CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

MASTER'S THESIS

Bc. Vít Krátký

Safe Autonomous Aerial Surveys of Historical Building Interiors

Department of Cybernetics
Thesis supervisor: **Ing. Vojtěch Spurný**

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Krátký Vít**	Personal ID number:	**434740**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Control Engineering**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

Safe Autonomous Aerial Surveys of Historical Building Interiors

Master’s thesis title in Czech:

Bezpečný průzkum interiérů historických budov za pomocí autonomních bezpilotních helikoptér

Guidelines:

The aim of the thesis is to improve upon a system, designed for stabilization of formations of Unmanned Aerial Vehicles (UAVs) in the task of cooperative filming [1], for safe real-world deployment by extending it with specific behaviours in case of different types of failures. Additionally, prepare the system for the Reflectance Transformation Imaging (RTI) scanning method [3] used for documentation of historical buildings. The implemented system will be compatible with the current MRS system for UAV control and it will lead to its deployment in real-world scenarios.

Work plan:

1) Extend the system for stabilization of formations of UAVs in the task of cooperative filming to increase its modularity.
2) Identify sources of potential system failures (absence of sensory data, failure of member of formation, etc.) and, based on this analysis, design and implement a subsystem that ensures safe carrying out of missions.
3) Prepare the system for RTI [3] method used for documentation of historical buildings.
4) Verify the system in the Gazebo simulator under ROS using scenarios inspired by the environment of historical buildings.
5) Prepare the system for real-world experiments with UVDAR [2] that will be conducted based on availability of the multi-rotor helicopters in the MRS laboratory.

Bibliography / sources:

[1] M. Saska, V. Krátký, V. Spurný and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 2017, pp. 1-8.

[2] V. Walter, M. Saska and A. Franchi, "Fast Mutual Relative Localization of UAVs using Ultraviolet LED Markers," 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, 2018, pp. 1217-1226.

[3] Cultural Heritage Imaging, “Reflectance Transformation Imaging”, http://culturalheritageimaging.org/Technologies/RTI/ [cit. 2019-1-31], 2016.

Name and workplace of master’s thesis supervisor:

Ing. Vojtěch Spurný, Multi-robot Systems FEL

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

Date of master’s thesis assignment: **15.02.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until:
by the end of summer semester 2019/2020

Ing. Vojtěch Spurný Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. 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

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.

Prague, date ....

Acknowledgements

Firstly I would like to thank Ing. Vojtěch Spurný for his great support throughout this project. Further, my thanks go to Mgr. Michaela Čadilová for the expert consultation, and to other people from Multi-robot Systems group for valuable advice and assistance with the realization of experiments.

Abstract

This thesis is aimed at development of the system for safe autonomous survey of historical building interiors by the cooperative formation of multi-rotor unmanned aerial vehicles (UAVs). The proposed solution involves the method for safe trajectory tracking based on the leader-follower scheme and model predictive control, detection of potential faults and failures, and the mission controller which ensures the control of cooperation of particular UAVs and proper reaction on occurrence of faults and failures. The proposition of the whole system is influenced by the aim at its deployment in real world scenarios motivated by the documentation of historical monuments. The developed system is firstly evaluated in simulations. After that, it is tested in a real world scenario with the real UAVs.

Keywords: unmanned aerial vehicles, multi-robot formation, model predictive control, three points lighting, reflectance transformation imaging, mission control, historical buildings scanning

Abstrakt

Cílem této práce je vývoj systému pro bezpečný autonomní průzkum interiérů historických budov za pomocí vícerotorových autonomních bezpilotních helikoptér. Navržené řešení zahrnuje metodu pro sledování požadované trajektorie založené na přístupu lídr-následovník a prediktivním řízení, detekci potenciálních chyb a systému pro řízení mise, který zprostředkovává spolupráci mezi jednotlivými členy formace a korektní reakci na nastalé chyby jednotlivých podsystémů. Návrh celého systému je ovlivněn jeho plánovaným nasazením v rámci skenování interiérů historických budov. Funkčnost navrženého systému je nejprve otestována v rámci početných simulací a následně během experimentu s reálnými bezpilotními helikoptérami.

Klíčová slova: bezpilotní vzdušné helikoptéry, formace více robotů, prediktivní řízení, metoda tří bodového osvětlení, plánování a řízení mise, skenování historických budov

Contents

List of Figures ix

List of Tables xiii

1 Introduction 1

1.1 State-of-the-art 2

1.2 Problem statement 3

2 System overview 5

3 Formation control 8

3.1 Leader-follower scheme 9

3.2 Kinematic model 12

3.3 Representation of obstacles 14

3.4 Formation control method 15

3.4.1 Positional control 17

3.4.2 Orientation control 22

3.5 Comparison of solvers 25

4 Mission controller 26

4.1 Mission controller for normal operation 27

4.2 Faults and failures analysis 29

4.3 Mission controller for faulty operation 31

5 Reflectance transformation imaging 36

5.1 Reflectance transformation imaging method 36

5.2 RTI scanning implementation 37

5.2.1 Set generation 37

5.2.2 Determination of the best sequence 40

5.2.3 Trajectory generation and tracking 45

5.3 Image post-processing 48

6 Experimental results 50

6.1 Complex experiment 50

6.2 RTI experiment 61

6.3 Real-world experiment 67

7 Conclusion 71

Bibliography 73

Appendices 79

Appendix List of abbreviations 81

List of Figures

1.1 The images from the deployment of multi-rotor helicopters within the interiors of historical buildings. 2

2.1 The three dimensional scan of the church in Chlumin obtained from the measurement of stationary terrestrial laser scanner Leica MS60. 6

2.2 The scheme of the complete system for documentation of interiors of historical buildings proposed in this thesis. 7

3.1 Illustration of the leader-follower scheme originally presented in [18]. 10

3.2 Illustration of the fixed formation leader-follower scheme defined by equations (3.4). 11

3.3 The illustration of the problem of usage of car-like kinematic model within the formation control method used in the task of cooperative documentation of historical building interiors. 13

3.4 Graphical illustration of the octree principle [23]. 15

3.5 The time demands of methods used for finding the distance from the nearest obstacle. 16

3.6 Graphical illustration of meaning of particular symbols used in equations (3.30) for computation of part of the objective function penalizing the occlusion caused by followers. 21

3.7 Graphs of parts of the objective function $J_{j,position}$. 24

4.1 Explanation of symbols used in the figures within Chapter 4 in which the mission controller is described. 26

4.2 The mission controller for normal operation as the finite state machine. . . 33

4.3 Part of the mission controller responsible for control of landing. 34

4.4 Part of the mission controller responsible for handling of faulty operation. . 35

5.1 The example of the generated set of RTI goals 38

5.2 The E-shape trajectory presented with three different orientations used for the experimental determination of dependence of the consumed energy on the direction of flight. 41

5.3 Illustration of the procedure of determining the predictable sequence of RTI positions for even number of horizontal rows. 43

5.4 Illustration of the procedure of determining the predictable sequence of RTI positions for odd number of horizontal rows. 44

5.5 Comparison of two different solutions of the RTI sequence determination problem. 45

5.6 Comparison of length of TSP solution obtained from the LKH solver and our predictable solution 46

5.7 Graph of the objective function for penalization of the occlusion during the RTI scanning phase. 47

5.8 Example of the PTM representation of the image obtained from the onboard camera without any post-processing. 49

6.1 The simulation environment used in the experiment described in Section 6.1. 51

6.2 Trajectories of particular UAVs in the formation during the transition between two OoIs with the use of the leader-follower scheme with virtual OoI. 54

6.3 Snapshots from the simulation of the formation flying through the narrow corridor during the experiment presented in Section 6.1. 55

6.4 Trajectories of robots during the complex experiment presented in Section 6.1. 56

6.5 The z coordinate of the trajectories of robots during the complex experiment presented in Section 6.1. 57

6.6 The φ_i and ε_i angles describing the orientation of particular robots during the complex experiment presented in Section 6.1. 58

6.7 The values of the objective functions $J_{j,p}$ and $J_{j,o}$ of particular robots during the complex experiment presented in Section 6.1. 59

6.8 The values of control inputs applied to leader during the complex experiment presented in Section 6.1. 60

6.9 The generated RTI positions and the trajectory flown by the follower carrying the light during the RTI scanning procedure. 64

6.10 The set of images taken by the onboard camera mounted on the leading UAV during the RTI experiment described in Section 6.2. 65

6.11 Comparison of PTMs representations of the image of scanned object obtained from the properly registered images (simulated by static camera) (a) and from the onboard camera without any post-processing (b). 65

6.12 Presentation of PTM representation of the images of scanned object obtained from the images taken during the RTI experiment performed in the realistic simulator Gazebo. 66

6.13 Specialized platform developed within Multi-robot Systems group for scanning of historical buildings interiors during its deployment in the experiment presented in Section 6.3. 67

6.14 The real-world scenario used within the experiment described in Section 6.3. 69

6.15 The sequence of images of the experimental scene taken by static camera during the experiment presented in Section 6.3. 70

List of Tables

4.1 Description of events and conditions used within the figure describing state machine for normal behaviour of the system (Figure 4.2). 30

5.1 Ranges of particular parameters used within generation of testing set for comparison of our predictable solution of TSP with solution provided by the LKH solver. 44

6.1 Overview of the values of particular constants used for the complex experiment presented in Section 6.1. 52

6.2 Overview of the values of particular constants used for the RTI experiment presented in Section 6.2. 62

6.3 Overview of the values of particular constants, connected with the RTI scanning procedure, used for the RTI experiment presented in Section 6.2. . . . 62

6.4 Overview of the values of particular constants connected with the RTI scanning procedure used for the outdoor RTI experiment presented in section Section 6.3. 68

1 CD Content 80

2 Lists of abbreviations 81

Chapter 1

Introduction

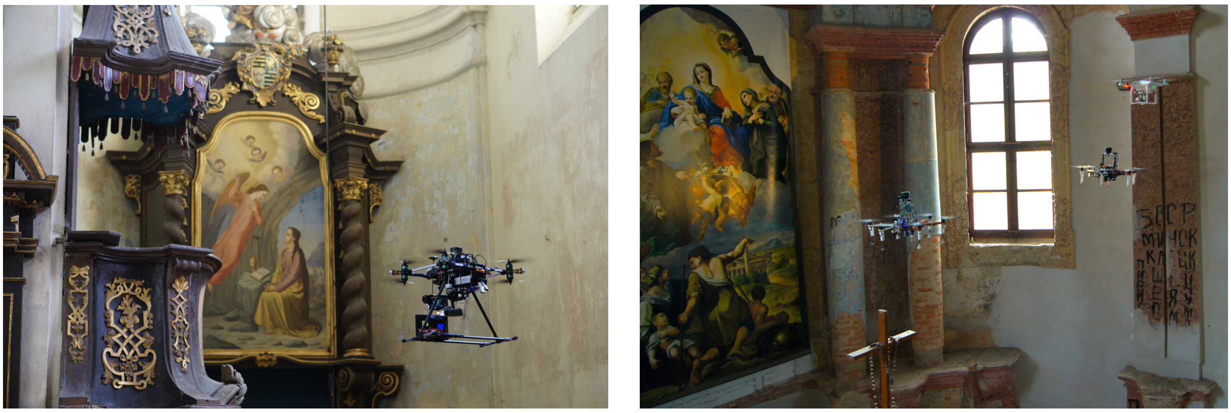
Robotic systems based on multi-rotor Unmanned Aerial Vehicles (UAVs) are becoming popular in a wide range of applications. They usually take advantage of the ability of multi-rotor UAV to hover in the air, move arbitrarily slowly in any direction and carry various sensors. The application of single, manually controlled UAV can be very profitable in numerous situations. Nevertheless, the number of possible applications can be significantly increased by introducing the autonomous cooperative teams of UAVs.

One of these applications is the documentation of interiors of historical buildings with distributed lighting, which is motivated by the preservation of cultural heritage in the form of digital documentation. It enables to plan renovations, perform later reconstructions of already destroyed historical buildings or art pieces, and also the visualization of the models of these objects. Methods for obtaining data needed for planning of restoration and conservation work as well as monitoring of the state of artefacts were already developed. However, these methods usually require to take the images of an artefact from different angles of view with various lighting conditions. This setup can be easily achieved within the typical reach of person, but it becomes problematic when we want to scan the areas located in the higher and hardly accessible parts of historical buildings.

One way how to overcome this problem and get the sensors and light sources into the proximity of scanned artefact is to build a scaffolding, which is not only expensive but also a very time-consuming process. In our previous work [1], we proposed the alternative approach - to use the team of cooperative multi-rotor UAVs, which are capable of carrying various sensors and also light sources (Figure 1.1). This method applies the leader-follower approach together with the Model Predictive Control (MPC) on receding horizon to safely track the desired trajectory and to achieve the required lighting.

