even if it is not in the sensors' field of view. With a fire being in a sensor field of view, it can be detected, classified, and localized in 3D space with the fire localization system described in Section 2.3. But to spatially and temporally track a set of out-of-FoV fires, using only the detection system is not sufficient and a global fire-tracking system has to be introduced. This system should keep all fire hypotheses with their positions and uncertainty, update their properties and merge close and similar hypotheses together.

To allow an efficient work with position and its uncertainty, a sufficient position representation must be used. For that purpose probabilistic model for stochastic variable with normal probability distribution is used. Have stored or measured object position estimate $\vec{p} \in \mathbb{R}^3$ with a covariance matrix $\mathbf{P} \in \mathbb{R}^{3\times3}$ representing its uncertainty, probability of the object on exact position $\vec{x} \in \mathbb{R}^3$ with respect to this position estimate $\vec{p}$ is

$$p(\vec{x}|\vec{p}) = \frac{1}{\sqrt{(2\pi)^3|\mathbf{P}|}} e^{-\frac{1}{2}(\vec{x}-\vec{p})^T\mathbf{P}^{-1}(\vec{x}-\vec{p})}. \tag{4.1}$$

This representation is ideal for fire position because a fact that the fire appears and disappears continuously in close proximity to its epicenter match the representation of it as probability on specific position according to its position estimate (the epicenter).

4.1 Measurement representation

Measurement consists of fire position $\vec{p} \in \mathbb{R}$ and its temperature t (°C) estimated by fire localization subsystem (described in Section 2.3) a and covariance matrix $\mathbf{P} \in \mathbb{R}^{3\times3}$ expressing its uncertainty. The covariance matrix is not estimated by the fire localization system and therefore must be specified afterwards. A measurement is given as

$$m = (\vec{p}, \mathbf{P}, t). \tag{4.2}$$

Accuracy of fire-localization method using a pair of thermal and depth cameras depends depends on two factors. The first is the accuracy of back-projecting a fire detection in the thermal image to unit vector $\vec{r} \in \mathbb{R}^3$ pointing from the thermal camera optical center in direction of the fire. This possible direction inaccuracy can come from imprecise camera calibration or imperfect detection of the fire. The second factor is the accuracy of the fire distance estimate d (m) which depends on mutual camera position and the depth camera accuracy. Because the measured fire position position $\vec{p}$ is composed from the measured direction vector $\vec{r}$ and distance from camera d as

$$\vec{p} = d\vec{r}. \tag{4.3}$$

Let σ_{dir} (rad) denote the standard deviation of angle ϕ (rad) belonging to $\vec{r}$ pointing in the direction of a fire and σ_{dist} (m) denote the standard deviation of the measured distance d. It is common to specify a linear quotient α_{dist} (%) representing sensor uncertainty in percentage of the measured value as

$$\sigma_{dist} = \alpha_{dist} d. \tag{4.4}$$

The above parameters can be used for estimating the variance of each coordination of fire position $\vec{p}$. Figure 4.1 shows that the standard deviation of position in direction perpendicular to $\vec{r}$ is

$$\sigma_n = 2d \tan\left(\frac{\sigma_{dir}}{2}\right) \tag{4.5}$$

which remains the same in all directions perpendicular to $\vec{r}$, because deviation σ_{dir} is identical in all directions. Given σ_n, the measurement covariance matrix $\mathbf{P}$ can be specified.

The covariance matrix describing the variance of a fire position can be expressed in orthonormal basis $\mathcal{O}^N$ defined by matrix

$$\mathbf{B}_m = [\vec{n}_1 \, \vec{n}_2 \, \vec{r}], \tag{4.6}$$

where the vector $\vec{n}_1$ is an arbitrary unit vector perpendicular to $\vec{r}$ and $\vec{n}_2 = \vec{p} \times \vec{n}_1$. In this base, the covariance matrix is given as

$$\mathbf{P}^{\parallel} = \begin{bmatrix} \sigma_n^2 & 0 & 0 \\ 0 & \sigma_n^2 & 0 \\ 0 & 0 & \sigma_{dist}^2 \end{bmatrix}. \tag{4.7}$$

To transform the matrix from basis $\mathcal{O}^N$ to basis $\mathcal{O}$, the covariance matrix can be transformed by matrix $\mathbf{B}$ as

$$\mathbf{P} = \mathbf{B}_m^{-1} \mathbf{P}^{\parallel}. \tag{4.8}$$

Matrix $\mathbf{P}$ is used in associating measurements with hypotheses (Section 4.3) and in updating hypotheses states (Section 4.4).

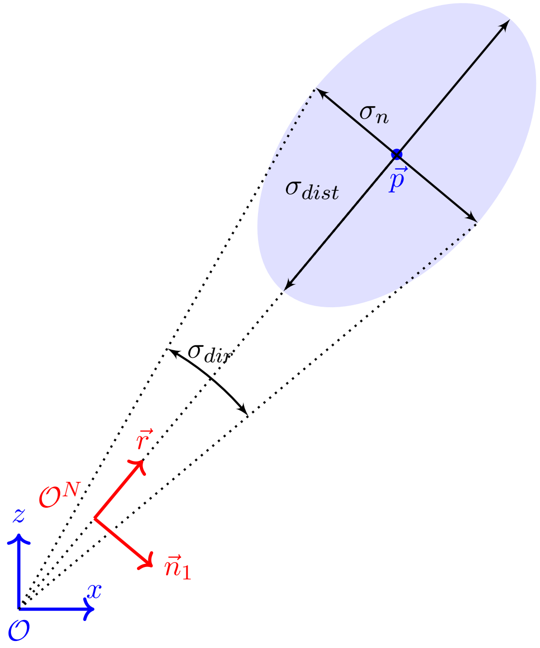


Figure 4.1: Fire position uncertainty given by the covariance matrix **P** (highlighted as blue ellipse) defined from sensors' uncertainties σ_{dist} and σ_{dir}.

4.2 Hypothesis representation

A fire hypothesis stores information about a single detected and localized fire. The hypothesis consists of a position of the fire $\vec{p}$ with uncertainty represented by a covariance matrix $\mathbf{P}$, fire temperature t and time of last update τ. Each hypothesis i represents one real fire in the environment as tuple

$$h_i = (\vec{p}_i, \mathbf{P}_i, t_i, \tau_i). \tag{4.9}$$

As the fire is a dynamical system that can change its properties over time, the hypothesis i is understood as state description of a stochastic linear dynamic system with state

$$\vec{x}_i[\tau] = \begin{bmatrix} \vec{p} \\ t \end{bmatrix} \tag{4.10}$$

in discrete time step τ.

The purpose of the fire tracking system is to keep all fire hypotheses saved in a set of hypotheses $\mathcal{F}$ with their properties $h_i \in \mathcal{F}$ and update them when new measurement m arrives from the localization system. Before the hypothesis update, the measurement is associated (see Section 4.3) to the most likely hypothesis or new hypothesis is created. When a new measurement is associated to the most likely hypothesis, the measurement is fused into the hypothesis. The fusion process, discussed in Section 4.4, is similar to the correction step of the Kalman filter.

4.3 Measurement-hypothesis association

The new measurement m is compared with all stored fire hypotheses $h_i = (\vec{p}_{h_i}, \mathbf{P}_{h_i}, t_{h_i}, \tau_{h_i}) \in \mathcal{F}$, based on mutual distance and marginal likelihood. The most similar hypothesis h_c is updated by this measurement if the measurement is spatially closer than an update distance threshold t_u (m) and its logarithmic marginal likelihood s greater than threshold t_l (-). If the measurement is not suitable for hypothesis update, then a new hypothesis is created from the measurement and stored within $\mathcal{F}$.

The association starts with finding the maximal logarithmic marginal likelihood between the measurement and all the hypotheses. Hypothesis h_c with the greatest logarithmic marginal likelihood $l(m, h_c)$ is updated by the incoming measurement $m = (\vec{p}_m, \mathbf{P}_m, t_m)$, if it fulfills following criteria.

- The logarithmic marginal likelihood is greater than threshold $l(m, h_c) > t_l$.
- The distance between the measurement and hypothesis is smaller than threshold $\|\vec{p}_{h_c} - \vec{p}_m\| < t_u$.

If the above criteria are not satisfied, a new hypothesis is created based on the measurement.

The logarithmic marginal likelihood between hypothesis h_i and a measurement m represents the probability that measurement m belongs to hypothesis h_i. Using Equation 4.1, the logarithmic marginal likelihood can be expressed as

$$l(m, h) = \log p(\vec{p}_m | \vec{p}_h) = -\frac{1}{2}\left((\vec{p}_m - \vec{p}_h)^T \mathbf{S}^{-1} (\vec{p}_m - \vec{p}_h) + \log|\mathbf{S}| + 3\log 2\pi\right), \tag{4.11}$$

where $\mathbf{S} = \mathbf{P}_m + \mathbf{P}_h$. The hypothesis h_c associated and fused with measurement m is then given as

$$h_c = \arg\max_{h \in \mathcal{F}} l(m, h). \tag{4.12}$$

4.4 Hypothesis update

Fire hypothesis represents a linear discrete time system, therefore it can be updated by the Kalman filter (KF) see Section 4.4.1. As fire position is assumed to be static and self-change in its temperature negligible the system matrix is assumed to be $\mathbf{A} = \mathbf{I}$. The output matrix $\mathbf{C} = \mathbf{I}$ because states of the system (see Equation 4.10) directly represent a hypothesis position and temperature. The process noise is $\mathbf{Q} = \mathbf{0}$ since the system model is stationary. With these definitions, Equation 4.26 and Equation 4.27 do not affect the system state, which is affected only in the correction step computed at arrival of measurements.

When new measurement $m = (\vec{p}_m, \mathbf{P}_m, t_m)$ arrives and is associated with hypothesis $h_c = (\vec{p}_c, \mathbf{P}_, t_c, \tau_c)$ the hypothesis is updated according to the KF correction steps

$$\mathbf{K} = \mathbf{P}_c \left(\mathbf{P}_c + \mathbf{P}_m\right)^{-1} \tag{4.13}$$

$$\vec{p}_c \leftarrow \vec{p}_c + \mathbf{K}\left(\vec{p}_m - \vec{p}_c\right) \tag{4.14}$$

$$\mathbf{P}_c \leftarrow \left(\mathbf{I} - \mathbf{K}\right)\mathbf{P}_c. \tag{4.15}$$

And time of last update is updated by actual time step τ_{now}

$$\tau_c \leftarrow \tau_{now}. \tag{4.16}$$

4.4.1 The Kalman filter

The Kalman filter (KF) [52] is a system state estimator proposed in 1960. It is a very efficient algorithm to estimate the evolution of dynamic systems. KF is used in many different applications [53] such as in navigation systems [54–57], financial models [58, 59], and image processing [60] also in monitoring of forest fires [61].