However, this method does not implement the proper reactions on unexpected failures, and so the system requires to be operated by an experienced person. The goal of this thesis is to propose the system for obtaining data from hardly accessible places of historical buildings with a high level of autonomy. This approach does not lead only to speeding up of the whole process of documentation, but thanks to the eliminating of human faults and the possibility of fast autonomous reactions on occurred failures of the system, it also increases its reliability and safety. The entire system for autonomous documentation of interiors of historical buildings, presented in this thesis, is build above the system for control of multi-rotor helicopters developed by Multi Robot Systems group (MRS) at Faculty of Electrical Engineering at Czech Technical University in Prague.

The thesis is structured as follows: after an introduction, an overview of the state-of-the-art methods and the problem statement are given in this chapter. The thesis continues with an overview of the system built from the particular subsystems (Chapter 2). In Chapter 3, we provide a brief description of the original method for formation control proposed in [1] and its modifications made to increase its performance and modularity. Chapter 4 describes the mission controller, the main added part, which increases the autonomy of the system. Chapter 5 gives the description of the approach to implement the additional method of object documentation called Reflectance Transformation Imaging (RTI). In the last chapter, the verification of the system in the realistic robotic simulator Gazebo and experiments in real-world scenarios are presented.



(a) St. Mary Magdalene church in Chlumin (b) abandoned church in Stara Voda

Figure 1.1: The images from the deployment of multi-rotor helicopters within the interiors of historical buildings.

.

1.1 State-of-the-art

The problem of the documentation or monitoring of heritage sites is addressed in many publications [2, 3, 4]. However, most of the authors focus on the methods used for data processing rather than on the data acquisition process. Thus, they introduce different variants on photogrammetry, processing of laser scans, or less traditional methods for building three-dimensional models based on a set of images, laser scans or point clouds. Nevertheless, several papers, which are interested in acceleration or optimization of the data acquisition process were also presented in recent years.

The easiest way how to get the three dimensional model of some object is to use the static terrestrial laser scanner, which produces the point cloud representation. This scanner has to be moved to numerous sensing locations to obtain the complete 3D model not degenerated by self occlusion and occlusion caused by other objects. In [5], the possible scanned area is enlarged by introducing the handheld laser scanner, which enables continuous scanning while the operator is walking through the environment. The necessity of human involvement is significantly decreased in [6], where the authors introduce the unmanned ground vehicle (UGV), AVENUE, equipped with a terrestrial laser scanner, which is capable of choosing the set of sensing locations and autonomous navigation through the outdoor environment.

Another group of methods for obtaining the three dimensional model of some object uses the set of images from the camera as the input data. While the UGVs are more suitable for carrying a laser scanner than UAVs, since they are capable of staying still and have a higher maximum payload, the UAVs are favoured as carriers of the lightweight cameras. In comparison to UGVs, they have more extensive operational space and higher maximum velocity. Therefore they are often deployed for documentation or monitoring of large areas and hardly accessible places. However, mostly they are remotely controlled by the human operator [7, 8, 9] or navigated based on the defined waypoints and Global Navigation Satellite System (GNSS) [10, 11]. The article [11] is more related to our work since it aims at determining an optimal set of sensing locations to maximize the quality of the resulting 3D model while not exceeding the allowed travel budget.

Although the UAVs are often used for the purpose of surveillance, monitoring or documentation, they are rarely applied in indoor environments. We have found only one work, which proposes the system for documentation of interiors of historical buildings and presents experimental results [12]. In this paper, the authors describe the system for safe data acquisition with the use of a UAV in outdoor and indoor scenarios. The system provides valuable information from various onboard sensors, which helps the operator to remotely control the UAV.

We go much further beyond all the works above in several ways. Firstly, we actively influence the environment to provide the best lighting conditions and thus increase the quality of the gathered data. Secondly, to achieve the desired lighting, we deploy a formation of cooperating UAVs for documentation of interiors of heritage sites. Lastly, our proposed system is exceptional with regards to autonomy. Contrary to all presented works except for [6], we do not apply unmanned vehicles merely as remotely controlled carriers, but we aim to maximize their autonomy in order to decrease the time required for the data acquisition process and to increase the tolerance to human error.

1.2 Problem statement

The aim of this thesis is to design and implement the autonomous system for documentation of interiors of historical buildings with the usage of multiple cooperating multi-rotor helicopters, where one is supposed to carry the camera, while other carry the light sources. The system should be build above the MRS framework for multi-rotor helicopters control and uses the results of our previous work on formation control in task of cooperative filming in dark conditions [1]. Its main purpose is to minimize the human involvement in the scanning process and ensure proper reactions on failures including providing notifications about necessary intervention of human operator.

The system should enable simple definition of the desired scanning mission, which is assumed to be given by the experts in the field of restoration, conservation and historical science. It should also provide the possibility to set all relevant lighting parameters and switch among three lighting methods, namely the Three point lighting method [13], the method using the raking light [14], and the lighting approach enabling the usage of Reflectance Transformation Imaging (RTI) method (described in Chapter 5).

We assume the use of the UAVs that are capable of changing its orientation around the vertical axis independently on the direction of its motion. These UAVs have to be capable of carrying a light source or camera mounted on the mechanism that enables the change of their tilt in the vertical direction. Further, we assume that we have the map of the environment in the form of a point cloud, and arbitrary system or method, that provides reliable information about the position of particular UAVs within this map. Last important assumption is that the environment is free of dynamic obstacles apart from the UAVs which participated in the scanning mission.

To increase the reliability of the mutual avoidance within the formation, the system should incorporate the method for relative localization based on nearly Ultra-Violet (UV) Light Emitting Diodes (LEDs) presented in [15, 16]. The advantage of this approach in comparison to marker based relative localization (e.g. WhyCon [17]) is its independence from the light conditions. Nevertheless, in case of sufficiently precise method of localization in global map, the system for relative localization is not the necessary part of the proposed system.

As the system is supposed to be built above the MRS framework, it makes use of features and data that it provides, for detection of faults and commanding particular helicopters in the formation. Nevertheless, the system can be easily modified to work with any other framework, which provides similar features as the MRS system based on Robot Operating System (ROS). Finally, let us note that some of the proposed solutions presented in this thesis are significantly influenced by the specific properties of the task and by our aim to deploy the system in real-world scenarios.

Chapter 2

System overview

In order to clarify reasons for the usage of particular approaches in the following chapters, the overview of the whole system is provided in this chapter.

The whole system consists of the hardware parts, software parts and also necessary human resources. The first deployed part of the system is the 3D laser scanner, which is able to scan almost whole interior within tens of minutes and thus provide the map of the environment to other parts of the system (visualization of such data is provided in Figure 2.1). One of them is the expert(s) from the field of restoration and historical science, who use the map to specify the desired sensing locations together with desired lighting setup including choice of the lighting method. This scanning plan and the map of the environment are passed on to the robots, that are prepared to perform the assigned mission.

The robotic part of the system consists of several multi-rotor helicopters (UAVs). One of these helicopters (further referenced as leader) carries the high-resolution camera for photography, while the other (further referenced as followers) carry the light sources. All these helicopters are equipped with various onboard sensors, autopilot, and an onboard computer. The onboard computer includes the software which does the processing of sensory data, provides the localization of the robot in the environment, controls the motion of the UAV, enables the trajectory tracking by specifying the sequence of UAV configurations, and running of other software for high-level control of the UAV.

The behaviour of particular UAVs in the course of the mission is driven by the onboard running program for formation control and safe trajectory tracking originally presented in [1] and further improved in this thesis (Chapter 3). This method slightly varies for the leader and the followers. Nevertheless, all UAVs which participate in the mission communicate with each other and share the information about their position, their intentions and future trajectory. Due to inability to ensure the redundancy of all necessary hardware parts, each UAV has to have its human operator who can remotely take over the control of the UAV in case of failure of the onboard computer or another part of the system that disables the autonomous control of the UAV. Although it can seem to be arguable to

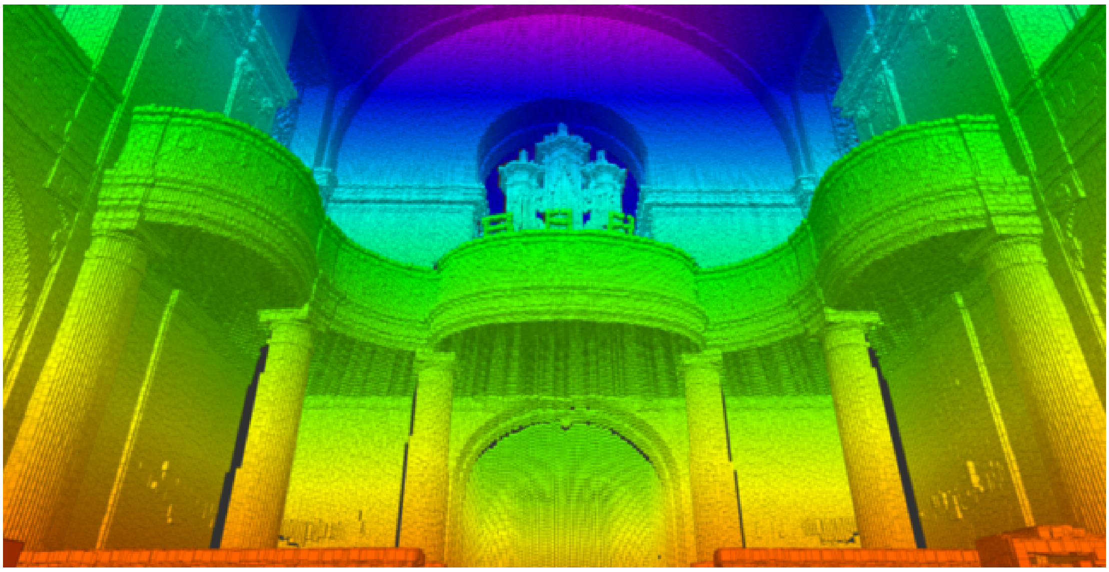


Figure 2.1: The three dimensional scan of the church in Chlumin obtained from the measurement of stationary terrestrial laser scanner Leica MS60.

talk about an autonomous system in connection with the necessary participation of human operators, note that the human operators serve only as another part of safety mechanism and they are not supposed to remotely control the UAV until the situation requires it.

The last part of the system is the computer (server) on which the mission controller program runs (described in Chapter 4 in details). The server communicates with all UAVs participating in the mission and provides important information about the state of particular UAVs and the whole mission to human operators. It autonomously coordinates the UAVs to achieve safe and cooperative carrying out of the mission and ensure the deterministic behaviour in case of failures. The mission controller also provides methods for safe pause or restart of the mission, change of the formation shape, or immediate automatic landing which can be called by the user. Apart from these methods, the behaviour of the mission controller can be easily set up by setting values for several variables and thus achieve its desired behaviour for a specific mission. The scheme of the complete system is shown in Figure 2.2.

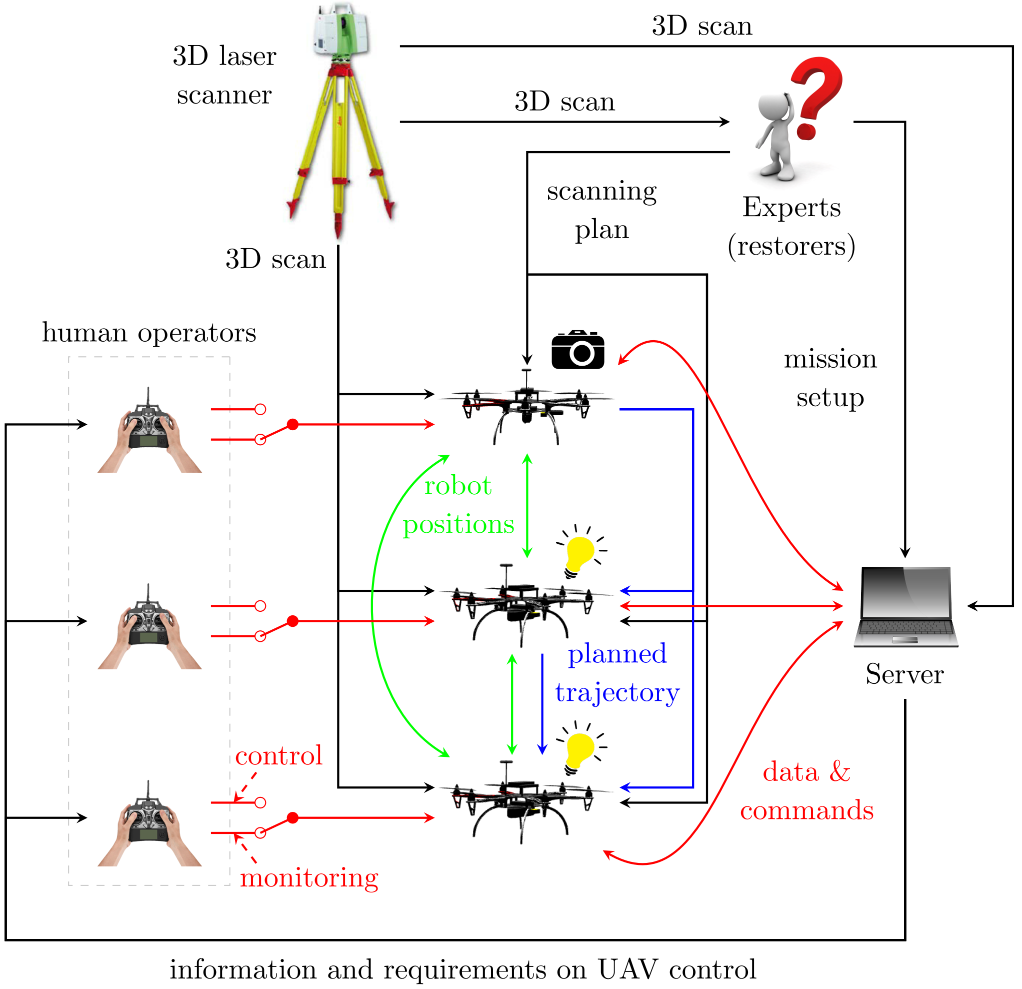


Figure 2.2: The scheme of the complete system for documentation of interiors of historical buildings proposed in this thesis. Although the scheme shows two UAVs carrying the lights, the system is capable of working with an arbitrary number of this type of UAVs.

Chapter 3

Formation control

As was already mentioned in previous chapters, the system for formation control, which is a necessary part of the proposed system, was originally presented in [1] and [18]. Although the method was intensively tested and numerous experiments were published within these two works, further testing shows some shortcomings and rooms for improvement. All changes, which were made within the system in comparison to [1] are described in this section. To provide insight into the original system, we will start with its general description in the following paragraphs.

The system is based on the leader-follower approach and the model predictive control on the receding horizon and comes from our previous works on formation control [19] and [20]. It requires an initial plan defining the trajectory of the leader with the camera, desired sensing locations together with positions of Objects of Interest (OoIs), and desired lighting setup. Given these inputs, the trajectory of the leader on the prediction horizon is optimized according to proposed objective and constraint functions.

Due to the possible independence of control of the position and orientation of multi-rotor UAV, the problem of finding optimal control inputs can be split into two separated optimization tasks. The first task is responsible for control of the position of j-th robot $P_j(t) = \{x_j(t), y_j(t), z_j(t)\}$ in global coordinate system $\mathcal{C}$, while the second task optimizes its orientation $O_j(t) = \{\varphi_j(t), \xi_j(t)\}$, where $\varphi_j(t)$ denotes the angle from the x-axis in xy-plane of $\mathcal{C}$ and $\xi_j(t)$ stands for the angle from the xy-plane in $\mathcal{C}$. In case of the leader, the objective function for positional control penalizes the distance from the desired position, fast changes in positional control inputs, positions close to the obstacles, and trajectories near positions of other UAVs in the formation. The objective function for orientation control takes into account the deviation from desired orientation and magnitude of changes in control inputs responsible for control of orientation.

The optimized trajectory of the leader together with desired lighting angles and positions of OoI then serve as the input for the computation of desired trajectories of followers. These trajectories are computed based on the defined leader-follower scheme and they are optimized in a similar way as the trajectory of the leader. The objective function for the positional control of followers is composed of all parts from the objective function for the leader, but it has two additional parts. The first one penalizes the trajectories which cause an occlusion in the camera field of view. The second one penalizes the trajectories which collide with the planned trajectory of the leader or other followers with higher priority. The objective function for orientation control of followers is the same as for the leader.

In the following sections, we provide the description of changes made within the above outlined method for formation control in order to increase its performance together with the necessary description of the original method, reasons for the changes and argumentation for them. The last section in this chapter deals with the choice of a proper solver for defined optimization task.

3.1 Leader-follower scheme

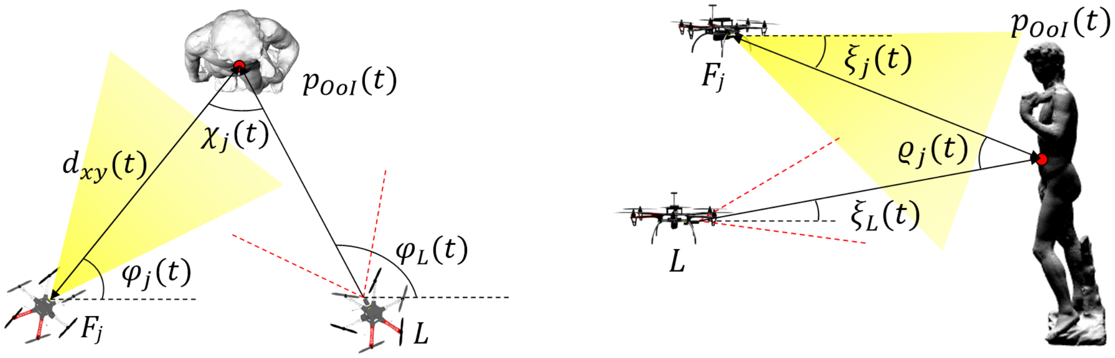
In the original method for formation control, we define the single leader-follower scheme for computation of desired trajectories of particular followers, which were computed based on the position of leader $P_L(t)$, orientation of its camera $O_L(t)$, position of OoI $p_{OoI} = \{x_{OoI}(t), y_{OoI}(t), z_{OoI}(t)\}$, desired lighting angles of j-th light $\chi_j(t)$ and $\varrho_j(t)$, and its desired distance from OoI d_j. The desired trajectory at time t was then given by equations

$$\begin{aligned}
\varphi_j(t) &= \varphi_L(t) + \chi_j(t), \\
\xi_j(t) &= \xi_L(t) + \varrho_j(t), \\
x_j(t) &= x_{OoI}(t) - d_j \cos(\varphi_j(t)), \\
y_j(t) &= y_{OoI}(t) - d_j \sin(\varphi_j(t)), \\
z_j(t) &= z_{OoI}(t) + d_{xy}(t) \tan(\xi_j(t)),
\end{aligned} \tag{3.1}$$

where $\chi_j(t)$ and $\varrho_j(t)$ are desired lighting angles relative to the camera optical axis and $d_{xy}(t)$ is the Euclidean distance computed without considering z coordinate.

To prevent big unnecessary jumps in the desired positions of followers during switching between particular OoIs, caused by fast changes in the orientation of camera, we have replaced the computation of desired light orientations $\varphi_j(t)$, $\xi_j(t)$ presented in equation (3.1) by

$$\begin{aligned}
\varphi_j(t) &= \begin{cases} \varphi_L(t) + \chi_j(t) & \text{if } |\varphi_L(t) - \text{ang}_h(P_L(t), P_{OoI}(t))| - \frac{AoV_h}{2} \leq 0, \\ \text{ang}_h(P_L(t), P_{OoI}(t)) + \chi_j(t) & \text{if } |\varphi_L(t) - \text{ang}_h(P_L(t), P_{OoI}(t))| - \frac{AoV_h}{2} > 0, \end{cases} \\
\xi_j(t) &= \begin{cases} \xi_L(t) + \varrho_j(t) & \text{if } |\xi_L(t) - \text{ang}_v(P_L(t), P_{OoI}(t))| - \frac{AoV_v}{2} \leq 0, \\ \text{ang}_v(P_L(t), P_{OoI}(t)) + \varrho_j(t) & \text{if } |\xi_L(t) - \text{ang}_v(P_L(t), P_{OoI}(t))| - \frac{AoV_v}{2} > 0, \end{cases}
\end{aligned} \tag{3.2}$$



(a) top view

(b) side view

Figure 3.1: Illustration of the leader-follower scheme originally presented in [18].

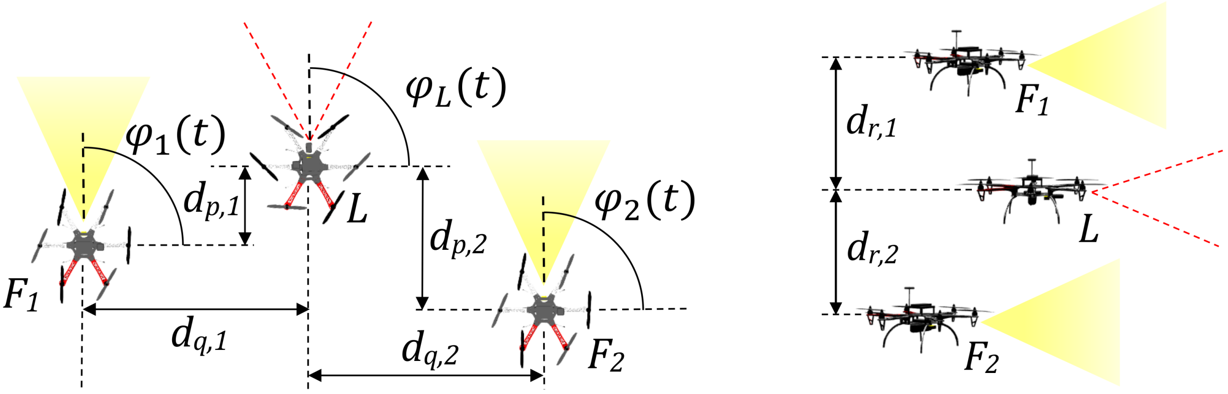
where AoV_h and AoV_v stand for the horizontal and vertical angle of view of the camera respectively, function $\text{ang}_h(\cdot)$ returns the angle between the projection of vector defined by its arguments into xy-plane and x-axis, and function $\text{ang}_v(\cdot)$ returns the angle between the vector defined by its arguments and the xy-plane. The usage of equation (3.2) ensures that the current orientation of camera is used only when the OoI is inside its field of view, otherwise the virtual camera orientation is computed from the current position of leader and OoI. The graphical illustration of the leader-follower scheme described by equations (3.1) and (3.2) is provided in Figure 3.1.

The aim of this approach is not to fly in the fixed formation, but to precisely achieve the desired lighting, which is the primary goal of the task. Nevertheless, in some situations, this behaviour can lead to potentially dangerous manoeuvres and also disable the usage of relative localization that is based on the sensors with a limited field of view. Therefore we propose another two alternatives to the original leader-follower scheme.

The first alternative is based on the previously presented leader-follower scheme, but instead of the position of the OoI, it makes use of the virtual OoI placed in a certain distance in front of the camera. The position of such a virtual OoI can be computed based on equations

$$\begin{aligned} d_{v,xy}(t) &= d_v \cos(\xi_L(t)), \\ x_v(t) &= x_L(t) + d_{v,xy} \cos(\varphi_L(t)), \\ y_v(t) &= y_L(t) - d_{v,xy} \sin(\varphi_L(t)), \\ z_v(t) &= z_L(t) + d_v \sin(\xi_L(t)), \end{aligned} \tag{3.3}$$

where d_v is the desired distance between the virtual OoI and camera and $x_v(t)$, $y_v(t)$ and $z_v(t)$ denotes the position of virtual OoI at time t. By applying equation (3.3) and substituting triplet $\{x_v(t), y_v(t), z_v(t)\}$ for triplet $\{x_{OoI}(t), y_{OoI}(t), z_{OoI}(t)\}$ in equation (3.1), we got the new leader-follower scheme, which results in the fixed shape formation for constant lighting angles $\chi_j(t)$, $\varrho_j(t)$ and more compact formation when these angles varies. The drawback of



(a) top view

(b) side view

Figure 3.2: Illustration of the fixed formation leader-follower scheme defined by equations (3.4).

this approach is that the desired lighting is achieved only if the current OoI is exactly in the middle of the camera field of view in distance equals to d_v.

The second alternative does not aim to achieve desired lighting but to define the fixed shape of the formation. Therefore it is not influenced by the position of OoI at all. Since we consider the UAVs capable of flying in any direction, we can define this leader-follower scheme simply as

$$\begin{aligned} x_j(t) &= x_L(t) + d_{p,j}\cos(\varphi_L(t)) - d_{q,j}\sin(\varphi_L(t)),\\ y_j(t) &= y_L(t) + d_{p,j}\sin(\varphi_L(t)) + d_{q,j}\cos(\varphi_L(t)),\\ z_j(t) &= z_L(t) + d_{r,j},\\ \varphi_j(t) &= \varphi_L(t),\\ \xi_j(t) &= 0, \end{aligned} \tag{3.4}$$

where $d_{p,j}$ is the desired distance of j-th follower from the leader in direction of its heading $\varphi_L(t)$, $d_{q,j}$ is the desired distance of j-th follower from the leader in direction orthogonal to direction defined by heading $\varphi_L(t)$, and $d_{r,j}(t)$ is the desired distance of j-th follower from the leader in vertical direction. The graphical illustration of this approach is shown in Figure 3.2.