Have linear discrete-time stochastic dynamic system its n states in step k is saved in $\vec{x}(k) \in \mathbb{R}^n$ then KF can be used to predict future states of the system $\vec{x}(k+1)$. To describe a stochastic system it is necessary to use probabilistic values. Lets have N of the same systems with the same initial state $\vec{x}_i(0) = \vec{x}(0), i = 1, \ldots, N$, then mean state of the systems in step k is

$$\vec{x}_m(k) = \sum_{i=1}^{N} \frac{\vec{x}_i(k)}{N} \tag{4.17}$$

Covariance matrix of state vector $\vec{x}_m(k)$ is

$$\mathbf{P}_{\vec{x}_m}(k) = \sum_{i=1}^{N} \frac{\vec{x}_i(k)\vec{x}_i^T(k)}{N}. \tag{4.18}$$

The dynamics of system i are described by

$$\vec{x}_i(k+1) = \mathbf{A}\vec{x}_i(k) + \vec{w}_i(k), i = 1, \ldots, N \tag{4.19}$$

$$\vec{y}_i(k+1) = \mathbf{C}\vec{x}_i(k) + \vec{v}_i(k), i = 1, \ldots, N, \tag{4.20}$$

where $\vec{y}_i \in \mathbb{R}^m$ is output of the system, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is system matrix, $\mathbf{C} \in \mathbb{R}^{m \times n}$ is output matrix, $\vec{w}_i \in \mathbb{R}^n$ is process noise and $\vec{v}_i \in \mathbb{R}^m$ is measurement noise. The random noises, $\vec{w}_i$ and $\vec{v}_i$, are assumed to be stationary white noises. Its covariance matrices are

$$\mathbf{Q}(k) = \sum_{i=1}^{N} \frac{\vec{w}_i(k)\vec{w}_i^T(k)}{N} \tag{4.21}$$

$$\mathbf{R}(k) = \sum_{i=1}^{N} \frac{\vec{v}_i(k)\vec{v}_i^T(k)}{N}. \tag{4.22}$$

Then KF defines two steps to obtain stochastic system state estimation. The first is the so-called correction step and is performed when new measurement $\vec{y}_m(k)$ come. This step update system states $\vec{x}_m$ to $\vec{x}'_m$ and its covariance matrix according to new measurement $\vec{y}_m(k)$ concerning its inaccuracy. This step is performed by computation of

$$\vec{x}'_m(k) = x_m(k) + \mathbf{K}(k)\left[\vec{y}_m(k) - \mathbf{C}x_m(k)\right] \tag{4.23}$$

$$\mathbf{P}_{\vec{x}'_m}(k) = \left[\mathbf{I} - \mathbf{K}(k)\mathbf{C}\right]\mathbf{P}_{\vec{x}_m}(k). \tag{4.24}$$

In correction step the $\mathbf{K}(k)$ is Kalman gain defined as

$$\mathbf{K}(k) = \mathbf{P}_{\vec{x}_m}(k)\mathbf{C}^T\left[\mathbf{C}\mathbf{P}_{\vec{x}_m}(k)\mathbf{C}^T + \mathbf{R}\right]^{-1}. \tag{4.25}$$

The second so called prediction step uses corrected values $\vec{x}'_m(k)$ and $\mathbf{P}_{\vec{x}'_m}(k)$ to predict future state $\vec{x}_m(k+1)$ and its covariance matrix $\mathbf{P}_{\vec{x}_m}(k+1)$ by computation of

$$\vec{x}_m(k+1) = \mathbf{A}x'_m(k) \tag{4.26}$$

$$\mathbf{P}_{\vec{x}_m}(k+1) = \mathbf{A}\mathbf{P}_{\vec{x}'_m}(k)\mathbf{A}^T + \mathbf{Q}. \tag{4.27}$$

In the purposed design fire hypothesis positions are represented by a simple linear stochastic dynamic system which is updated by the KF correction step every time a new estimation of position is done.

4.5 Fire tracking algorithm

Every time a measurement is obtained from the localization system, the measurement is amended by its position covariance matrix $\mathbf{P}_m$. The measurement $m = (\vec{p}_m, \mathbf{P}_m, t_m)$ is either assigned to the existing fire hypothesis $h_i = (\vec{p}_{h_i}, \mathbf{P}_{h_i}, t_{h_i}, \tau_{h_i}) \in \mathcal{F}$, or a new hypothesis is created and added into

$$\mathcal{F} \leftarrow \mathcal{F} \cup \{h_n\}, \tag{4.28}$$

where $h_n = (\vec{p}_m, \mathbf{P}_m, t_m, \tau_{now})$ is new hypothesis created fro the measurement m. The full procedure is summarized in Algorithm 1.

This procedure can be improved by evaluating whether the hypothesis should or should not be present in the FoV of the thermal camera. If a hypothesis should be visible and it is not, there is either a new obstacle blocking the view or the fire has been extuinguished already. If no hypothesis should be visible and there is a fire detected, a new hypothesis should be formed. Checking whether a hypothesis should be visible can be done geometrically (from known parameters of the camera) together with 3D ray tracing, similarly as in Section 5.3. First, a point is projected to the image plane and evaluated, whether it lies within the FoV bounds. Second, a ray from the camera center is traced towards the hypothesis position. If the ray hits an obstacle in the map before it reaches the hypothesis, the hypothesis is not visible. If the hypothesis is visible and no corresponding measurement was obtained the fire is probably extinguished and the hypothesis can be marked as obsolete or its credibility can be reduced.

Algorithm 1 Fire tracking

```
Input:
    $\mathcal{F}$        ▷ set of fire hypotheses $h_i = (\vec{p}_{h_i}, \mathbf{P}_{h_i}, t_{h_i}, \tau_{h_i})$ from previous run
    $m$        ▷ new measurement $m = (\vec{p}_m, \mathbf{P}_m, t_m)$
    $\sigma_{dir}$        ▷ standard deviation of fire direction measurement in radians
    $\sigma_{dist}$        ▷ standard deviation of fire distance measurement in meters
    $t_u$        ▷ measurement-hypothesis distance threshold in meters
    $t_l$        ▷ measurement-hypothesis likelihood threshold
Output:
    $\mathcal{F}$        ▷ set of fire hypotheses $h_i$ updated by the measurement $m$
$d = \|\vec{p}_m\|$        ▷ camera-fire distance
$\vec{f}_r = \frac{\vec{p}_m}{d}$        ▷ unit fire direction vector
$\sigma_n = 2d \tan\left(\frac{\sigma_{dir}}{2}\right)$        ▷ standard deviation of fire in direction perpendicular to $\vec{f}_r$
$\mathbf{P}_m^{||} = \mathrm{diag}(\sigma_n^2, \sigma_n^2, \sigma_{dist}^2)$        ▷ measurement covariance matrix in the fire direction-based base
$\vec{n}_1 = \mathrm{perpendicular}(\vec{f}_r)$        ▷ unit vector perpendicular to $\vec{f}_r$
$\vec{n}_2 = \vec{n}_1 \times \vec{f}_r$
$\mathbf{P}_m = \begin{bmatrix} \vec{n}_1 & \vec{n}_2 & \vec{f}_r \end{bmatrix} \mathbf{P}_m^{||}$        ▷ measurement covariance matrix
for $i = 1, \ldots, |\mathcal{P}|$ do
    $\mathbf{S}_i = \mathbf{P}_m + \mathbf{P}_{h_i}$
    $l_i = -\frac{1}{2}\left((\vec{p}_m - \vec{p}_{h_i})^T \mathbf{S}_i^{-1} (\vec{p}_m - \vec{p}_{h_i}) + \log|\mathbf{S}_i| + 3\log 2\pi\right)$
end for
$c = \arg\max_{i=1,\ldots,|\mathcal{P}|} l_i$        ▷ obtain max-likelihood hypothesis
if $l_c > t_l$ and $\|\vec{p}_c - \vec{p}_m\| < t_u$ then
    $\mathbf{K} = \mathbf{P}_c \mathbf{S}_c^{-1}$
    $\vec{p}_c \leftarrow \vec{p}_c + \mathbf{K}\left(\vec{p}_m - \vec{p}_c\right)$        ▷ update position of hypothesis $c$
    $\mathbf{P}_c \leftarrow \left(\mathbf{I} - \mathbf{K}\right)\mathbf{P}_c$        ▷ update covariance of hypothesis $c$
    $t_c \leftarrow t_m$        ▷ update temperature of hypothesis $c$
else
    $h_n = (\vec{p}_m, \mathbf{P}_m, t_m, \tau_{now})$        ▷ create new hypothesis $n$
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{h_n\}$
end if
```

Chapter 5: Mission planner

Contents

The mission planner takes care of all high-level planning of UAV motions and it specifies partial goals that the UAV should visit to inspect the whole building surface. This task consists of locating the building to be searched, segmenting its walls and other surfaces, generating viewpoints under which the wall surfaces are covered by onboard sensors, and planning the paths and UAV trajectories to optimally visit the viewpoints. All these processes are discussed individually in the following sections.

5.1 Locating the building

Primary inputs to the mission planner is a volumetric representation of the local environment and the pose of the UAV $\vec{p}_{uav} \in \mathbb{R}^3$. The volumetric map is represented by OctoMap [42], which consists of cubed cells with defined size, cell $\mathcal{M}[x, y, z] = \mathcal{M}[\vec{p}] \in \{\mathfrak{o}, \mathfrak{f}, \mathfrak{u}\}, \forall x, y, z \in \mathbb{R}$ with its center on position $\vec{p} = [x\, y\, z]^T$ can contain one of three values: occupied $\mathfrak{o}$, free $\mathfrak{f}$ and unknown $\mathfrak{u}$. The cell-based map discretize space which allows easy planning of the UAV motion in the map, marking of parts of the map, and segmentation of the object in the map.

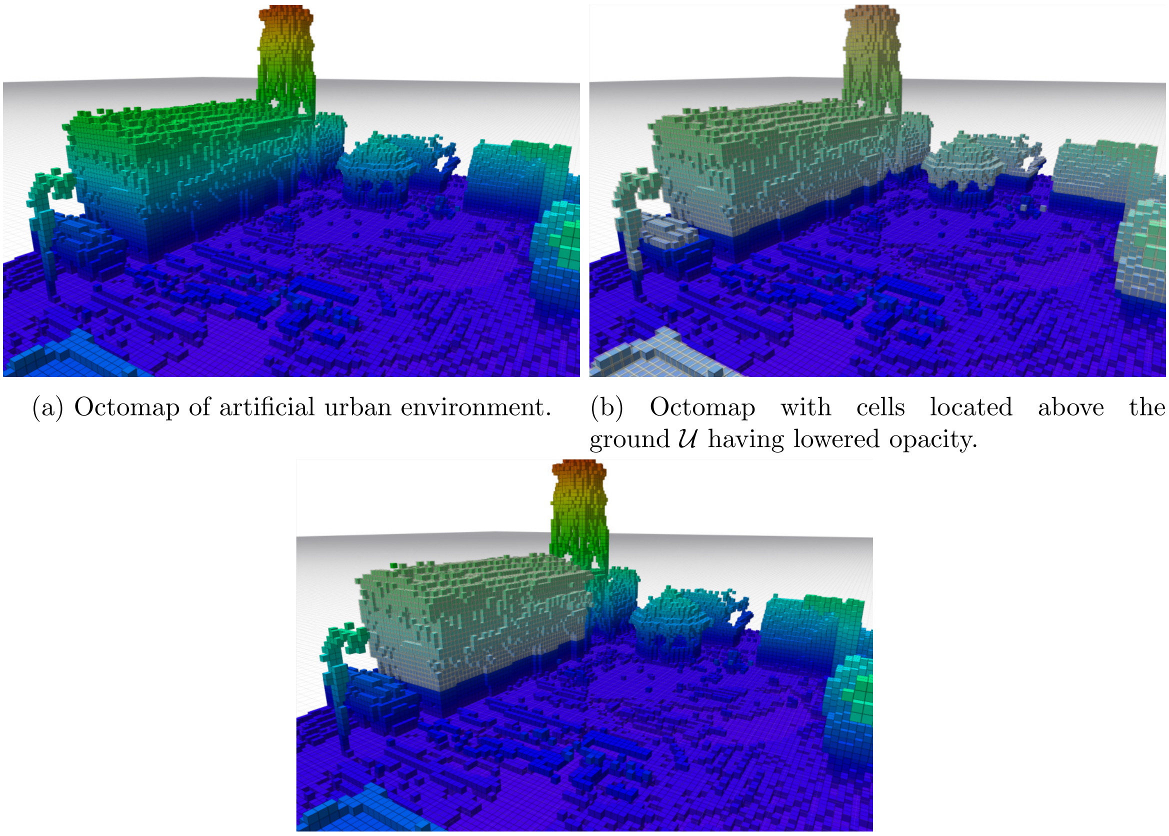
Locating the building then equals the problem of determining which map cells represent the building. Once these cells are determined, they have to be dynamically segmented from the ambient background on the run during the flight. The building-cell determination can be unclear in case when the building is connected to other buildings by a common wall or when two buildings are close. In such a case, the connected building should be also searched for fires as the fire can pervade between them. However, this is out of scope of this thesis and hence this thesis assumes no connection among two buildings.