Introducing and implementation of these three different variants of leader-follower scheme enable to use the system in various and more complex scenarios. While the first, original scheme, is best for maintaining the desired lighting setup, the second scheme is more suitable for the filming of continuous snapshots with defined, possibly varying, lighting. The third scheme is clearly ideal for flying without need to take any snapshots or images, e.g. flying to trajectory start, flying between particular OoIs etc. Since the system enables the switching between the presented leader-follower schemes during one mission, the experts can define in which parts of the desired trajectory they are interested in taking images or snapshots. In the remaining parts of the trajectory, more safe and easily monitored behaviour can be achieved by applying the fixed formation leader-follower scheme.

Further reason for introducing the alternatives of the original leader-follower scheme is also the possibility of the usage of relative localization. The methods for relative localization are often based on the sensors with limited range and field of view. Therefore, for their safe application, it is necessary to ensure that particular members of the formation stay within the admissible space for the whole course of the mission. This condition can be easily fulfilled by applying the fixed formation flying and also the approach with virtual OoI with limited lighting angles $\chi_j(t)$ and $\varrho_j(t)$. On the other hand, within the original method, part of the trajectories are often outside the admissible space for standard sensors used for relative localization. Nevertheless, since the UAVs are equipped with the system for global localization and they are able to communicate with each other, the temporary absence of data from relative localization is acceptable. Moreover, the mission controller (described in Chapter 4 in details) includes the control mechanism for detection of absence of localization data and proper reaction on this situation.

3.2 Kinematic model

Within the original method for formation control presented in [1], we have used the extended car-like model described in [21] with additional control inputs for control of the orientation of camera or light. It comes out from the standard car-like model with inputs velocity $v(t)$ and curvature $K(t)$ defined as

$$K(t) = \frac{\tan(\phi(t))}{L}, \tag{3.5}$$

where $\phi(t)$ stands for the steering angle of the model and L denotes the distance between front and rear pair of wheels. The third control input $w_j(t)$ is the ascent velocity, which enables the control of the model in the vertical direction. Other two inputs are angular rates $\omega_j(t)$ and $\varepsilon_j(t)$, which controls the orientation of camera or light given by angles $\varphi_j(t)$ and $\xi_j(t)$. The complete kinematic model of j-th UAV was given by equations

$$\begin{aligned}
\dot{x}_j(t) &= v_j(t)\cos(\theta_j(t)),\\
\dot{y}_j(t) &= v_j(t)\sin(\theta_j(t)),\\
\dot{z}_j(t) &= w_j(t),\\
\dot{\theta}_j(t) &= K_j(t)v_j(t),\\
\dot{\varphi}_j(t) &= \omega_j(t),\\
\dot{\xi}_j(t) &= \varepsilon_j(t),
\end{aligned} \tag{3.6}$$

where $\theta_j(t)$ is the virtual heading of kinematic model.

The reason for the usage of this kinematic model was to ensure the generation of smooth trajectories. However, this approach does not fully exploit the capabilities of multi-rotor UAVs and in some situations leads to failures in finding feasible trajectory even if it clearly exists. One of these example situations is illustrated in Figure 3.3, where the UAV carrying the light has almost zero velocity and should fly to the next OoI. Although the way towards its next desired position is clear, it cannot fly in this direction since the heading of kinematic model $\theta_j(t)$ has another course.

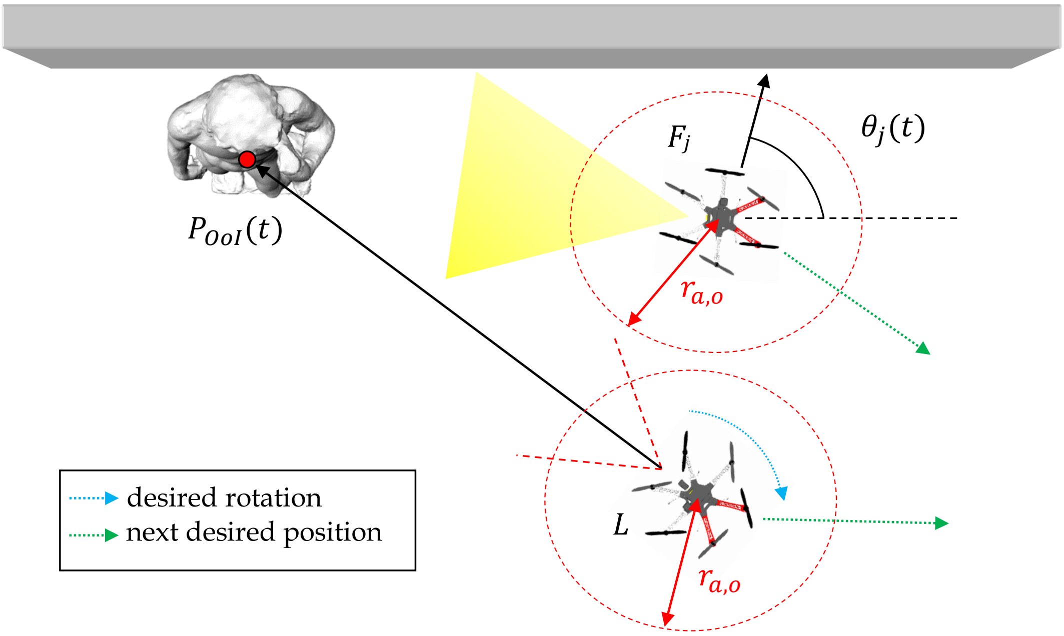


Figure 3.3: The illustration of the problem of usage of car-like kinematic model within the formation control method used in the task of cooperative documentation of historical building interiors. Although the velocity of the follower F_j is zero, it cannot start to fly in desired direction (marked by green arrow) since the heading of the car-like model $\theta_j(t)$ is different. In the picture, $r_{a,o}$ stands for the avoidance radius with respect to obstacles and $P_{OoI}(t)$ for the position of OoI at time t.

Therefore, we propose to use different kinematic model, which corresponds more to the capabilities of deployed UAVs. Since the desired application requires only small velocities, usually not exceeding 1 m s^{-1}, it is possible to use as simple model as possible, the kinematic model of point mass in three dimensional space extended by control of orientation of light or camera. This kinematic model can be defined with equations

$$\begin{aligned}\dot{x}_j(t) &= v_{x,j}(t),\\ \dot{y}_j(t) &= v_{y,j}(t),\\ \dot{z}_j(t) &= v_{z,j}(t),\\ \dot{\varphi}_j(t) &= \omega_j(t),\\ \dot{\xi}_j(t) &= \varepsilon_j(t),\end{aligned} \tag{3.7}$$

where $v_{x,j}(t)$, $v_{y,j}(t)$ and $v_{z,j}(t)$ are the velocities in particular axes of the global coordinate frame.

For our usage within the MPC on the receding horizon framework, we suppose a constant value of control inputs between particular transition points and a constant time interval between any two consequent transition points. Thus, we can get the discrete kinematic model by integration of equation (3.7) over the interval T_s, which results in

$$\begin{aligned} x_j(k+1) &= x_j(k) + v_{x,j}(k+1)T_s, \\ y_j(k+1) &= y_j(k) + v_{y,j}(k+1)T_s, \\ z_j(k+1) &= z_j(k) + v_{z,j}(k+1)T_s, \\ \varphi_j(k+1) &= \varphi_j(k) + \omega_j(k+1)T_s, \\ \xi_j(k+1) &= \xi_j(k) + \varepsilon_j(k+1)T_s. \end{aligned} \tag{3.8}$$

3.3 Representation of obstacles

In the original method, we have approximated the obstacles by cylinders and flat planes. Nevertheless, this approach is efficient only for a low number of obstacles and not so clattered environments. Therefore, we have chosen more suitable structure for the representation of obstacles - the octree.

The octree, which was introduced by D. Meagher in [22], is a structure used to represent any three-dimensional object efficiently. It utilizes the three dimensional binary trees with branching factor equal to eight. Each node of this tree corresponds to certain part of a 3D object or space and has assigned value, which informs about its occupation. The root node of the tree represents the whole three dimensional object or space by a cube of certain size. Each of its eight child nodes then represents one eighth of this cube. Each of these cubes are again evenly divided into eight smaller cubes assigned to eight child nodes. This procedure is repeated until a desired resolution is met. Graphical illustration of the octree principle is shown in Figure 3.4.

The octree structure enables fast computation of different kinds of transformations and effective nearest neighbour search. Therefore, it is often deployed not only in the computer graphics but also in the field of robotics to represent the environment [24, 25]. Since our objective function used within the formation control method requires the computation of the distance between a certain point and nearest obstacle, we also take the benefit of octree structure as the space partitioning representation of the environment. Its main advantage for our application is that it speeds up the process of solving the optimization task while enabling to model complex environments. The quantitative comparison of computational time required to find the nearest obstacle for all transition points on the planning horizon with the original representation of obstacles and with the octree representation is provided in Figure 3.5. The second significant advantage is that the three-dimensional scan of the environment, which we can get from the laser scanner can be easily converted to the octree structure and used directly within the formation control method.

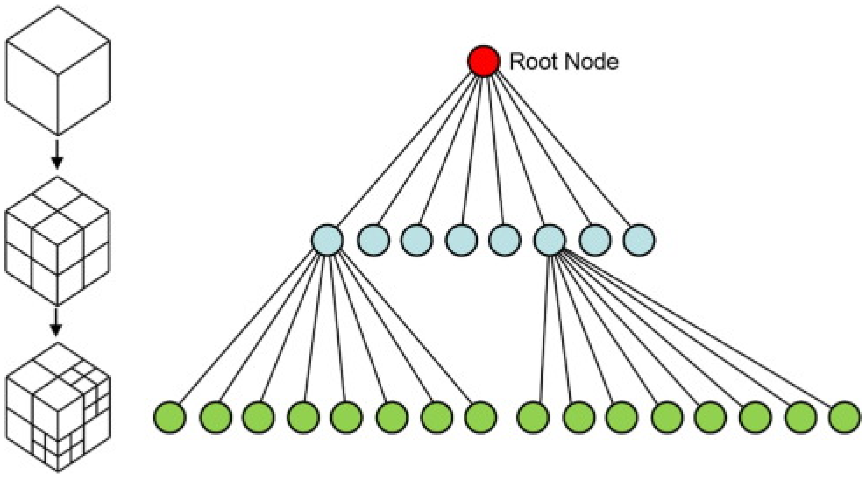


Figure 3.4: Graphical illustration of the octree principle [23].

3.4 Formation control method

Although the method for formation control was originally presented in [1] and [18] and the subject of this thesis regarding the formation control method is only its alteration, the shortened description of the complete method is provided in this section, to achieve the comprehensibility without reading another work. The presented modifications are introduced either due to requirements given by formerly described changes in the system, like the representation of obstacles, kinematic model, or changes based on the revelations of potential improvements during the long-term testing.

The modified method for formation control is built on the same principle as the original one. The leader takes the part of the initial trajectory prepared by experts. This trajectory is then optimized on the horizon of length N according to the defined objective function and sent to followers. The process goes on with computing of desired trajectories of followers with the usage of leader-follower formation scheme presented in Section 3.1, which are then optimized with a similar approach as in the case of the leader.

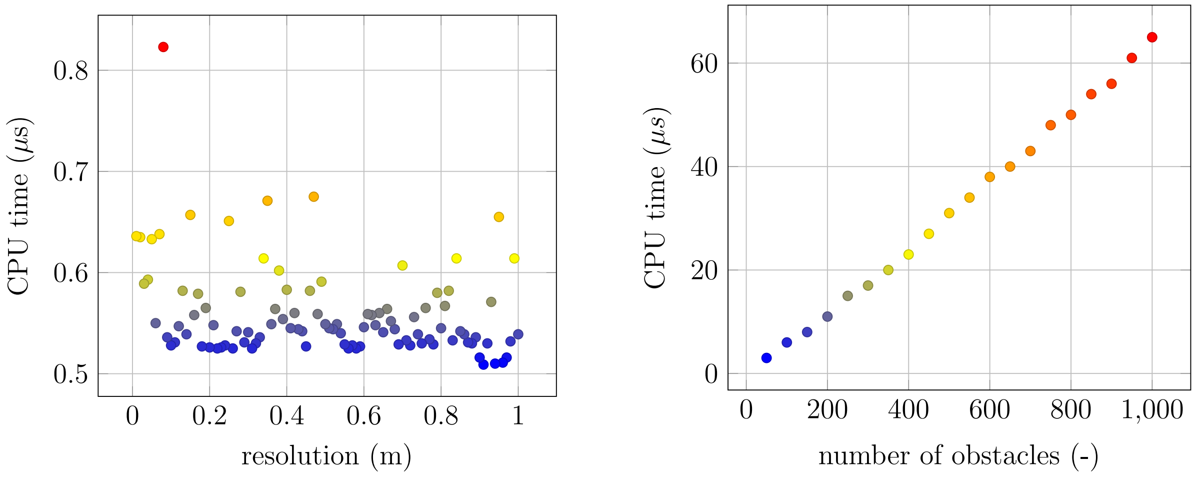
Let us firstly marked the configuration of the j-th robot at time t as

$$\psi_j(t) = \{P_j(t), O_j(t)\}, \tag{3.9}$$

where $P_j(t)$ are variables describing the position of the robot in $\mathcal{C}$ and the $O_j(t)$ corresponds to the orientation of camera or light, that is supposed to be independent on the control of robot position. Next, we define the sequence of robot configurations at particular transition points on the receding horizon with length N as

$$\Psi_j(t) = \{\psi_j(t + kT_s) | k \in \{1, 2, \ldots, N\}\}, \tag{3.10}$$

where T_s is the time difference between two consecutive transition points.



(a) method using octree

(b) original method

Figure 3.5: The time demands of methods used for finding the distance from the nearest obstacle. The presented time equals to one run of the algorithm with length of the planning horizon $N = 12$. The results were obtained by running the algorithms 10000 times with the usage of data from the church in Chlumin.

In a similar manner we mark the set of control inputs of j-th robot at time t as

$$u_j(t) = \{u_{j,p}(t), u_{j,o}(t)\}, \tag{3.11}$$

where $u_{j,p}(t)$ stands for the positional control inputs and $u_{j,o}(t)$ for control inputs responsible for control of camera or light orientation. With the use of this notation, we can compose the sequence of sets of control inputs for all segments between particular transition points on receding horizon with length N as

$$\mathcal{U}_j(t) = \{u_j(t + kT_s)|k \in \{1, 2, \ldots, N\}\}. \tag{3.12}$$

Due to the assumption on the independence of positional control and control of orientation, we can solve the task of optimization of the position of the robot and its orientation separately and thus reduce the number of decision variables. For these reasons we further divide each of the sequence $\mathcal{U}_j(t)$ and $\Psi_j(t)$ into two separate parts defined as

$$\begin{aligned}
\Psi_{j,p}(t) &= \{P_j(t + kT_s)|k \in \{1, 2, \ldots, N\}\},\\
\Psi_{j,o}(t) &= \{O_j(t + kT_s)|k \in \{1, 2, \ldots, N\}\},\\
\mathcal{U}_{j,p}(t) &= \{u_{j,p}(t + kT_s)|k \in \{1, 2, \ldots, N\}\},\\
\mathcal{U}_{j,o}(t) &= \{u_{j,o}(t + kT_s)|k \in \{1, 2, \ldots, N\}\}.
\end{aligned} \tag{3.13}$$

In the following sections, we use a discrete time indexing to reference values of variables at times corresponding to particular transition points. This indexing is defined as

$$G(k) := G(t + kT_s), k \in \{0, 1, \ldots, N\}, \tag{3.14}$$

where $G(\cdot)$ is an arbitrary variable and t is the current time.

3.4.1 Positional control

With the use of the above-described variables, we can define the process of finding the optimal sequence of control inputs $\mathcal{U}_{j,p}(t)$ on the horizon of length N as the generally nonlinear constrained optimization task with the objective function $J_{j,p}$ and set of nonlinear constraints $g_{j,p}(\cdot)$

$$\mathcal{U}_{j,p}(t)^* = \arg\min J_{j,p}(\mathcal{U}_{j,p}(t)),\ s.\ t.\ g_{j,p}(\mathcal{U}_{j,p}(t), \Psi_m(t), \mathcal{O}(t)) \leq 0, m \in R_p \tag{3.15}$$

where R_p is the set of indices of all robots participating in the mission and $\mathcal{O}(t)$ is the set of all obstacles that can be mathematically described by equation

$$\mathcal{O}(t) = \mathcal{O}_s \cup \{P_m(t) | m \in R_p \backslash j\}, \tag{3.16}$$

where $\mathcal{O}_s$ is the set of static obstacles and the second part of the union represents the current positions of other robots in the formation. The information about this positions is provided either by sharing the positions of particular robots in the formation within a global map through the communication channel or by the possibly available system for the relative localization.

The objective function $J_{j,p}(\cdot)$ can be split into several parts in the following way

$$J_{j,p} = \alpha J_{j,position} + \beta J_{j,control} + \gamma J_{j,obstacles} + \delta J_{j,occlusion} + \eta J_{j,trajectories}, \tag{3.17}$$

where $J_{j,position}$ stands for the part penalizing the deviations from the desired trajectory, $J_{j,control}$ is the part penalizing the changes in sequence of control inputs, $J_{j,obstacles}$ responds for the penalization of trajectories in the proximity of obstacles, the value of $J_{j,p}$ for trajectories that are near trajectories of other robots is increased with addend $J_{j,trajectories}$, and $J_{j,occlusion}$ penalizes the solutions that leads to occlusions caused either by obstacles or by followers carrying the light sources. Coefficients $\alpha, \beta, \gamma, \delta$ and η are used for scaling of particular parts of the objective function.

In the similar manner, the set of nonlinear constraints $g_{j,p}(\cdot) \leq 0$ can be break down into following constraints

$$\begin{aligned} g_{j,controls}(\mathcal{U}_{j,p}(k)) &\leq 0, \forall k \in \{1, ..., N\}, \\ g_{j,obstacles}(P_j(k), \mathcal{O}(t)) &\leq 0, \forall k \in \{1, ..., N\}, \\ g_{j,trajectories}(P_j(k), P_m(k)) &\leq 0, m \in R_p \backslash j, \forall k \in \{1, ..., N\}, \\ g_{j,occlusion}(P_j(k), \mathcal{O}(t), \psi_L(k)) &\leq 0, \forall k \in \{1, ..., N\}, \end{aligned} \tag{3.18}$$

where $g_{j,controls}(\cdot)$ includes the limitations on control inputs, $g_{j,obstacles}(\cdot)$ defines the unfeasibility of trajectories colliding with obstacles, $g_{j,trajectories}(\cdot)$ represents the constraints eliminating the solutions resulting in collision with trajectories of other robots, and $g_{j,occlusion}(\cdot)$ complements the objective function for occlusion $J_{j,occlusion}$. The exact definitions of particular parts of objective function $J_{j,p}(\cdot)$ and constraint function $g_{j,p}(\cdot)$ are provided in the following sections.

Position deviations

The first part of the objective function penalizes the deviation of the planned robot position from its desired position at a certain time. Although it is possible to use various metrics, the most proper for our application is the Euclidean distance. Therefore, we define the $J_{j,position}$ as

$$J_{j,position} := \sum_{k=1}^{N} ||P_j(k) - P_{d,j}(k)||^2, \tag{3.19}$$

where $P_j(k)$ is the planned position of j-th robot at time corresponding to k-th transition point of the planning horizon and $P_{d,j}(k)$ is an appropriate desired position of j-th robot. The graph of an example of the dependence of values of $J_{j,position}$ on x, y coordinates is shown in Figure 3.7a.

Obstacle avoidance

The part of the objective function, that penalizes the solutions in the proximity of obstacles is given by equation

$$J_{j,obstacles} := \sum_{k=1}^{N} k \left(\min \left\{ 0, \frac{\mathrm{dist}(P_j(k), \mathcal{O}(t)) - r_{s,o}}{\mathrm{dist}(P_j(k), \mathcal{O}(t)) - r_{a,o}} \right\} \right)^2, \tag{3.20}$$

where function dist$(\cdot)$ returns a distance between the position given by its first argument and the nearest object from the set of positions provided as the second argument. Variable $r_{s,o}$ stands for the detection radius of robot with respect to the set of obstacles $\mathcal{O}(t)$ and $r_{a,o}$ marks the critical avoidance radius of robot with respect to the obstacles. These two variables are used to define the radius around the robot, in that the presence of an obstacle should be penalized and the radius in which the presence of any obstacle results in the unfeasible trajectory. This part of the objective function was originally presented in [26] and its proper functionality is conditioned by

$$r_{s,o} \geq r_{a,o}. \tag{3.21}$$

The graph of an example of the dependence of values of $J_{j,obstacles}$ on x, y coordinates is shown in Figure 3.7b.

The obstacle avoidance part of the optimization task is completed by the inequality constraint

$$g_{j,obstacles}(P_j(k), \mathcal{O}(t)) \leq 0, \forall k \in \{1, ..., N\}, \tag{3.22}$$

where the function $g_{j,obstacles}(\cdot)$ is defined as

$$g_{j,obstacles}(P_j(k), \mathcal{O}(t)) := r_{a,o} - \mathrm{dist}(P_j(k), \mathcal{O}(t)). \tag{3.23}$$

By introducing equation (3.23), all trajectories, that consist of one or more transition points within the radius $r_{a,o}$ of the nearest obstacle, are made unfeasible.

Trajectories avoidance

The objective function for avoidance to trajectories of other robots is defined in a similar way as for the obstacle avoidance, only the set of obstacles is replaced by planned positions of other robots in consequent transition points. Thus, the value of $J_{j,trajectories}$ is computed according to the equation

$$J_{j,trajectories} := \sum_{k=1}^{N} \left(\min \left\{ 0, \frac{\min\limits_{m \in \{1,2,\dots,j-1\}} \{||P_j(k) - P_m(k)||\} - r_{d,r}}{\min\limits_{m \in \{1,2,\dots,j-1\}} \{||P_j(k) - P_m(k)||\} - r_{a,r}} \right\} \right)^2, \tag{3.24}$$

where $r_{d,r}$ is the detection radius with respect to other robots and $r_{a,r}$ is the avoidance radius with respect to other robots. This objective function is also accompanied by corresponding constraint function $g_{j,trajectories}$ determined by equation

$$g_{j,trajectories}(P_j(k), P_m(k)) := r_{a,r} - \min_{m \in \{1,2,\dots,j-1\}} \{||P_j(k) - P_m(k)||\}. \tag{3.25}$$

This condition has to hold throughout all transition points.

In the above-defined equation, we use trajectories of robots with an index lower than the index of the j-th robot, so that it can be understood as the priority of robot, that is defined in a way that the lower index, the higher priority a robot has. The reason for introducing this concept is to avoid mutual avoidance of trajectories of robots, that results in getting stuck in narrow passages due to the inability of robots to perform cooperative flight through these passages.

Control inputs

Since smooth trajectories without fast changes in control inputs are preferred within our application, the objective function $J_{j,controls}$ is defined as

$$J_{j,controls} := \sum_{d \in \{x,y,z\}} \left[(v_{d,j}(1) - v_{d,j}(0))^2 + \sum_{k=2}^{N} (v_{d,j}(k) - v_{d,j}(k-1))^2 \right]. \tag{3.26}$$

To achieve the feasibility of generated optimized trajectories according to the motion capabilities of employed robots, the set of inequality constraints should include part, that establishes the limitations on control inputs. Since our application expects low velocities in comparison of abilities of employed robots, we introduce the limitations on control inputs to limit the speed of the motion of particular robots, which also ensures the feasibility of generated trajectory. Thus, we define the constraint function $g_{j,controls}$ as

$$g_{j,controls}(u_{j,p}(k)) := \begin{pmatrix} v_{x,j}(k) - v_{x,j,max} \\ v_{x,j,min} - v_{x,j}(k) \\ v_{y,j}(k) - v_{y,j,max} \\ v_{y,j,min} - v_{y,j}(k) \\ v_{z,j}(k) - v_{z,j,max} \\ v_{z,j,min} - v_{z,j}(k) \end{pmatrix}, \tag{3.27}$$

where $v_{x,j,min}, v_{x,j,max}, v_{y,j,min}, v_{y,j,max}, v_{z,j,min}, v_{z,j,max}$ are lower and upper bounds of corresponding control inputs.

Occlusion

The occlusion part of the objective function is the only one that significantly differs for the leader and the followers. While the leader should try to avoid the occlusion caused by the presence of obstacles within the camera field of view (FoV), the followers should aim to avoidance of its own presence inside the FoV of the camera.

In the case of the leader $J_{j,occlusion}$ can be expressed as

$$J_{j,occlusion} := \sum_{k=1}^{N} \left(\min\left\{0, ||P_{OoI}(k) - P_{d,j}(k)|| - ||P_{OoI}(k) - P_j(k)||\right\}\right)^2, \tag{3.28}$$

where $P_{OoI}(\cdot)$ stands for the position of an object of interest. By adding the function $J_{j,occlusion}$ to the objective function $J_{j,p}$, the solutions that result in flying further from the OoI than in the desired distance, are penalized. Therefore, if some obstacle is present near the desired trajectory and can be safely avoided by flying between OoI and this obstacle and also behind the obstacle, defined $J_{j,occlusion}$ helps to prefer the solution without the occlusion. Nevertheless, its aim is not to ensure occlusion-free trajectories, which is often in contradiction with ensuring of collision-free trajectories. The graph of an example of the dependence of values of $J_{j,occlusion}$ of the leader on x, y coordinates is shown in Figure 3.7c.