Segmentation of the building cells from ambient map is difficult because there are no difference between occupied cell belongs to the searched building and occupied cell belonging to other object. As Figure 5.1a shows the only property allowing determination of the building its mutual positions of its cells. These cells belonging to one building generally forms large cluster of occupied cells. As the main usage of a UAV is to help with a firefight in multi-floor building assumption that searched building forms high cluster of cells can be done. To separate these cells from other is assumed that the below part of the building can be completely ignored based on fact that the UAV will be used on the upper floors (second and higher) of the building. This assumption helps with the distinction between the building and nearby objects located on the ground connecting them all to one cluster. Because the lower part of the map containing the ground can now be omitted. Hence only separated clusters of occupied cells representing individual objects will stay on the above-ground part of map. The above-ground part of map is defined by centers of its occupied cells as $\mathcal{U} \subset \mathbb{R}^3$ where

$$\mathcal{U} = \left\{ [x\, y\, z]^T \mid \mathcal{M}[x, y, z] = \mathfrak{o}, z > t_h, \forall x, y, z \in \mathbb{R} \right\}. \tag{5.1}$$

In Equation 5.1, t_h (m) is the height threshold corresponding to the height of one floor, which is commonly about 3 m.

Set $\mathcal{U}$, shown in Figure 5.1b, contains all the cells located above the ground. This set clearly contains at least one distinct cluster of cells. These clusters represent buildings and other objects above the ground. Locating the building to be searched thus equals to the problem of clustering this set and classifying a single cluster as the search area.



(a) Octomap of artificial urban environment.

(b) Octomap with cells located above the ground $\mathcal{U}$ having lowered opacity.

(c) Octomap with segmented building $\mathcal{B}$ (cells in $\mathcal{B}$ have lowered opacity).

Figure 5.1: Visualization of building segmentation process.

However, these processes can be simplified if a prior position of the building is known. Our solution obtains this base position $\vec{p}_b \in \mathbb{R}^3$ by one of the following two methods — the global lateral position is obtained from traffic maps (e.g., GNSS) or the position is interactively specified by a human operator if it is present in the map before the takeoff of the UAV. In the latter case, the base position is chosen as the UAV position before taking off $\vec{p}_b = \vec{p}_{uav}$ assuming the UAV starts closer to the proper building than another object. This assumption can be ensured by firefighters before mission starts.

Once the base position is known, a single cluster of cells $\mathcal{B} \subseteq \mathcal{U}$ can be selected from $\mathcal{U}$. The selection process starts by finding the occupied cell closest to the base position, such as

$$\vec{p}_s = \underset{\vec{x} \in \mathcal{U}}{\arg\min} \left\| \vec{x} - \vec{p}_b \right\| . \tag{5.2}$$

Having the cell $\vec{p}_s$ allows to utilize the flood-fill algorithm [62] to find the cluster $\mathcal{B}$ shown in Figure 5.1c.

5.2 Wall segmentation

To generate viewpoints under which a thermal camera inspects the building, the building has to be converted to a parametric surface representation allowing for quantifying the inspection state. The proposed solution exploits typical characteristic of urban buildings, which are built from large planar surfaces (walls). The set of building cells $\mathcal{B}$ is segmented into N planar segments, each one representing a single wall. One planar segment, the closest to the UAV, is used for the generation of viewpoints. Roofs generally also contain planar segments which can be explored. Because of that, roofs are also treated as walls with higher slopes given by the normal vector of its planar segment. Horizontal roofs are ignored since a rigidly-mounted front-facing thermal camera, assumed for the inspection task, is unable to capture the surfaces below the UAV.

The plane $\mathcal{W} \subset \mathbb{R}^3$ representing one wall of the building have plenty of parametric descriptions which completely determines it. Between the most common belongs description by triplet of points, by equation, and by normal and on-lying point. For our purposes usage of the description defined by normal vector $\vec{n} \in \mathbb{R}^3$ and point $\vec{p} \in \mathbb{R}^3$ laying on the plane will be benefiting. In this description, plane is defined by a set of points

$$\mathcal{W} = \left\{ \vec{x} \,|\, \vec{x}^T \vec{n} = \vec{p}^T \vec{n}, \vec{x} \in \mathbb{R}^3 \right\} . \tag{5.3}$$

Obtaining parametric description of planar surface $\mathcal{W}$ from set of cells $\mathcal{B}$ is complex problem regarding to possible complex shape of the building, limited resolution of the map, and just daintily differences between cells belonging to wall and cells that not. This problem is normally solved by so called model fitting procedure while parameters describing the plane are optimized in such way that plane fits (contains or is reasonable close to) a subset of points in the given set of points. This procedure always find the biggest planar segment until less than three points are present in the given set of points. Therefore repeating of the model fitting procedure and removal of points laying in close proximity to the segmented plane from the input set of points is good practice [63, 64].

This process is very efficient for many purposes, but assumes dense point cloud as its input. Using it on cells from map with limited resolution and some noise results in awkward segmentation which outputted planes greatly depends on order of the segmentation and removing points from the set. Therefore other method able to segment the needed plane segment was designed and used.

Segmentation of single plane from the set of building's cells $\mathcal{B}$ is done be improved flood-fill procedure [62]. This procedure need to be initialized in a cell $\vec{c}_0 \in \mathcal{B}$ with an initial estimate of the segmented wall normal vector $\vec{n}_{v\,0} \in \mathbb{R}^3$. The wall normal vector $\vec{n}_{v\,0}$ is specified on the fly and compared with normal vectors of cells neighboring with $\vec{c}_0$. Cells with normal vector direction similar to the direction of the wall normal vector estimate $\vec{n}_{w,0}$ are assumed as part of the segmented wall and theirs neighboring cells are also explored. This method segments single wall containing the cell $\vec{c}_0$ from the building $\mathcal{B}$. To allow normal comparison each cell $\vec{c}_i \in \mathcal{B}$ is equipped by its normal vector $\vec{n}_i \in \mathbb{R}^3$ estimated by marching cubes surface reconstruction algorithm [65] provided by the OctoMap library.

The algorithm iterate through set $\mathcal{N} \subset \mathcal{B}$ of cells neighboring to actually iterated cell $\vec{c}_i \in \mathcal{B}$. This set is given as

$$\mathcal{N} = \mathcal{N}_i \cup \{\vec{x} \,|\, \|\vec{x} - \vec{c}_i\| \leq t_c, \vec{x} \notin \mathcal{W}_i, \vec{x} \in \mathcal{B}\}\,, \tag{5.4}$$

where the $\mathcal{N}_i \subset \mathcal{B}$ is set of neighboring cells from previous iteration of the algorithm, the $\mathcal{W}_i \subset \mathcal{B}$ is set of cells identified as part of the segmented wall in previous iteration and t_c (m) is the distance threshold between neighboring cells. This threshold is $t_c >= a$, where a (m) is the cell size of the octomap representation. When $t_c > a$, wall cells can be skipped during the traversing of neighbors. In the proposed solution, $t_c = 2.5a$ which provides ability to ignore missing neighboring cell or neighboring cell with poorly estimated normal and continue traversing. Before first iteration where $i = 0$, initialization of the sets $\mathcal{N}_0 = \emptyset, \mathcal{W}_0 = \emptyset$ is done.

For each of the traversed cells $\vec{c}_i$ and its normal vector $\vec{n}_i$ direction of the normal $\vec{n}_i$ is compared with direction of the previous wall normal vector estimate $\vec{n}_{n,i}$ by

$$s_i = \vec{n}_i^T \vec{n}_{w,i}. \tag{5.5}$$

Equation 5.5 represents angular similarity of normal vector $\vec{n}_i$ of cell $\vec{c}_i$ with wall parametrized by $\vec{n}_{w,i}$. If angular similarity s_i is bigger than threshold t_a (-) this cell is assumed as part of the wall

$$\mathcal{W}_{i+1} = \mathcal{W}_i \cup \{\vec{c}_i\}, \tag{5.6}$$

else the set of the wall cells stays the same

$$\mathcal{W}_{i+1} = \mathcal{W}_i. \tag{5.7}$$

If the cell $\vec{c}_i$ is added to the wall the estimate of the wall normal vector $\vec{n}_{w,i}$ is updated by the normal vector of the added cell $\vec{n}_i$ using

$$\vec{n}_{w,i+1} = 0.9\vec{n}_{w,i} + 0.1\vec{n}_i. \tag{5.8}$$

This update represents linear combination of the two vectors in specific ratio this ratio was empirically established. Otherwise the wall normal vector estimate stays unchanged

$$\vec{n}_{w,i+1} = \vec{n}_{w,i}. \tag{5.9}$$

Regardless of whether the cell $\vec{c}_i$ is added to the wall $\mathcal{W}_{i+1}$ or not the cell $\vec{c}_i$ is removed the set of neighbors

$$\mathcal{N}_{i+1} = \mathcal{N} / \left\{ \vec{c}_i \right\}. \tag{5.10}$$

The algorithm continue by next iteration with another cell $\vec{c}_{i+1} \in \mathcal{N}_{i+1}$ selected from the actual set of neighbors.

The algorithm ends after n iterations when all neighbors are traversed and $\mathcal{N}_n$ is empty. Then a center of the wall can be estimated as

$$\vec{w}_c = \frac{1}{|\mathcal{W}_n|} \sum_{\vec{c}_i \in \mathcal{W}_n} \vec{c}_i. \tag{5.11}$$

Normal vector of the wall is specified as

$$\vec{n}_w = \frac{1}{|\mathcal{W}_n|} \sum_{i \in \{x \,|\, \vec{c}_x \in \mathcal{W}_n, x \in \mathbb{N}\}} \vec{n}_i. \tag{5.12}$$

The wall center $\vec{w}_c$ and its normal $\vec{n}_w$ specifies plane $\mathcal{W} \subset \mathbb{R}^3$ treated as the explicit expression of the segmented wall

$$\mathcal{W} = \left\{ \vec{x} \,|\, \vec{x}^T \vec{n}_w = \vec{w}_c^T \vec{n}_w, \vec{x} \in \mathbb{R}^3 \right\}. \tag{5.13}$$

This method of wall segmentation performs well in a simulated environment, but it is very sensitive for initialization of the estimate of the wall normal vector $\vec{n}_{v\,0}$ and starting position $\vec{c}_0$, therefore good initialization must be done before the algorithm starts. Selecting the cell closest to the UAV form the building $\mathcal{B}$ as $\vec{c}_0$ with its estimated normal as $\vec{n}_{v\,0}$ does not provide sufficient results. This initialization cases very unstable output of the procedure, as the UAV moves the initial cell $\vec{c}_0$ and initial normal vector $\vec{n}_{v\,0}$ changes dependent of map and normal estimation quality. This leads to outputting highly different wall representation each time the method runs.