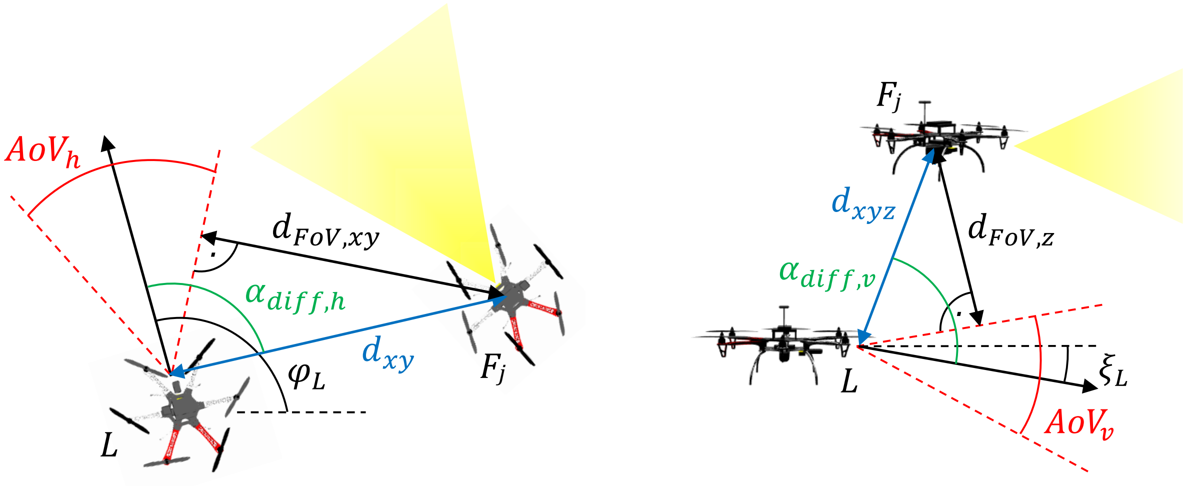
The $J_{j,occlusion}$ for followers is defined as

$$J_{j,occlusion} = \sum_{k=1}^{N} \left(\min\left\{0, \frac{d_{j,FoV}(k) - r_{d,FoV}}{d_{j,FoV}(k) - r_{a,FoV}}\right\}\right)^2, \tag{3.29}$$

where $r_{d,FoV}$ and $r_{a,FoV}$ are detection and avoidance radius of j-th robot with respect to camera FoV, and $d_{j,FoV}(\cdot)$ stands for the distance from the nearest border of the FoV. This distance can be computed according to equations

$$
\begin{aligned}
d_{xy}(k) &= \sqrt{(x_L(k)-x_j(k))^2+(y_L(k)-y_j(k))^2},\\
\alpha_{diff,h}(k) &= \left|\operatorname{atan2}(y_j(k)-y_L(k), x_j(k)-x_L(k))-\varphi_L(k)\right|,\\
\alpha_{diff,v}(k) &= \left|\operatorname{atan2}(z_j(k)-z_L(k), d_{xy}(k))-\xi_L(k)\right|,\\
d_{FoV,xy}(k) &= d_{xy}(k)\sin\left(\alpha_{diff,h}(k)-\frac{AoV_h}{2}\right),\\
d_{FoV,z}(k) &= \sqrt{d_{xy}(k)^2+(z_L(k)-z_j(k))^2}\sin\left(\alpha_{diff,v}(k)-\frac{AoV_v}{2}\right),\\
d_{j,FoV}(k) &= \begin{cases} \sqrt{d_{FoV,z}(k)^2+d_{FoV,xy}(k)^2}-r_d & \text{iff} \quad \alpha_{diff,h}(k) \le \left(\frac{\pi}{2}+\frac{AoV_h}{2}\right) \text{ and} \\ & \phantom{\text{iff} \quad} \alpha_{diff,v}(k) \le \left(\frac{\pi}{2}+\frac{AoV_v}{2}\right), \\ \sqrt{d_{xy}(k)^2+(z_j(k)-z_L(k))^2}-r_d & \text{else,} \end{cases}
\end{aligned}
\tag{3.30}
$$

where $d_{FoV,xy}(\cdot)$ is the distance to the nearest vertical border of FoV, $d_{FoV,z}(\cdot)$ is the distance to the nearest horizontal border of FoV, $\alpha_{diff,h}(\cdot)$ and $\alpha_{diff,v}(\cdot)$ stands for the angle between the nearest vertical respectively horizontal border of the FoV and connecting line between the leader and the j-th follower, and r_d marks the radius of the j-th robot. The graphical illustration of meaning of particular symbols is provided in Figure 3.6.



(a) top view

(b) side view

Figure 3.6: Graphical illustration of meaning of particular symbols used in equations (3.30) for computation of part of the objective function penalizing the occlusion caused by followers.

By addition of $J_{j,occlusion}$ defined in equations (3.29) and (3.30) to objective function for positional control, the FoV is introduced as another dynamic obstacle for followers.

The graph of an example of the dependence of values of $J_{j,occlusion}$ of the follower on the x, y coordinates is shown in Figure 3.7d. To complete the occlusion avoidance part of the optimization task for followers, we have to define the inequality constraint

$$g_{j,occlusion}(P_j(k), \psi_L(k)) \leq 0, \forall k \in \{1, ..., N\}, \tag{3.31}$$

where the value of function $g_{j,occlusion}(\cdot)$ is computed according to equation

$$g_{j,occlusion}(P_j(k), \psi_L(k)) := r_{a,FoV} - d_{j,FoV}(k). \tag{3.32}$$

3.4.2 Orientation control

In a similar manner as for the positional control, we can define the process of finding the optimal sequence of control inputs $\mathcal{U}_{j,o}(t)$ on the horizon of length N as the quadratic constrained optimization task with the objective function $J_{j,o}$ and set of nonlinear constraints $g_{j,o}(\cdot)$

$$\mathcal{U}_{j,o}(t)^* = \arg\min J_{j,o}(\mathcal{U}_{j,o}(t)), \; s.\; t. \; g_{j,o}(u_{j,o}(k), O_j(k)) \leq 0, \forall k \in \{1, ..., N\}. \tag{3.33}$$

The objective function $J_{j,o}(\cdot)$ consists of two parts and is defined as

$$J_{j,o} = \zeta J_{j,orientation} + \kappa J_{j,control}, \tag{3.34}$$

where $J_{j,orientation}$ is the part penalizing the deviation from desired orientation of camera, $J_{j,control}$ stands for the part penalizing the fast changes in consequent control inputs $u_{j,o}(\cdot)$, and ζ and κ are coefficients used for scaling of particular parts of the objective function $J_{j,o}$. The set of nonlinear constraints $g_{j,o}(\cdot) \leq 0, \forall k \in \{1, ..., N\}$ can be split into following constraints

$$\begin{aligned} g_{j,control}(u_{j,o}(k)) \leq 0, \forall k \in \{1, ..., N\}, \\ g_{j,orientation}(O_j(k)) \leq 0, \forall k \in \{1, ..., N\}, \end{aligned} \tag{3.35}$$

where $g_{j,control}(\cdot)$ stands for the constraints introducing the limits on control inputs, and $g_{j,orientation}(\cdot)$ introduces the limitations on angles $\varphi_j(k)$ and $\xi_j(k)$ which defines the orientation of the camera or light.

The task of orientation control is simple in comparison to the task of positional control due to the fact that the robot cannot get into a collision with any object in the environment or cause occlusion by only changing its orientation. The parts of the mentioned objective function and constraint functions are briefly described in the following sections.

Deviations of orientation

The part of the objective function responsible for the penalization of deviations from desired orientation is given by equation

$$J_{j,orientation} := \sum_{k=1}^{N} \left[(\varphi_j(k) - \varphi_{d,j}(k))^2 + (\xi_j(k) - \xi_{d,j}(k))^2 \right], \tag{3.36}$$

where $\varphi_{d,j}(\cdot)$ and $\xi_{d,j}(\cdot)$ are angles defining the desired orientation of camera or light.

The inequality constraint function $g_{j,orientation}(\cdot)$ is introduced to limit the allowed orientation of camera or light given by the construction of particular UAVs. Since the angle $\varphi_j(\cdot)$ has to be unlimited to allow the general deployment of the method, the only limited angle is the $\xi_j(\cdot)$. Hence the $g_{j,orientation}(\cdot)$ is defined as

$$g_{j,orientation}(u_{j,o}(k)) := \begin{pmatrix} \xi_j(k) - \xi_{j,max} \\ \xi_{j,min} - \xi_j(k) \end{pmatrix}, \tag{3.37}$$

where $\xi_{j,min}$ is the minimum allowed value of $\xi_j(k)$ and $\xi_{j,max}$ is the maximum allowed value of $\xi_j(k)$.

Control inputs

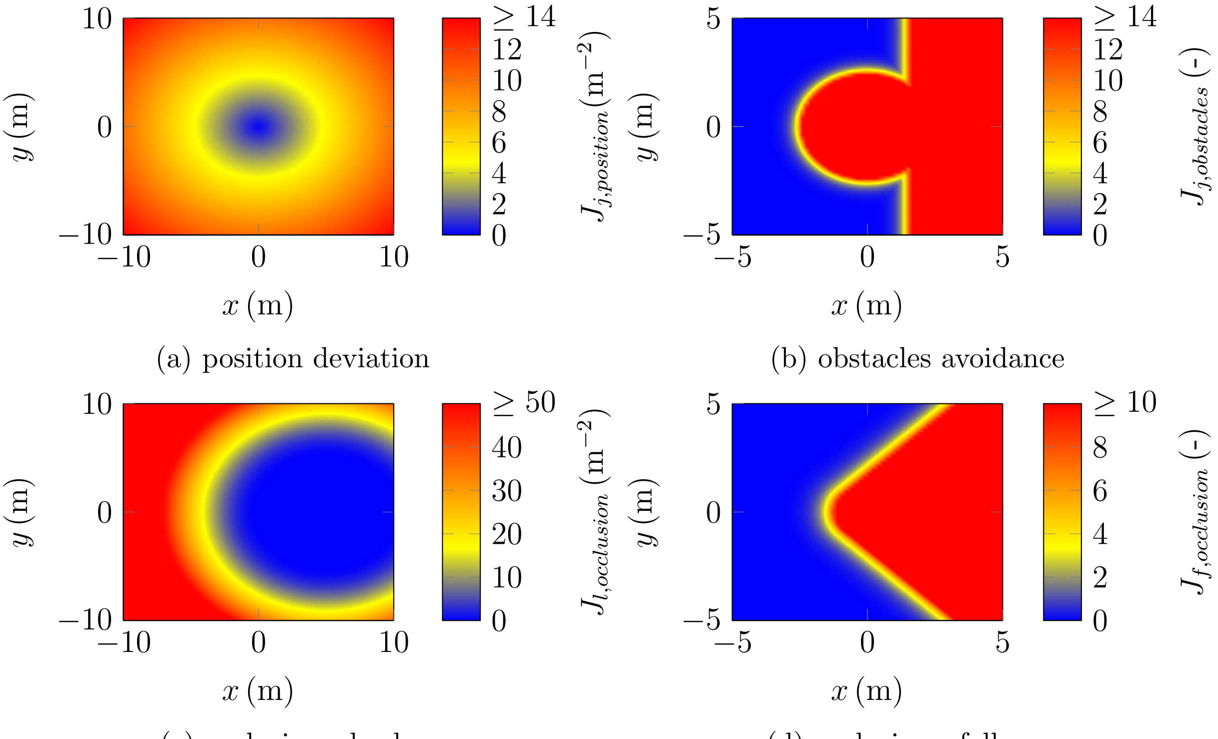
The part of the objective function for the penalization of fast changes in the sequence of control inputs $\mathcal{U}_{j,o}$ is defined in a similar way as for the positional control inputs. Thus the value of $J_{j,controls}$ is computed according to equation

$$J_{j,controls} := \sum_{k=1}^{N} \left[(\omega_j(k) - \omega_j(k-1))^2 + (\varepsilon_j(k) - \varepsilon_j(k-1))^2 \right]. \tag{3.38}$$

The constraint function $g_{j,controls}(\cdot)$ represents the limitations on control inputs introduced to restrict the maximum speed of turning. The value of $g_{j,controls}(\cdot)$ is computed according to equation

$$g_{j,controls}(u_{j,o}(k)) := \begin{pmatrix} \omega_j(k) - \omega_{j,max} \\ \omega_{j,min} - \omega_j(k) \\ \varepsilon_j(k) - \varepsilon_{j,max} \\ \varepsilon_{j,min} - \varepsilon_j(k) \end{pmatrix}, \tag{3.39}$$

where $\omega_{j,min}, \omega_{j,max}, \varepsilon_{j,min}, \varepsilon_{j,max}$ are lower and upper bounds of corresponding control inputs. These limitations can be expressed as the kinematic constraints of deployed UAVs representing their motion capabilities. Nevertheless, as in the case of positional control, aggressive turning manoeuvres are undesirable. Therefore, the bounds on particular control inputs are set below the capabilities of applied UAVs and thus they represent the maximum allowable angular rates of UAV, camera and light.