To provide sufficient initialization, more than one cell must be used to provide some robustness towards noise in normal vector estimation. To do so, all cells in a certain distance t_s (m) to the UAV are used. Instead of starting the algorithm with just starting position $\vec{c}_0$, it is started with a non-empty set of unexplored neighbors $\mathcal{N}_0$. This set is filled by the cells in a certain distance from the UAV itself

$$\mathcal{N}_0 = \left\{ \vec{x} \,|\, \|\vec{x} - \vec{p}_{uav}\| \leq t_s, \vec{x} \in \mathcal{B} \right\}, \tag{5.14}$$

where $\vec{p}_{uav}$ is the position of the UAV in the map coordination system.

After the selection of of $\mathcal{N}_0$, the average normal vector $\vec{n}_a$ can be is counted to initialize $n_{w,0}$ as

$$\vec{n}_a = \frac{1}{|\mathcal{N}_0|} \sum_{i \in \{x \,|\, \vec{c}_x \in \mathcal{N}_0, x \in \mathbb{N}\}} \vec{n}_i, \tag{5.15}$$

Simple averaging does not provide sufficient results since the wall representation in the map is usually thicker than one cell and cells on the opposite sides of the wall representation have opposite directions of normal vectors which will cancel each other while averaging. In other words, random noise in cells' normals has major impact on the average. This is highlighted in Figure 5.2, which shows a comparison between the normal averaging approach and the method utilizing position of the UAV.

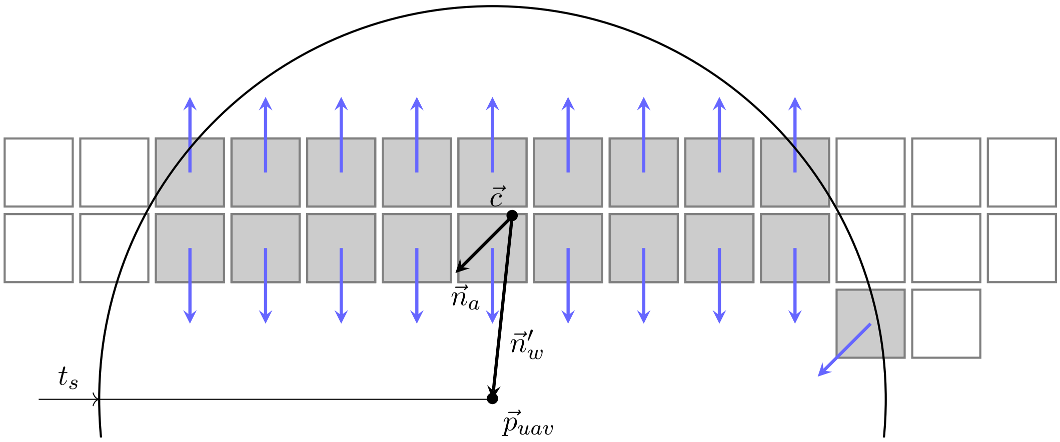


Figure 5.2: Estimation of wall normal vector $\vec{n}'_w$ using cells center $\vec{c}$ within distance t_s from the UAV compared to average normal $\vec{n}_a$.

The selected cells in $\mathcal{N}_0$ lie in a sphere defined by the UAV position and radius t_s, but because the UAV will always be outside of the building in front of the wall the cells always occupy just part of the sphere. The occupied part of the sphere specifies the direction from the UAV to the wall. This direction corresponds to the opposite of the wall normal vector. This idea is also shown in Figure 5.2 where the geometric center of the selected cells

$$\vec{c} = \frac{1}{|\mathcal{N}_0|} \sum_{i \in \{x \,|\, \vec{c}_x \in \mathcal{N}_0, x \in \mathbb{N}\}} \vec{c}_i, \tag{5.16}$$

is used to specify direction of the wall normal

$$\vec{n}'_w = \vec{p}_{uav} - \vec{c} \tag{5.17}$$

which is normalized to obtain the estimate of the wall normal vector estimate

$$\vec{n}'_{w,0} = \frac{\vec{n}'_w}{\|\vec{n}'_w\|}. \tag{5.18}$$

This initialization overcomes all previous problems, however filtration step which combines the estimate of wall normal $\vec{n}'_w$ with wall normal obtained in the previous run of the segmentation method $\vec{n}^{prev}_w$, is performed by

$$\vec{n}_{w,0} = 0.9\vec{n}'_{w,0} + 0.1\vec{n}^{prev}_w \tag{5.19}$$

to improve robustness. This allows the creation of a robust wall segmentation procedure described by Algorithm 2 and used for wall determination in the proposed design.

Algorithm 2 Wall segmentation

```
Input:
    $\mathcal{B}$    ▷ set of cells belonging to the building
    $\vec{p}_{uav}$    ▷ UAV position
    $\vec{n}_w^{prev}$    ▷ wall normal estimate $\vec{n}_w$ from previous run of the algorithm
    $t_s$    ▷ normal estimation distance in meters
    $t_c$    ▷ distance between neighboring cells in meters
    $t_a$    ▷ normal direction similarity threshold
Output:
    $\vec{w}_c$    ▷ wall center
    $\vec{n}_w$    ▷ wall normal vector
$\mathcal{N}_0 = \{\vec{x} \mid \|\vec{x} - p_{uav}\| \leq t_s, \vec{x} \in \mathcal{B}\}$    ▷ get cells in a certain distance from UAV
$\vec{c} = \frac{1}{|\mathcal{N}_0|} \sum_{\vec{x} \in \mathcal{N}_0} \vec{x}$    ▷ count center $\vec{c}$ of cells in $\mathcal{N}_0$
$\vec{n}'_w = \frac{\vec{p}_{uav} - \vec{c}}{\|\vec{p}_{uav} - \vec{c}\|}$    ▷ estimate of wall normal vector
$\vec{n}_{w,0} = 0.9\vec{n}'_w + 0.1\vec{n}_w^{prev}$    ▷ filtration step
$\mathcal{W}_0 = \emptyset$    ▷ set of cells belonging to the wall
$\mathcal{E}_0 = \emptyset$    ▷ set of explored cells
$i = 0$
while $\mathcal{N}_i \neq \emptyset$ do    ▷ traversing through neighboring cells
    $c_i = \mathcal{N}_i$.peek()    ▷ select cell from the set of neighbors
    $\vec{n}_i = \text{estimateNormal}(\mathcal{B}, \vec{c}_i)$    ▷ estimate cell normal
    $s_i = \vec{n}_i^T \vec{n}_{w,i}$
    if $s_i > t_a$ then    ▷ if cell normal has similar direction as wall normal estimate $\vec{n}_{w,i}$
        $\mathcal{W}_{i+1} = \mathcal{W}_i \cup \{\vec{c}_i\}$    ▷ add cell to wall set $\mathcal{W}$
        $\vec{n}_{w,i+1} = 0.9\vec{n}_{w,i} + 0.1\vec{n}_i$    ▷ update estimate of wall normal $\vec{n}_w$
    else
        $\mathcal{W}_{i+1} = \mathcal{W}_i$
        $\vec{n}_{w,i+1} = \vec{n}_{w,i}$
    end if
    $\mathcal{E}_{i+1} = \mathcal{E}_i \cup \{\vec{c}_i\}$    ▷ mark cell as explored to prevent endless loop
    $\mathcal{N} = \mathcal{N}_i \cup \{\vec{x} \mid \|\vec{x} - \vec{c}_i\| \leq t_c, \vec{x} \notin \mathcal{E}_i, \vec{x} \notin \mathcal{W}_i, \vec{x} \in \mathcal{B}\}$    ▷ add neighbors of $\vec{c}_i$ to set $\mathcal{N}$
    $\mathcal{N}_{i+1} = \mathcal{N} / \{\vec{c}_i\}$    ▷ remove cell from $\mathcal{N}$
    $i \leftarrow i + 1$
end while
$\vec{w}_c = \frac{1}{|\mathcal{W}_i|} \sum_{\vec{x} \in \mathcal{W}_i} \vec{x}$    ▷ count precise wall center $\vec{w}_c$
$\vec{n}_w = \frac{1}{|\mathcal{W}_i|} \sum_{j \in \{x \mid \vec{c}_x \in \mathcal{W}_i, x \in \mathbb{N}\}} \vec{n}_j$    ▷ count precise wall normal $\vec{n}_w$
```

5.3 Marking of seen cells

While planning the inspection path to an unknown part of the building, the distinction between an already inspected and unchecked part of the building must be done. Therefore every time a new thermal image arrives from the thermal camera occupied map cells that are in the field of view of the camera are marked as seen or inspected. These cells are elided from the next planing.

The process of marking seen cells utilizes tracing rays from the camera to the first occupied cell in the map. Position of the center of this cell $\vec{p}_r \in \mathbb{R}^3$ is then added to a group of seen cells $\mathcal{S} \subset \mathbb{R}^3$. The OctoMap library provides implementation of he ray tracing `computeRayKeys` function [42]. This function allows iterating through map cells laying on a line $\mathcal{Y}$ defined by a starting point $\vec{p}_c \in \mathbb{R}^4$ and direction vector $\vec{t} \in \mathbb{R}^4$. The cells on the line are iterated from the cell containing starting point $\vec{p}_c$ in direction $\vec{t}$ until an occupied cell is found or a maximal ray distance t_d (m) is reached. If the occupied cell is found, the center of that cell $\vec{p}_r$ is added to the seen-cells set $\mathcal{S}$.

The line $\mathcal{Y}$ can be obtained for each thermal camera pixel by applying a pinhole camera model [66]. The pinhole camera model specifies the projection of a real 3D point $\vec{x}^t$ in camera coordination system $\mathcal{O}^t$ to image plane 2D point $\vec{x}^I$ in image coordination system $\mathcal{O}^I$. Homogeneous coordinates are used for this projection, therefore $\vec{x}^t \in \mathbb{R}^4$ and $\vec{x}^I \in \mathbb{R}^3$. The projection is specified as

$$\vec{x}^I = \mathbf{C}\vec{x}^t, \tag{5.20}$$

where $\mathbf{C}$ is a projection matrix defined by real camera geometry as

$$\mathbf{C} = \begin{bmatrix} f_l & 0 & 0 & c_x \\ 0 & f_l & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{5.21}$$

where f_l is the focal length of the camera in pixels and

$$c_x = \frac{I_w}{2},\ c_y = \frac{I_h}{2} \tag{5.22}$$

specify the position of the camera optical center in the image plane and I_w respective I_h are image dimensions in pixels. The situation is also visualized in Figure 5.3.

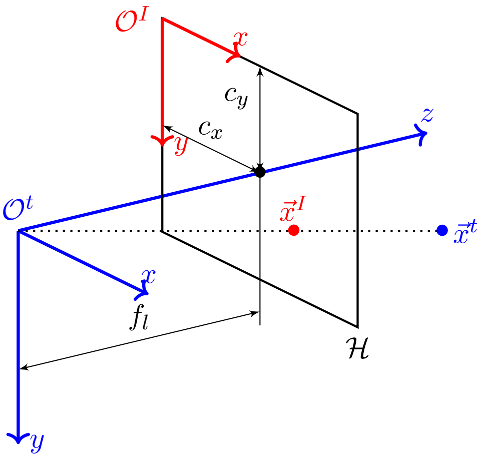


Figure 5.3: Coordination system of a camera $\mathcal{O}^t$ and its image plane $\mathcal{H}$ with coordination system $\mathcal{O}^I$.