(a) position deviation

(b) obstacles avoidance

(c) occlusion - leader

(d) occlusion - followers

Figure 3.7: Graphs of parts of the objective function $J_{j,position}$. The graph of the position deviation part shows the value of $J_{j,position}$ for particular combinations of x, y coordinates for the desired position placed at $P_d = (0,\, 0)^T$. The graph of the obstacles penalization part shows the value of $J_{j,obstacles}$ for particular combinations of x, y coordinates for the circular obstacle with radius equal to 1 m placed at position with coordinates $x = 0$, $y = 0$ and the wall represented by equation $x = 2$. The graph of the leader occlusion part shows the value of $J_{j,occlusion}$ for particular combinations of x, y coordinates for the desired position $P_d = (0,\, 0)^T$ and the position of OoI $P_{OoI} = (5,\, 0)^T$. The graph of the follower occlusion part shows the value of $J_{j,occlusion}$ for particular combinations of x, y coordinates for the position of the leader $P_L = (0,\, 0)^T$ and the orientation of camera $\varphi_L = 0\,\text{rad}$. All graphs presented in this figure are computed without considering the values of objective functions connected with z coordinate.

3.5 Comparison of solvers

The method for formation control originally proposed in [1] and later improved within this thesis is based on the model predictive control concept well known in the field of control theory. In Section 3.4 we have redefined the objective function, which should be minimal for the optimal sequence of control inputs $\mathcal{U}_j(t)$ satisfactory to all defined constraints.

The usual way how to obtain this optimal sequence is to apply some optimization solver. The issue of our optimization task defined in Section 3.4 is that it is generally nonlinear and non-convex. Therefore we cannot employ modern fast linear and quadratic solvers like Gurobi [27], CPLEX [28] or CVXGEN [29] (code generator for convex optimization) and so we have to choose among the available solvers for non-linear optimization problems.

The need to solve the optimization task onboard in time not exceeding nT_s (mostly equal to 400 ms), where n is the control horizon (number of applied control inputs from the sequence $\mathcal{U}_j(t)$), determines the computational time as the most crucial parameter for the choice of the ideal solver. Although a wide range of the nonlinear solvers exists, most of them are not able to solve our optimization task within the time nT_s. In particular, based on our survey we have chosen and tested four nonlinear optimization solvers - IPOPT [30], NLopt [31], SNOPT [32] and CFSQP [33]. During the tests, we deploy particular solvers for solving our simplified optimization task, containing only the first three summands of equation (3.17), on the horizon with length $N = 12$. Based on the measured computational time we find out that even for this simplified task, only the CFSQP solver is able to meet desired time constraints. Our results correspond to complex survey and testing of the nonlinear optimization solvers published in [34], where the authors compare the performance of seven different nonlinear solvers including CFSQP, IPOPT and NLopt on the Hock-Schittkowski optimization test suite.

Since the global optimization of general nonlinear problems is a really difficult and time-consuming process, we cannot utilize the solvers, which attempt to guarantee to find the global minimum. Therefore, based on the results presented in the previous paragraph, we decided to use the CFSQP solver. We are aware of the possibility that this solver can get stuck in local minimum or even find no solution satisfying the constraints. Although the situations when the solver provides unfeasible solution are rare, they are detected and adequately resolved within the formation control method and further by the mission controller described in Chapter 4.

Chapter 4

Mission controller

Our so-called mission controller is the separate part of the system independent of any other robot in the formation, which is running on a standalone computer. Despite its name, it has several functions. The first, clear and main purpose is to control the cooperative carrying out of the mission. The second one is to provide the interface for the operator, so that he can get the necessary information about the state of the formation and particular robots and also actively influence the behaviour of the formation in the course of the mission, including sending the request to land, pause the mission etc. Its last, but not less important, functionality is the detection of system faults and failures and ensuring of safe deterministic reaction on their occurrence. The whole proposed mission controller is implemented as the finite state machine.

In this chapter, we consecutively describe the mission controller for expected behaviour without fault occurrence and the design of the safety part of the mission controller including the analysis of potential faults and failures. The explanation of particular symbols used in the figures within this chapter is provided in Figure 4.1.

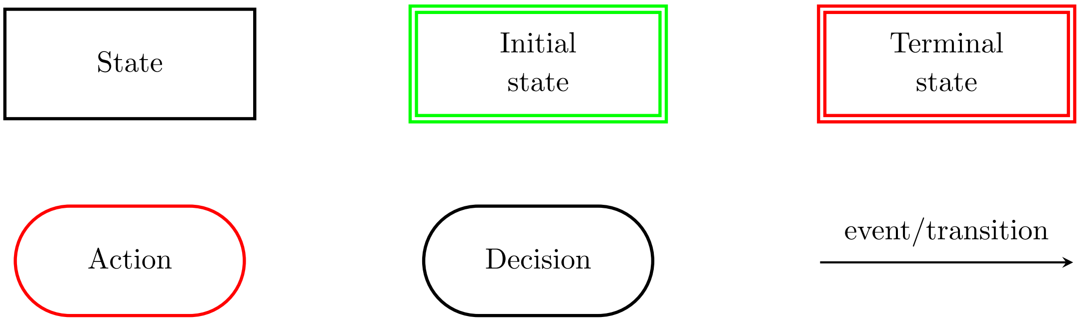


Figure 4.1: Explanation of symbols used in the figures within Chapter 4 in which the mission controller is described.

4.1 Mission controller for normal operation

The part of the mission controller, which is responsible for the control of the normal operation, consists of 14 states and 24 events. To provide a more comprehensible description, we have divided it into three separate sections - takeoff, proper scanning mission and landing. The whole mission controller for normal operation is graphically illustrated in Figure 4.2 along with Table 4.1, which provides the explanatory notes for events used in the figure.

Takeoff

The mission controller starts in the *Initialization* state. After completion of all initialization procedures, it switches to state *Waiting for planning nodes* in which it waits until it gets in touch with all defined robots in the formation and their corresponding systems for formation control.

When the mission controller already registers all members of the formation, the state is changed to *Waiting for takeoff*. In this state, the behaviour depends on the predefined requirements of the user. If the user requires an automatic takeoff, it is automatically called immediately when all members of the formation are in autonomous mode, and so they can be fully controlled by the onboard computer. If the automatic call of the takeoff procedure is undesirable, the mission controller waits until all formation members successfully reach their initial position after manually called takeoff. The initial position is placed in predefined height $h_{takeoff}$ above the starting position.

Scanning mission

After successful takeoff, the finite state automaton advances to state *Waiting in initial position*, in which the system waits for the activation signal for start of the predefined scanning mission. Based on the preferences, this signal can be again generated automatically or directly by the user. After successful activation, the state is switched to *Flying to trajectory start*. When all mission participants reach the initial position of the desired trajectory, the trajectory following is activated, and the state of the mission controller is changed to *Trajectory following*. Within this state, the UAVs are following the trajectories given by the initial scanning plan and leader-follower scheme with the use of the proposed system for safe trajectory tracking described in Chapter 3.

During the trajectory following, the formation can switch to performing Reflectance Transformation Imaging scanning (RTI scanning phase) in predefined parts of the trajectory or in case that the user sends the request to start RTI scanning immediately. The transition to state *RTI phase running* is followed by the stop of the leader at the current position, at which it hovers with the use of the formation control method by setting all transition points on the receding horizon to current position saved at the beginning of RTI scanning phase. The follower chosen as the carrier of light source for the RTI scanning phase generates the desired RTI trajectory (this process is described in Chapter 5 in details) and flies among particular lighting positions while the leader consequently takes the photos of OoI under various lighting conditions. After finishing the RTI scanning phase, the mission controller switches back to the *Trajectory following* state and the formation continues in performing the scanning plan.

In both *Trajectory following* and *RTI phase running* states, the user has the opportunity to pause, reactivate or possibly restart the mission. When the mission is paused, all robots in the formation are safely hovering at their current positions with the benefit of the formation control method presented in Chapter 3. After reactivation, the mission controller turns from the *Planning paused* state back to the original state (either *Trajectory following* or *RTI phase running*) and proceeds in scanning mission. If the scanning mission is reset by the user, the state machine switches to the *Waiting for activation* state in which it again safely hovers until the activation signal is received. Then the mission controller switched to *Flying to trajectory start*, the whole formation moves to the trajectory start, and the scanning mission is restarted.

Once the mission is completed the mission controller advances to state *Waiting in final position*. From this state, it either switches to the *Waiting for activation*, if the restart of the mission after its completion is required or starts the procedure which results in landing.

Landing

The landing process is again influenced by preferences, which has to be set by the user prior to the mission. Basically, there are two options regarding the landing place and two choices regarding the control of the UAV.

In a matter of the landing place, we suppose that common desired landing places are the initial position and the final position. The advantage of landing at an initial position consists of the fact that the place which is safe for takeoff is also safe for landing. On the other hand, the advantage of landing at final position is that it does not require any additional flight time and energy for returning back into the initial position. Thus, based on the preferences of a particular application, the UAVs after finishing of the scanning mission either stay at the final position or they fly back to the initial position by safely following the trajectory to initial position. This trajectory is generated by the sampling of the direct segment between the final and initial position without considering the presence of obstacles. The trajectory tracking is then achieved with the use of the presented formation control method omitting the last addend, which penalizes the occlusion. By applying this approach, the optimized generated trajectories are collision-free, but in the case of existence of large obstacles on the initial trajectory, the method can possibly fail in leading the formation to the desired initial position. Nevertheless, in this case, the proposed mission controller detects this failure and informs the human operator about the demand for manual control.

As regards the way of control during landing, we can distinguish between autonomous landing or manually controlled landing. In the case of requirement on autonomous landing, the state machine switches from the *Waiting in initial position* or *Waiting in final position* to *Landing* state in which it sends the sequence of landing commands to all members of the formation. When the mission controller detects successful landing of all mission participants, it turns to *Mission finished* state, which is the terminal state of the presented state machine. In case of preferred manually controlled landing, the mission controller switches from the *Waiting in initial position* or *Waiting in final position* to *Manual control required* state in which it produces the notifications to operators, that manual control of particular UAVs is required. If it detects that all UAVs are manually controlled, the state is changed to *Mission finished* state and the mission ends.

Although definitely not all places within the expected application environment are safe for landing, based on the given map of the environment we can conclude about the safety of places regarding the landing. Thus, we have implemented the method for checking whether some place is safe to land to allow the user to call the landing procedure at any time. By the time, when the landing request is detected, the mission controller stops the motion of UAVs and check whether the current position of every single UAV is safe for landing. If this condition is fulfilled and the autonomous landing is required, the landing procedure is started. If not, it informs the operators about the safety of the position according to landing and the requirement on manual control. The graphical interpretation of the landing part of the mission controller is shown in Figure 4.3.

4.2 Faults and failures analysis

Systems for detection, prevention and handling of faults are important parts of all robotic systems deployed in real-world environments. For our proposed system, this importance is even increased by the danger of destroying priceless historical objects in case of a system collapse. The necessary part of designing such a system is the analysis of all potential faults and failures, which is described in this section.

As was already mentioned, the system is built above the MRS system for control of multi-rotor helicopters, which handles most of the hardware faults like broken rotor or missing sensor. Thus the proposed mission controller is mainly oriented on faults and failures mostly specific for formation control system described in Chapter 3 and uses the information provided by the MRS system as an input data.

Faults and failures (F&F) of the proposed system can be divided into several groups according to possible reactions to their occurrence. The first group of F&F includes the loss of communication between the mission controller and a UAV, failure of an onboard computer, an error of the MRS system for UAV control, unexpected breakdown of the mission controller or failure of the computer on which is the mission controller running. All these F&F result in absolute inability to autonomously control the behaviour of UAV and to detect the flying mode (autonomous or manual controlled), which means that the

Event	description of event or condition	auto	set	call
ϵ_0	initialization completed	✓	×	×
ϵ_1	planning methods running and takeoff allowed	✓	×	×
ϵ_2	takeoff detected (trajectory tracker active)	✓	×	×
ϵ_3	planning activated	✓	×	✓
ϵ_4	all robots achieved their starting positions	✓	×	×
ϵ_5	planning mission completed	✓	×	×
ϵ_6	planning paused	×	×	✓
ϵ_7	planning reactivated	×	×	✓
ϵ_8	planning reset	×	×	✓
ϵ_9	Reflectance transformation imaging scanning required	×	✓	✓
ϵ_{10}	Reflectance transformation imaging scanning finished	✓	×	×
ϵ_{11}	landing after completion of planning mission not required and planning reset	×	✓	✓
ϵ_{12}	automatic landing at final position required	×	✓	×
ϵ_{13}	landing at initial position required	×	✓	×
ϵ_{14}	all UAVs automatically landed	✓	×	×
ϵ_{15}	all UAVs are under manual control of operators	✓	×	×
ϵ_{16}	planning mission completed and manual landing required	✓	✓	×
ϵ_{17}	planning mission completed and automatic landing required	✓	✓	×
ϵ_{18}	all UAVs achieved their initial positions	✓	×	×
ϵ_{20}	landing after completion of planning mission required, landing at final position required, and manual landing required	✓	✓	×
ϵ_{21}	any kind of fault appears	✓	×	×

Table 4.1: Description of events and conditions used within the figure describing state machine for normal behaviour of the system (Figure 4.2). The columns auto, set, and call marks whether the event or condition is generated/detected automatically by the system (auto), set by the user prior the mission (set) or called by user during the mission (call).

mission controller cannot even check whether the faulty UAV is manually controlled or not. In this situation, the autonomous mission has to be ended, and control of all UAVs should be taken over by human operators.