The line $\mathcal{Y}$ is then solution of Equation 5.20 for the selected pixel $\vec{x}^I$ with unknown $\vec{x}^t$ which can be written as

$$\mathcal{Y} = \left\{ \vec{x}^t \,|\, \mathbf{C}x^t = \vec{x}^I, \vec{x}^t \in \mathbb{R}^4 \right\}. \tag{5.23}$$

The line is given in the camera coordination frame. To obtain the solution expressed in map coordination frame $\mathcal{O}$ transformation matrix from camera frame to map frame, a transformation matrix $\mathbf{T}_t \in \mathbb{R}^{4\times4}$ is used. This matrix represents the mutual position and orientation between the map and camera coordination systems. It is obvious that this transformation depends on the actual position and orientation of the UAV in the map, therefore this matrix is obtained from the MRS UAV system which takes care of the localization of the UAV and provides actual transformation matrices between various coordinate systems. The line $\mathcal{Y}$ can be expressed also as

$$\mathcal{Y} = \left\{ a\vec{t^t} \mid [0\,0\,1\,0]\vec{t^t} > 0, a \in \mathbb{R} \right\}, \tag{5.24}$$

where $\vec{t^t} \in \mathbb{R}^4$ is the direction vector of the line in thermal camera coordination frame $\mathcal{O}^t$. In the implemented solution is this vector obtained by calling `image_geometry` package function `projectPixelTo3dRay`. The direction vector $\vec{t}$ can be expressed in the map coordination frame as

$$\vec{t} = \mathbf{T}_t\vec{t^t} - \vec{p}_c, \tag{5.25}$$

where p_c is the origin of the camera coordination frame expressed in the map coordination frame as

$$\vec{p}_c = \mathbf{T}_t\vec{o}. \tag{5.26}$$

This process can be repeated for each pixel $\vec{x}^I$ of the thermal image but for its computation complexity is advantageous to repeat this process only for every n-th pixel. The limited length of the traced ray with the limited resolution of the map enables this computation facilitation without losing precision for a reasonable amount of skipped pixels $p_s \in \mathbb{N}$. The whole process of marking seen cells is summarized in Algorithm 3.

Algorithm 3 Marking of explored cells

```
Input:
    $\mathcal{S}$                                ▷ group of seen cells during previous exploration
    $\mathcal{M}$                                ▷ actual map of ambient environment
    $\mathbf{T}_t$                               ▷ transformation matrix between camera and map
    $\mathbf{C}$                                 ▷ camera projection matrix
    $t_d$                                        ▷ ray length threshold in meters
    $I_h$                                        ▷ thermal image height in pixels
    $I_w$                                        ▷ thermal image width in pixels
    $p_s$                                        ▷ step size in pixels
Output:
    $\mathcal{S}$                                ▷ group of seen cells completed by cells explored in actual step
$\vec{p_c} = \mathbf{T}_t \vec{o}$               ▷ gets camera center in map coordination frame
for $y = 0, \dots, \frac{I_h}{p_s}$ do
    for $x = 0, \dots, \frac{I_w}{p_s}$ do
        $\vec{x}^I_{x\,y} = [xp_s \, yp_s \, 1]^T$                        ▷ pixel position in the image
        $\vec{t^t}_{x\,y} = \text{projectPixelTo3dRay}(\mathbf{C}, \vec{x}^I_{x\,y})$
        $\vec{t}_{x\,y} = \mathbf{T}_t \vec{t^t}_{x\,y} - \vec{p_c}$      ▷ express direction vector $t$ in map coordination frame
        $\mathcal{R}_{x\,y} = \text{computeRayKeys}(\mathcal{M}, \vec{p_c}, \vec{t}_{x\,y}, t_d)$
        for $c \in \mathcal{R}_{x\,y}$ do
            if $c = \mathtt{o}$ then                             ▷ if occupied cell is found
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{\vec{p_r}\}$   ▷ add cell $\vec{p_r}$ to group of seen cells $\mathcal{S}$
                break                                            ▷ break the inner for loop
            end if
        end for
    end for
end for
```

5.4 Viewpoints generation

A set of viewpoints (VPs) to be visited by a UAV is generated based on segmented wall plane $\mathcal{W}$ defined by its normal vector $\vec{n}_w$ and point $\vec{w}_c$ laying on the plane, set of points belonging to inspected building $\mathcal{B}$, and set of already inspected points $\mathcal{S}$. Viewpoint consists of position $\vec{c}_p \in \mathbb{R}^3$ of camera center (origin of camera coordination system $\mathcal{O}^t$) and direction of the Z-axis of the camera coordination system $\mathcal{O}^t$ (camera optical axis) specified by direction vector $\vec{c}_d \in \mathbb{R}^3$. Both $\vec{c}_p$ and $\vec{c}_d$ are expressed in map coordination system $\mathcal{O}$. The viewpoint specifies the intended camera position and orientation to inspect specific parts of the wall. As the thermal camera is mounted in a fixed manner on the UAV platform, it can not rotate to change the vertical slope of its optical axis. In this thesis, a horizontally-mounted camera is assumed, which allows for inspection of vertical surfaces only. Therefore highly sloped surfaces and horizontal roofs are ignored in this design, but they could be easily added by a generalization of the process of VPs generation described below.

Decision to ignore a sloped surface is based on the normal vector of the surface $\vec{n}_w$.

Surfaces fulfilling criteria

$$\vec{n}_w^T \vec{z} > 0.5 \tag{5.27}$$

where $\vec{z} = [0\,0\,1]^T$, are ignored. Walls that are not omitted are verticalized by the creation of a new vertical plane specified by the same center point $\vec{w}_c$ and new horizontal normal vector $\vec{n}_v \in \mathbb{R}^3$. The normal vector is created as

$$\vec{n}_v = \frac{\mathbf{A}\vec{n}_w}{\|\mathbf{A}\vec{n}_w\|}, \tag{5.28}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.29}$$

For efficient distinction between inspected and unseen parts of the wall, the wall must be divided into discrete objects holding the information. In the proposed design, cells from $\mathcal{B}$ close to the plane $\mathcal{W}$ are projected to the vertical plane and the projected points are henceforth called points of interest. A set of these points of interest is given as

$$\mathcal{I}' = \left\{ \vec{x} + (-\vec{n}_v^T \vec{x} + \vec{n}_v^T \vec{w}_c)\vec{n}_v \,|\, |\vec{n}_w^T \vec{x} - \vec{n}_v^T \vec{w}_c| < t_w, \vec{x} \in \mathcal{B} \right\}, \tag{5.30}$$

where t_w (m) is distance threshold for assumption of cell $\vec{x}$ in $\mathcal{B}$ belonging to wall $\mathcal{W}$. As already inspected cells in $\mathcal{S}$ are not interesting for planning, they are projected to the vertical wall plane by

$$\mathcal{S}' = \left\{ \vec{x} + (-\vec{n}_v^T \vec{x} + \vec{n}_v^T \vec{w}_c)\vec{n}_v \,|\, |\vec{n}_w^T \vec{x} - \vec{n}_v^T \vec{w}_c| < t_w, \vec{x} \in \mathcal{S} \right\} \tag{5.31}$$

and points of interest close to any of these projected cells are removed from $\mathcal{I}'$. The final set of points to be explored is

$$\mathcal{I} = \left\{ \vec{x} \,|\, \min_{\vec{y} \in \mathcal{S}'} \|\vec{x} - \vec{y}\| > t_r, \vec{x} \in \mathcal{I}' \right\}, \tag{5.32}$$

where t_r (m) is the distance threshold between a point of interest and its closest projected seen cell. Comparison of the distance $\|\vec{x} - \vec{y}\|, \vec{x} \in \mathcal{I}', \vec{y} \in \mathcal{S}'$ and the threshold t_r decide about removing the point of interest $\vec{x}$. If any projected seen cell is close to point of interest, this point is assumed as already inspected and therefore is no longer interesting.

For efficient work with the points of interest$\mathcal{I}$, a 2D coordination system in the vertical plane is established. Base $\mathcal{O}_w$ of this coordination system consists of the vertical vector $\vec{z}$ and vector $\vec{u} \in \mathbb{R}^3$ perpendicular to both $\vec{z}$ and $\vec{n}_v$. This vector is created as

$$\vec{u} = \frac{\vec{z} \times \vec{n}_v}{\|\vec{z} \times \vec{n}_v\|}. \tag{5.33}$$

To transform the coordination of points of interest to this base, a matrix

$$\mathbf{P} = \begin{bmatrix} \vec{u}^T \\ \vec{z}^T \end{bmatrix} \tag{5.34}$$

is used. The transformed points form set

$$\mathcal{I}_p = \left\{ \mathbf{P}\vec{x} \,|\, \vec{x} \in \mathcal{I} \right\}. \tag{5.35}$$

Assume an ideal case when the thermal camera optical axis is perpendicular to the vertical wall. In other words, the camera axis is collinear with the wall normal vector $\vec{n}_v$. In that case, the camera sees a rectangular part of the wall. The seen part of the wall can be specified by its height h (m), width w (m), and its center $\vec{r}_c \in \mathbb{R}^2$ in the base $\mathcal{O}_w$. These parameters of the seen rectangle correspond to the used camera lens and camera distance from the wall w_d (m). Using the pinhole camera model idea shown in Figure 5.3, the dimensions of the rectangle can be designed as

$$h = \frac{w_d}{f_l} I_h \tag{5.36}$$

and

$$w = \frac{w_d}{f_l} I_w, \tag{5.37}$$

where f_l is the camera focal length in pixels, I_w and I_h are thermal image width and height in pixels. The center of the rectangle is then projection of camera position $\vec{p}_c$ to the wall coordination system $\mathcal{O}_w$ by

$$\vec{r}_c = \mathbf{P}\vec{p}_c. \tag{5.38}$$

Viewpoints are generated from the set of points of interest $\mathcal{I}_p$ using the proportions of the seen rectangle. Firstly, a set $\mathcal{Q}_p \subset \mathbb{R}^2$ of centers of a rectangles to be seen is created. Secondly, the viewpoints are created using this set and the wall normal vector $\vec{n}_v$.

To create the set $\mathcal{Q}_p$ a simple process is done, rectangles to be seen are aligned with each other such that every contain at least one point of interest. The rectangles are lined up from the bottom left corner of the wall to horizontal lines. Neighboring rectangles overlap by a specified distance o (m) to improve robustness and cope with inaccuracy in UAV positioning. The bottom left point is found by finding minimal Z-coordination as

$$m_z = \min_{\vec{x} \in \mathcal{I}_p} [0\,1]\vec{x}. \tag{5.39}$$

This minimal Z-coordination is used to keep the lines by finding the most left point in horizontal strip $\mathcal{G}$ with the height equal to the height of seen area. The strip is given as

$$\mathcal{G} = \{\vec{x} \mid [0\,1]\vec{x} < m_z + h - o, \vec{x} \in \mathcal{I}_p\} \tag{5.40}$$

and the left point is then found as

$$\vec{m} = \operatorname*{arg\,min}_{\vec{x} \in \mathcal{G}} [1\,0]\vec{x}. \tag{5.41}$$

The point m is used to count a center of the intended seen rectangle which is then added to $\mathcal{Q}_p$ by

$$\mathcal{Q}_p \leftarrow \mathcal{Q}_p \cup \left\{ \vec{m} + \frac{[w - o\, h - o]^T}{2} \right\}. \tag{5.42}$$

The intended seen rectangle is then removed from points of interest

$$\mathcal{I}_p \leftarrow \{\vec{x} \mid [1\,0]\vec{x} > [1\,0]\vec{m} + w - o,\ [0\,1]\vec{x} > [0\,1]\vec{m} + h - o,\ \vec{x} \in \mathcal{I}_p\}. \tag{5.43}$$

This process is repeated until the set of interesting points $\mathcal{I}_p$ is empty. Points in $\mathcal{Q}_p$ are expressed in the map coordination frame as

$$\mathcal{Q} = \left\{ [\vec{u}\,\vec{z}]\vec{x} + (\vec{n}_v^T \vec{w}_c)\vec{n}_v \mid \vec{x} \in \mathcal{Q}_p \right\}. \tag{5.44}$$

As the UAV should capture the wall from intended distance w_d all points in $\mathcal{Q}$ are shifted from the wall by the distance in direction of wall normal vector $\vec{n}_v$. The set $\mathcal{V} \subset \mathbb{R}^3$ of all intended camera positions $\vec{c}_p \in \mathcal{V}$ is then given as

$$\mathcal{V} = \left\{ \vec{x} + w_d \vec{n}_v \,|\, \vec{x} \in \mathcal{Q} \right\}. \tag{5.45}$$

As the camera should be pointing towards the wall, the direction of each viewpoint is opposite to the normal vector of the wall, thus

$$\vec{c}_d = -\vec{n}_v. \tag{5.46}$$

Vector $\vec{c}_d$, together with $\mathcal{V}$, forms viewpoints used in the next steps of mission planning. This entire procedure of viewpoints generation is described in Algorithm 4.

Algorithm 4 Viewpoints generation

```
Input:
    $\mathcal{B}$    ▷ set of cells belonging to the building
    $\mathcal{S}$    ▷ group of seen cells
    $\vec{n}_w$    ▷ wall normal
    $\vec{w}_c$    ▷ wall center point
    $t_w$    ▷ distance threshold for assigning cells to walls in meters
    $t_r$    ▷ distance threshold for seen point in meters
    $w_d$    ▷ UAV-wall desired distance in meters
    $o$    ▷ overlap of viewpoint images in meters
    $f_l$    ▷ camera focal length in pixels
    $I_h$    ▷ thermal image height in pixels
    $I_w$    ▷ thermal image width in pixels
Output:
    $\mathcal{V}$    ▷ group of intended camera centers
    $\vec{c}_d$    ▷ intended camera direction
if $[0\,0\,1]\vec{n}_w > 0.5$ then    ▷ omit horizontal planes
    return
end if
$n'_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{n}_w$    ▷ make new horizontal wall normal
$\vec{n}_v = \frac{\vec{n}'_v}{\|\vec{n}'_v\|}$    ▷ normalize it
▷ project points close to the wall into it
$\mathcal{I}' = \left\{\vec{x} + (-\vec{n}_v^T\vec{x} + \vec{n}_v^T\vec{w}_c)\vec{n}_v \,|\, |\vec{n}_w^T\vec{x} - \vec{n}_v^T\vec{w}_c| < t_w, \vec{x} \in \mathcal{B}\right\}$
▷ project seen points close to the wall into it
$\mathcal{S}' = \left\{\vec{x} + (-\vec{n}_v^T\vec{x} + \vec{n}_v^T\vec{w}_c)\vec{n}_v \,|\, |\vec{n}_w^T\vec{x} - \vec{n}_v^T\vec{w}_c| < t_w, \vec{x} \in \mathcal{S}\right\}$
$\mathcal{I} = \left\{\vec{x} \,|\, \min_{\vec{y} \in \mathcal{S}'} \|\vec{x} - \vec{y}\| > t_r, \vec{x} \in \mathcal{I}'\right\}$    ▷ remove points close to seen points
$\vec{u}' = [0\,0\,1] \times \vec{n}_v$
$\vec{u} = \frac{\vec{u}'}{\|\vec{u}'\|}$
$\mathbf{P} = \begin{bmatrix} \vec{u} & \vec{z} \end{bmatrix}^T$    ▷ create new wall base
$\mathcal{I}_p = \{\mathbf{P}\vec{x} \,|\, \vec{x} \in \mathcal{I}\}$    ▷ transform $\mathcal{I}$ into the base
$h = \frac{w_d}{f_l} I_h$    ▷ seen rectangle proportions
$w = \frac{w_d}{f_l} I_w$
$\mathcal{Q}_p = \emptyset$
while $\mathcal{I}_p \neq \emptyset$ do
    $m_z = \min_{\vec{x} \in \mathcal{I}_p} [0\,1]\vec{x}$    ▷ the lowest point of the unseen wall
    $\mathcal{G} = \{\vec{x} \,|\, [0\,1]\vec{x} < m_z + 0.8(h - o), \vec{x} \in \mathcal{I}_p\}$    ▷ bottom part of the unseen wall
    $\vec{m} = \arg\min_{\vec{x} \in \mathcal{G}} [1\,0]\vec{x}$    ▷ most left bottom point
    $\mathcal{Q}_p \leftarrow \mathcal{Q}_p \cup \left\{\vec{m} + \frac{[w-o\;h-o]^T}{2}\right\}$    ▷ save center of the seen rectangle
    ▷ Remove seen rectangle from the unseen wall
    $\mathcal{I}_p \leftarrow \{\vec{x} \,|\, [1\,0]\vec{x} > [1\,0]\vec{m} + w - o,\ [0\,1]\vec{x} > [0\,1]\vec{m} + h - o,\ \vec{x} \in \mathcal{I}_p\}$
end while
$\mathcal{Q} = \left\{[\vec{u}\,\vec{z}]\vec{x} + (\vec{n}_v^T\vec{w}_c)\vec{n}_v \,|\, \vec{x} \in \mathcal{Q}_p\right\}$    ▷ transform back to map coordination frame
$\mathcal{V} = \{\vec{x} + w_d\vec{n}_v \,|\, \vec{x} \in \mathcal{Q}\}$    ▷ move in front of the wall by capturing distance
$\vec{c}_d = -\vec{n}_v$    ▷ camera direction is opposite to wall normal
```

5.5 Generation of paths

To visit the generated VP or perform other motion with the UAV the motion must be described in a way that the UAV control subsystem will understate it. Planned UAV motion can be described in multiple levels of its exactness. The first rough description of the planned motion is a plan. The plan consists of ordered points (goals) that will be visited by the UAV in the given order. The plan describes just the order of goals. A more exact description is a path that describes also the way between the particular goals concerning the given environment. That means that path already avoids static obstacles. The most exact description of the future motion is a trajectory, which provides not just spatial but also temporal information about the motion. The trajectory is generated from the path concerning UAV dynamic abilities. The trajectory can be also generated in a way that minimalism time of the motion, or energy needed for it, or another parameter, and represents a description suitable for its performance by the control subsystem.

The MRS UAV system implements the generation of trajectory from any path. But it also provides the generation of the path in two different ways. Both ways of path generation utilize efficient A* planner but in different tasks. The first task is to plan a path between two given points on the map the second is to navigate the UAV to a specific position. In this design, both tasks are used because each is suitable for another purpose. A so-called pathfinder can lead the UAV to one specified goal. The pathfinder automatically plans a collision-free path from the UAV position towards the goal in the already explored map, one it found the end of the map the UAV is led to the end of the map to explore the next part of the environment, this process is repeated till the goal is reached. This feature is suitable for approaching the building or finding new building surfaces to be explored for its ability to traverse the unknown environment. But it is not suitable for planning between multiple goals. Therefore once some wall is found and VPs are generated the other task is used to generate a collision-free path between twice of the VP.

To generate a path through all generated VP a plan specifying its order is needed. This task was already solved in [67] where Lin-Kernighan heuristic is used to solve the traveling salesman problem [68]. The problem is defined as the finding of the optimal order to visit specific paces concerning the length or time of the way. When the optimal order of partial goals is assessed paths between twice consecutive goals are planned by the A* algorithm. This procedure ends by concatenation of the paths and delegating the path to the MRS UAV system which generates trajectory from it.

5.6 New wall search

When all VP on one side of the building was explored the other side of it should be inspected as well. But this side most probably was not even scanned to map because was all the time behind the first explored side and sensors can not see it. To scan the unknown surfaces of the building the UAV shod be led to some point $\vec{p}_e \in \mathbb{R}^3$ out of the explored map or at least on another side of the building, but the point should be in proximity of the building.

To find some point that may be on another side of the building the center of an already known building is counted

$$\vec{b}_c = \frac{1}{|\mathcal{B}|} \sum_{\vec{x} \in \mathcal{B}} \vec{x}. \tag{5.47}$$

The building center is then used for the creation of a building-UAV direction vector

$$\vec{d} = \vec{p}_{uav} - \vec{b}_c, \tag{5.48}$$

where $\vec{p}_{uav}$ is actual UAV position. The cross product of this vector with the upward pointing vector $\vec{z}$ gives a vector pointing to a wall behind left seen corner of the building

$$\vec{p}_e = a\vec{d} \times \vec{z} + \vec{b}_c \tag{5.49}$$

The only remaining issue is the distance between the building center and its unknown wall can be easily estimated. Therefore the distance of the target exploring point $\vec{p}_e$ from the building center could be tuned by parameter a. The assumption that the building center has similar distances to all walls can be done in that case it should be chosen as $a = 1$, for aggressive search can be $a > 1$. In the design is chosen $a = 2$. This point $\vec{p}_e$ is given to the pathfinder which leads UAV to it. If the point is in the building or another obstacle the pathfinder will lead the UAV to the closest accessible free spot. This process can be repeated until a new wall is found.

5.7 Mission control

Search the mission planner connecting all previously described system and decides when each take part. This part also connects the searching system with other systems that are out of the scope of this thesis as the main mission control which decides when to search and when to extinguish, a subsystem for approaching the building, and a system for managing search hints given by the operator. These missing parts make its development complicated therefore just the main idea of the process is presented here.

Till the search mission is active, the search controller will repeat these steps approach the building to the distance in which the wall detector work, it means closer than threshold t_s, navigate the UAV to the top of the wall to start from the top part of the building, perform the search, and find a new wall. Performing the search means repeating segmentation of the wall closest to the UAV, generation of VPs on this wall, and navigating through them till some new viewpoints are generated. These procedures were already details and described.

Chapter 6: Simulation Verification

The designed system for the autonomous search was tested in a simulated environment provided by the MRS system. The simulation environment is modeled within the Gazebo physical simulator that serves as a robust simulation platform platform widely used in field of robotics [69]. This simulator provides close-to-reality physics. The MRS UAV system contains useful UAV models, together with on-board sensors and their functionality plugins. Adding new models to this simulation pipeline is possible, thus the designed frame was integrated into it.

Beside UAV frames, other interesting object are simulated in the artificial environment. These objects are necessary for testing UAV missions interfacing with the environment. These objects do not affect the UAV just physically but affect data from the simulated sensors. This allows testing of recognition and segmenting algorithms as well as vision based navigation. For testing of various algorithms in urban environment, a credible model of an artificial city was created by the MRS group. Model of the city is used for the simulation of real conditions in an urban environment. Figure 6.1 shows a render of the building used as mockup in the simulated scenario.

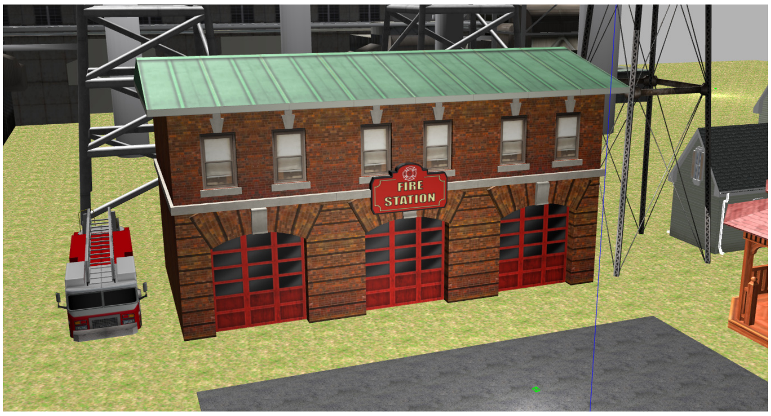


Figure 6.1: Render of building used as mockup in the simulation.

The simulated mission can be understood as maximum-coverage problem which aims to cover full full area specified by FoV of on-board thermal camera. As most interesting metric for the maximum-coverage problem is how well the surface was covered. This can be specified as the ratio between area of the seen part of the surface and area of the full specified surface. As discrete representation of the environment (the volumetric map) is available the ratio can be obtained by comparison of the total amount of seen map cells s to the number of cells on the specified surface w providing ratio

$$s_r = \frac{s}{w}, \tag{6.1}$$

used to quantitatively evaluate the performance in a single experiment. Other usable metric for comparison between multiple solution to this problem can be amount of generated viewpoints for the same surface or time spent by the search.

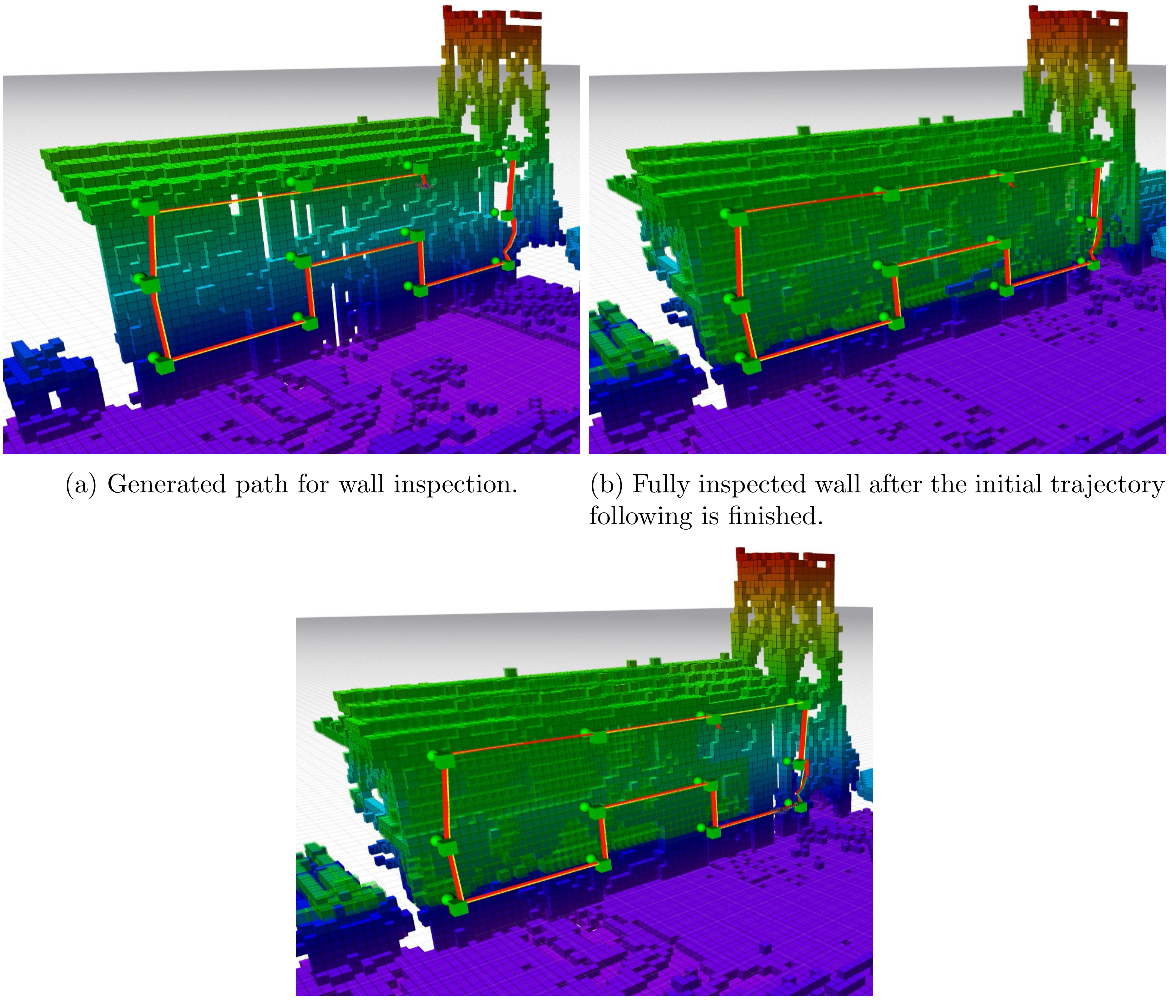
The mocap exhibition in Figure 6.2 was used for evaluating the autonomous 3D search in a simulated scenario. The simulation world presents a simple scenario where the autonomy may focus on solving the perception-constrained maximum-coverage problem with the surfaces to be inspected are planar only. During the presented experiment, the UAV detects a wall, generates a set of viewpoints covering the wall in full and plans the optimal (shortest) path via these viewpoints by solving the TSP. The generated path shown in Figure 6.2a is then delegated to the MRS UAV system, which samples the path to time-parametrized trajectory respecting dynamic constraints of the UAV. Afterwards, the trajectory is followed by the UAV with the on-board sensors looking for fires inside. While following the trajectory, the proposed algorithms store and mark cells in volumetric map, which were seen (and covered) by on-board thermal camera.

In this experiment, 98.21 % of the surface was covered by the on-board camera. The map of the environment after the trajectory is flown is shown in Figure 6.2b and total numbers of cells in Table 6.1.. The resulting 1.19 % cells missed by the camera appeared in the map only after the UAV inspected the previously-occluded parts of the surface.

Property	**Value**
Total number of cells belonging to wall w	756
Total number of cells seen by the camera s	747

Table 6.1: Simulated results.

The functionality of the wall segmentation system and viewpoint generation is essential for the functionality of the full search system. Their performance is shown in Figure 6.2 where green cubes represent positions of the viewpoints whereas the green spheres show their directions. Failure of the wall segmentation system will manifest itself in viewpoints generated in places not related to the real wall.



(a) Generated path for wall inspection.

(b) Fully inspected wall after the initial trajectory following is finished.

(c) Wall inspection in progress.

Figure 6.2: Planned surface-coverage paths together with volumetric maps build during the mission. In the map, the cells highlighted in green represent the cells inspected by the thermal camera. Yellow line highlights the generated path whereas the red line stands for the flown trajectory.

Chapter 7: Conclusion

Contents

This thesis focused on the entire scope of a complex task. The proposed solution ranges from designing and constructing of large heavy-paylod UAV platform (approximately four times heavier platform than ever built within the MRS research group) to designing and implementing algorithms for searching of indoor-burning fires and tracking their state estimated from on-board measurements. The designed and constructed hardware platform is capable of carrying six six fire-extinguishing ampules and launch them independently on each other into a fire detected and tracked with the implemented software solution. The proposed solution for autonomous search of indoor-burning fires with an outdoor-flying UAV was developed, implemented, and discussed in detail. The solution is capable of searching of fires burning in high-rise building by effectively exploring exterior surfaces of the building. These surfaces are detected dynamically mid-flight during the mission and used in replanning the vision-constrained maximum-coverage task. The entire solution is designed to work with minimum (but possible) human assistance in the loop.

Tasks given by the following list were successfully completed. Multimedia files supporting the thesis are available at `http://mrs.felk.cvut.cz/theses/nydrle2023`.

- The application-tailored design of the UAV platform capable of carrying heavy payload (up to 20 kg) and coping with dynamic recoil arising from launching ampoules from on-board mechanism is described in Chapter 3. The platform weights 41 kg in total (incld. payload) and has dimensions of 2×1.6 m (with propellers).

- A method for autonomous search for indoor-burning fires from outside of the building was designed in Chapter 5. The proposed solution utilizes preliminary algorithms for detecting and localizing fires and extends them with noise-resistant fusion of new measurements and temporal tracking of their states. This is further extended with autonomous navigation targeting perception-constrained search for indoor-burning fires with an outdoor-flying UAV.

- A methodology for tracking detected fires, estimating their state, and fusing new measurements into them was deisnged them was designed on the base of linear Kalman filter in Chapter 4.

- Both algorithms were implemented as ROS nodes interfacing with the system of Multi-Robot Systems group for stabilization and control of UAVs.

- A simulated search mission along the wall of an artificial building was performed and its performance was analyzed in Chapter 6.

All these tasks aim to create a robust semi-autonomous firefighting solution where the UAV assists firefighters in searching for fires in floors located high above the ground. The assistance may include a volumetric 3D map containing RGB and thermal aerial imaging, which a human responder may visualize dynamically and can act on their basis to provide fast, effective, and live-saving response.

7.1 Future work

To be able to perform such an ambitious assistance to human responders, the proposed solution described in Chapter 2 must be extended significantly. These extensions include a user interface between the autonomy and an operator in the loop, where the operator can interactively command and supervise the mission. The proposed algorithms also need to be tested more thoroughly and evaluated both qualitatively and quantitatively. The primary focus of these tests should be laid on robustness of these algorithms in real-world conditions, which significantly differ from the simulated ones.

Bibliography

[1] A. D. Team. (2019) Octorotor X8 configuration. Accessed on March 21, 2022. [Online]. Available: http://ardupilot.org/copter/docs/connect-escs-and-motors.html#connect-escs-and-motors3

[2] S. for All Markus Mueller. (2019) eCalc online tool (free non-commercial license). Accessed on January 27, 2022. [Online]. Available: https://www.ecalc.ch/

[3] 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, April 2020.

[4] G. Silano, J. Bednar, T. Nascimento, J. Capitan, M. Saska, and A. Ollero, "A multi-layer software architecture for aerial cognitive multi-robot systems in power line inspection tasks," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS).* IEEE, 2021, pp. 1624–1629.

[5] P. Ješke, v. Klouček, and M. Saska, "Autonomous compact monitoring of large areas using micro aerial vehicles with limited sensory information and computational resources," in *Lecture Notes in Computer Science, vol 11472.* Springer International Publishing, 2019.

[6] P. Ješke, Š. Klouček, and M. Saska, "Autonomous compact monitoring of large areas using micro aerial vehicles with limited sensory information and computational resources," in *International Conference on Modelling and Simulation for Autonomous Systesm.* Springer, 2018, pp. 158–171.

[7] M. Petrlik, P. Petracek, V. Kratky, T. Musil, Y. Stasinchuk, M. Vrba, T. Baca, D. Hert, M. Pecka, T. Svoboda, and M. Saska, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," *preprint arXiv:2206.08185*, 2022, submitted to Field Robotics Special Issue: DARPA Subterranean Challenge.

[8] J. Horyna, T. Baca, V. Walter, D. Albani, D. Hert, E. Ferrante, and M. Saska, "Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication," *Autonomous Robots*, pp. 1–17, 2022.

[9] V. Spurný, T. Báča, M. Saska, R. Pěnička, T. Krajník, J. Thomas, D. Thakur, G. Loianno, and V. Kumar, "Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles," *Journal of Field Robotics*, vol. 36, no. 1, pp. 125–148, 2019.

[10] J. Horyna, T. Baca, 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 (ICUAS).* IEEE, June 2021, pp. 1139–1147.

[11] V. Spurný, M. Petrlík, V. Vonásek, and M. Saska, “Cooperative transport of large objects by a pair of unmanned aerial systems using sampling-based motion planning,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019, pp. 955–962.

[12] G. Loianno, V. Spurny, T. Baca, J. Thomas, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar, “Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert like environments,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 1576–1583, 2018.

[13] Y. Stasinchuk, M. Vrba, M. Petrlik, T. Baca, V. Spurny, D. Hert, D. Zaitlik, T. Nascimento, and M. Saska., “A Multi-UAV System for Detection and Elimination of Multiple Targets,” in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, May 2021, pp. 555–561.

[14] BONPET LIQUID | Bonpet systems. Accessed on April 12, 2020. [Online]. Available: https://www.bonpet.si/en_US/bonpet-tekocina

[15] A. Özcan and O. Cetin, “A novel fusion method with thermal and rgb-d sensor data for human detection,” *IEEE Access*, vol. 10, pp. 66 831–66 843, 08 2022.

[16] Y. Xu, J. Li, and F. Zhang, “A UAV-Based Forest Fire Patrol Path Planning Strategy,” *Forests*, vol. 13, p. 1952, 11 2022.

[17] M. Evita, S. Seno, A. Zakiyyatuddin, A. Muid, and M. Djamal, “Application of UAV (Unmanned Aerial Vehicle) Photogrammetry for Forest Fire Early Detection System,” *IOP Conference Series: Earth and Environmental Science*, vol. 830, p. 012025, 09 2021.

[18] Y. Zhang, S. Chen, W. Wang, W. Zhang, and L. Zhang, “Pyramid Attention Based Early Forest Fire Detection Using UAV Imagery,” *Journal of Physics: Conference Series*, vol. 2363, p. 012021, 11 2022.

[19] C.-B. Liu and N. Ahuja, “Vision based fire detection,” *Proceedings - International Conference on Pattern Recognition*, vol. 4, pp. 134–137, 09 2004.

[20] J.-H. Kim and B. Lattimer, “Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot,” *Fire Safety Journal*, vol. 72, 02 2015.

[21] A. Cowlard, W. Jahn, C. Abecassis-Empis, G. Rein, and J. L. Torero, “Sensor Assisted Fire Fighting,” *Fire Technology*, vol. 46, no. 3, pp. 719–741, Jul 2010. [Online]. Available: https://doi.org/10.1007/s10694-008-0069-1

[22] V. Sherstjuk, M. Zharikova, and I. Sokol, “Forest Fire-Fighting Monitoring System Based on UAV Team and Remote Sensing,” in *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, 2018, pp. 663–668.

[23] D. W. Casbeer, R. W. Beard, T. W. McLain, Sai-Ming Li, and R. K. Mehra, "Forest fire monitoring with multiple small UAVs," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005, pp. 3530–3535 vol. 5.

[24] L. Merino, F. Caballero, J. R. M. de Dios, I. Maza, and A. Ollero, "Automatic forest fire monitoring and measurement using unmanned aerial vehicles," in *Proceedings of the 6th International Congress on Forest Fire Research*, 2010.

[25] D. A. Saikin, T. Baca, M. Gurtner, and M. Saska, "Wildfire fighting by unmanned aerial system exploiting its time-varying mass," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2674–2681, 2020.

[26] B. Aydin, E. Selvi, J. Tao, and M. J. Starek, "Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting," *Drones*, vol. 3, no. 1, p. 17, 2019.

[27] (2022) Drones for FireFighting : How Drones are Used by Firefighters. Accessed on December 20, 2022. [Online]. Available: https://www.flytnow.com/blog/drone-fire-fighting

[28] (2018) PUBLIC SAFETY DRONES: AN UPDATE. Accessed on December 20, 2022. [Online]. Available: https://dronecenter.bard.edu/files/2018/05/CSD-Public-Safety-Drones-Update-1.pdf

[29] S. Papaioannou, P. Kolios, T. Theocharides, C. Panayiotou, and M. Polycarpou, "Towards Automated 3D Search Planning for Emergency Response Missions," *Journal of Intelligent & Robotic Systems*, vol. 103, 09 2021.

[30] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Multiple UAVs in forest fire fighting mission using particle swarm optimization," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1404–1409.

[31] K. Satish and N. Kishan, *Using UAV Technologies for Fire Fighting.* F9 Info. Technologies (P) Limited.

[32] M. Zhang, W. Li, M. Wang, S. Li, and B. Li, "Helicopter–UAVs Search and Rescue Task Allocation Considering UAVs Operating Environment and Performance," *Computers & Industrial Engineering*, vol. 167, p. 107994, 02 2022.

[33] Y. Zheng, Y.-C. Du, H.-F. Ling, W.-G. Sheng, and S.-Y. Chen, "Evolutionary Collaborative Human-UAV Search for Escaped Criminals," *IEEE Transactions on Evolutionary Computation*, vol. PP, pp. 1–1, 06 2019.

[34] L. Hong, Y. Wang, Y. Du, X. Chen, and Y. Zheng, "UAV search-and-rescue planning using an adaptive memetic algorithm," *Frontiers of Information Technology & Electronic Engineering*, 10 2021.

[35] K. Holman, J. Kuzub, and G. Wainer, "UAV search strategies using cell-DEVS," *Spring Simulation Multiconference 2010, SpringSim'10*, p. 152, 09 2010.

[36] Z.-k. Li, H.-b. Wei, X.-m. Zhang, H.-y. Han, and L.-y. Zhang, *Research on Obstacle Avoidance Technology of Fire Fighting UAV.* Springer Singapore, 09 2022, pp. 543–555.

[37] V. Walter, V. Spurný, M. Petrlík, T. Báča, D. Žaitlík, L. Demkiv, and M. Saska, "Extinguishing Real Fires by Fully Autonomous Multirotor UAVs in the MBZIRC 2020 Competition," *Field Robotics*, vol. 2, pp. 406–436, 03 2022.

[38] M. Beul, M. Schwarz, J. Quenzel, M. Splietker, S. Bultmann, D. Schleich, A. Rochow, D. Pavlichenko, R. Rosu, P. Lowin, B. Scheider, M. Schreiber, F. Süberkrüb, and S. Behnke, "Target Chase, Wall Building, and Fire Fighting: Autonomous UAVs of Team NimbRo at MBZIRC 2020," *Field Robotics*, vol. 2, 03 2022.

[39] S. Martinez, R. Rey, D. Alejo, D. Acedo, J. Cobano, A. Rodríguez Ramos, P. Campoy, L. Merino, and F. Caballero, "Skyeye Team at MBZIRC 2020: A team of aerial and ground robots for GPS-denied autonomous fire extinguishing in an urban building scenario," *Field Robotics*, 04 2021.

[40] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "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 (IROS)*, 2018, pp. 1–8.

[41] 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, May 2021.

[42] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at https://octomap.github.io. [Online]. Available: https://octomap.github.io

[43] V. Nydrle, "Design of a Specialized UAV Platform for the Discharge of a Fire Extinguishing Capsule," Bachelor's Thesis, Czech Technical University in Prague, 2020.

[44] G. Gibbons, J. Segui-Garza, and R. Hansell, "Low-cost resin infusion mould tooling for carbon fibre composites manufacture," *Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering*, vol. 224, 04 2010.

[45] Mechanics of Materials: Bending – Normal Stress. Accessed on June 10, 2022. [Online]. Available: https://www.bu.edu/moss/mechanics-of-materials-bending-normal-stress/

[46] V. P. Leinveber Jiří, *Strojnické tabulky, 5.vydání.* Praha: Albra, 2011.

[47] (2021) Just How Strong is Carbon Fiber? Accessed on April 14, 2022. [Online]. Available: https://dragonplate.com/just-how-strong-is-carbon-fiber

[48] D. L. Gabriel, J. Meyer, and F. du Plessis, "Brushless DC motor characterisation and selection for a fixed wing UAV," in *IEEE Africon '11*, 2011, pp. 1–6.

[49] (2020) Pixhawk 4. Accessed on April 14, 2020. [Online]. Available: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html

[50] Neo-M8 u-blox M8 concurrent GNSS modules data sheet. Accessed on April 27, 2020. [Online]. Available: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%28UBX-15031086%29.pdf

[51] Intel® RealSense™ Depth Camera D435i. Accessed on April 1, 2020. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/

[52] E. Ulin-Avila and J. Ponce-Hernandez, "Kalman filter estimation and its implementation," in *Adaptive Filtering*, W. Cao and Q. Zhang, Eds. Rijeka: IntechOpen, 2021, ch. 3.

[53] S. Y. Chang and H.-C. Wu, "Tensor Kalman Filter and Its Applications," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 01 2022.

[54] P. Vanicek and M. Omerbashich, "Does a navigation algorithm have to use a Kalman filter?" *Canadian Aeronautics and Space Journal*, vol. 45, pp. 292–296, 01 1999.

[55] B. Leach, "A Kalman Filter Approach to Navigation on the NAE Convair 580 Aeromagnetics Research Aircraft," *Computer Science*, p. 111, 02 1981.

[56] X. Gao, H. Luo, B. Ning, F. Zhao, L. Bao, Y. Gong, Y. Xiao, and J. Jiang, "RL-AKF: An Adaptive Kalman Filter Navigation Algorithm Based on Reinforcement Learning for Ground Vehicles," *Remote Sensing*, vol. 12, p. 1704, 05 2020.

[57] S. Riaz, "Mathematical Modeling of INS/GPS Based Navigation System Using Discrete Time Extended Kalman Filter Schemes for Flapping Micro Air Vehicle," *International Journal of Micro Air Vehicles*, vol. 3, pp. 25–33, 03 2011.

[58] E. Benhamou, "Kalman filter demystified: from intuition to probabilistic graphical model to real case in financial markets," *Computer Science*, 11 2018.

[59] G. Rigatos, "Statistical Validation of Multi-Agent Financial Models Using the H-Infinity Kalman Filter," *Computational Economics*, vol. 58, pp. 1–22, 10 2021.

[60] K. Fronckova and A. Slaby, "Kalman filter employment in image processing," in *Computational Science and Its Applications – ICCSA 2020*, O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B. O. Apduhan, A. M. A. Rocha, E. Tarantino, C. M. Torre, and Y. Karaca, Eds. Cham: Springer International Publishing, 2020, pp. 833–844.

[61] B. Santana, E. Cherif, A. Bernardino, and R. Ribeiro, "Real-Time Georeferencing of Fire Front Aerial Images Using Iterative Ray-Tracing and the Bearings-Range Extended Kalman Filter," *Sensors*, vol. 22, p. 1150, 02 2022.

[62] J. Silvela and J. Portillo, "Breadth-first search and its application to image processing problems," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1194–1199, 2001.

[63] (2018) Extracting indices from a PointCloud — Point Cloud Library documentation. Accessed on September 16, 2022. [Online]. Available: https://pcl.readthedocs.io/projects/tutorials/en/latest/extract_indices.html#extract-indices

[64] J. M. Martinez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua, and B. Sierra, "RANSAC for Robotic Applications: A Survey," *Sensors*, vol. 23, p. 327, 12 2022.

[65] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, pp. 163–, 08 1987.

[66] Z. Zhang, "Perspective camera," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 590–592.

[67] 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, pp. 738/1–16, October 2021.

[68] K. Helsgaun, "An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems," *Computer Science*, 12 2017.

[69] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154.

Appendices

CD Content

Table 1 lists arrangement of directories on the attached CD.

Directory name	**Description**
thesis.pdf	the thesis in pdf format
sources	latex source codes
CAD_model.zip	model of the designed UAV
media	multimedia materials

Table 1: CD Content.

List of abbreviations

Table 2 lists abbreviations used in this thesis.

Abbreviation	Meaning
UAV	Unmanned Aerial Vehicle
FEE CTU	Faculty of Electrical Engineering, Czech Technical University in Prague
MRS	Multi-Robot Systems group at FEE CTU
FPV	First Person View
ROS	Robot Operating System
TCP	Transmission Control Protocol
MBZIRC	Mohamed Bin Zayed International Robotics Challenge
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
ESC	Electronic Speed Controller
LiPo	Lithium-Polymer accumulator
PCB	Printed Circuit Board
LiDAR	Light Detection and Ranging
PWM	Pulse Width Modulation
IMU	Inertial Measurement Unit
USB	Universal Serial Bus
EM	Electromagnetic
IR	Infra-red
FoV	Field of View
RMSE	Root Mean Squared Error

Table 2: Lists of abbreviations.