The MRS system uses more sources of information (Global Positioning System (GPS), magnetometer, Inertial Measurement Units (IMUs), cameras, laser rangefinders, etc.), that are fused to obtain the more reliable information about position and orientation of the UAVs. The subsets of these sources build particular odometry estimators, which can be used based on their availability and preferences of the user. The F&F which makes the autonomous control impossible or dangerous includes insufficient precision of odometry, the complete absence of odometry information and disallowed type of odometry estimator (e.g. GPS and compass inside of the building). In the case of these F&F, the control of a faulty UAV has to be taken over by an operator. However, the mission controller is able to check that the action was successful and thus enable the safe carrying on the mission with the remaining UAVs in the formation.

The third kind of F&F are those which disable both the autonomous and manual flight modes. To this group belongs all kinds of hardware failures, insufficient battery voltage and exceeding the allowable flight time. As was already mentioned above, the hardware failures and faults are controlled outside the mission controller, designed in this work. The battery voltage is also controlled outside our system for mission control, but its including into the system enables not only to set simple threshold but also to check the battery voltage regarding the remaining part of the mission and distance to the desired landing point. The checking of the flight time is redundant and is used only as another safety mechanism in the case of failure of measuring the battery voltage.

Finally, the last group incorporates faults and failures which do not have an impact on the ability of multi-rotor UAVs to hover in the air. These are an unexpected shutdown of planning systems, the absence of solution of the optimization task for trajectory tracking, and too big distance from the desired position (the trajectory is not followed precisely and so it is dangerous to continue). These F&F enable safe autonomous control including autonomous landing or restart of a mission. Therefore they can be autonomously resolved and do not require human intervention.

4.3 Mission controller for faulty operation

Based on the above-described analysis of possible faults and failures of the system, we have designed the part of the mission controller which is responsible for the handling of faults. This part is entered whenever any fault is detected within the periodically called method. After the detection of fault, the state machine for faulty operation stops the path following of all mission participants. After that, the behaviour depends on the type of faulty robot (leader or follower) and the exact type of fault or failure.

In the case of lost communication between any two robots or robot and server, the mission is immediately stopped. Since the robots are localizing themselves within the global map and share their position via the communication channel, the loss of communication with one of the robot results in the absence of information about its position. The absence of this information then disables the usage of collision avoidance provided by formation control method. Therefore also the autonomous landing is not safe and manual control of all UAVs is required. Although the system for relative localization is supposed to be part of the system, it is not necessarily required, and so it cannot be introduced as the general independent source of information about the position of the robot.

As we suppose that the leader carries the only camera needed for taking of images, it does not make sense to carry on the mission without the leader. Therefore, every time the autonomous control of the leader is impossible or dangerous, the mission is ended and the immediate landing of all UAVs is desired. In case of missing or inappropriate solution of optimization task or too big distance from desired position, the fault is autonomously resolved by the mission controller and the mission can be restarted.

The behaviour of the system in the case of fault or failure of follower depends not only on the type of fault but also on the preferences of the user. Since some situations in which the images with imperfect lighting are also valuable exist, the fault or failure of follower excluding lost communication do not have to lead to a prompt end of the mission. Instead of that, the mission can be paused and the follower can autonomously land, or his control can be taken over by the human operator. Afterwards, the respective robot is removed from the set of mission participants, and the scanning mission can be reactivated. Otherwise, the reactions on faults and failures depending on their type are similar to the case of the leader. The part of the mission controller responding for the handling of faulty operation is graphically illustrated in Figure 4.4.

A special type of fault is the breakdown of the mission controller itself. It is used as the provider of information about the state of the formation, particular UAVs and potential F&F. Therefore, in case of its failure, human operators lose the main source of information about the system and control mechanism for F&F occurrence and so the mission cannot be safely carried on and should be ended. For these reasons, particular UAVs are checking the communication with the mission controller, and if they detect its loss, they stop the trajectory tracking, turn to hover at the current position and inform the human operators about the necessity of manual control. Since the only remaining source of information about the formation state is the visual contact with the UAVs, they are supposed to indicate the requirement on manual control by blinking with light emitting diodes of defined colour.

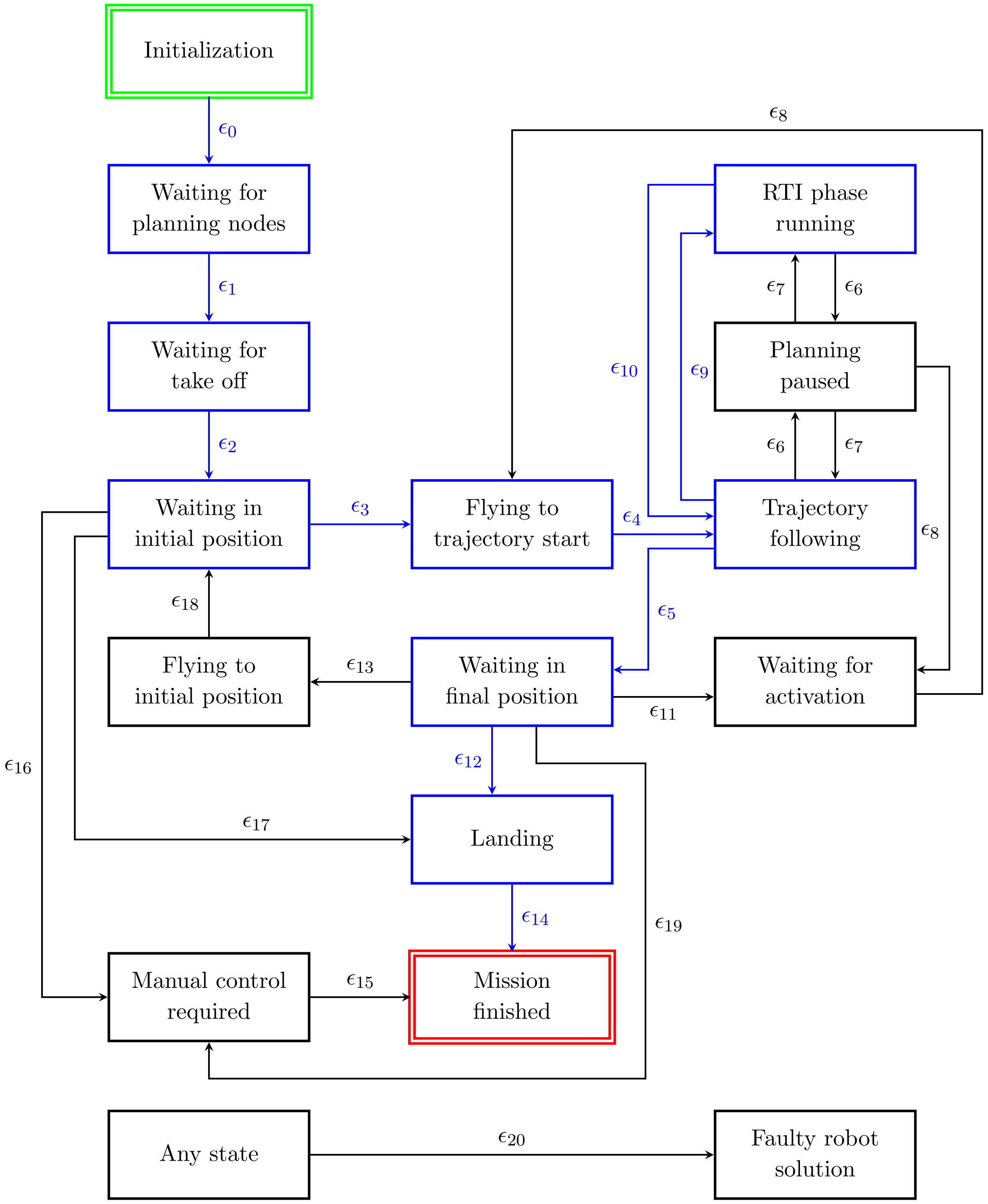


Figure 4.2: The mission controller for normal operation as the finite state machine. The ϵ_i stands for particular events or conditions which are briefly described in Table 4.1. The blue coloured states and arrows marks the most direct course of the mission.

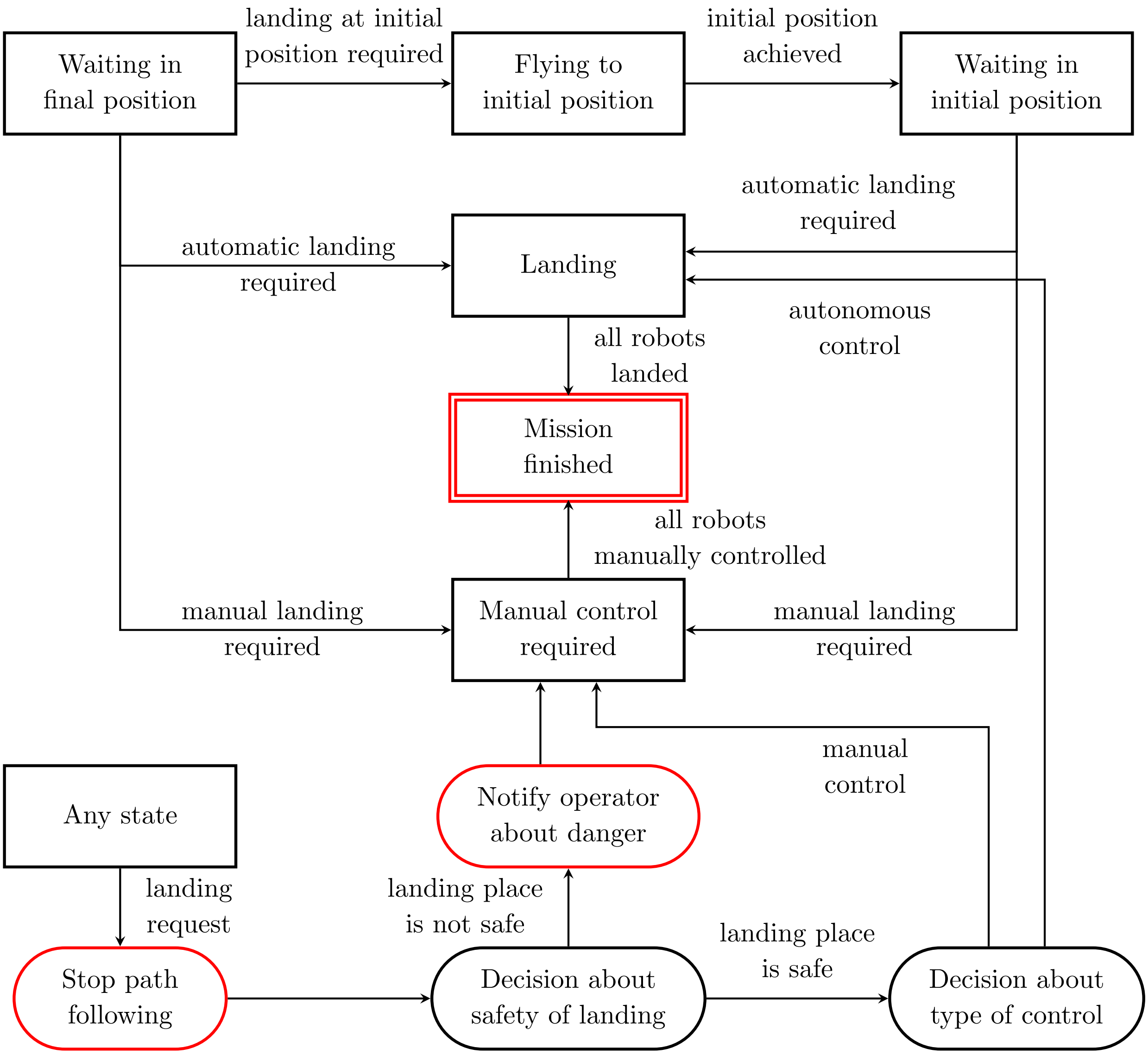
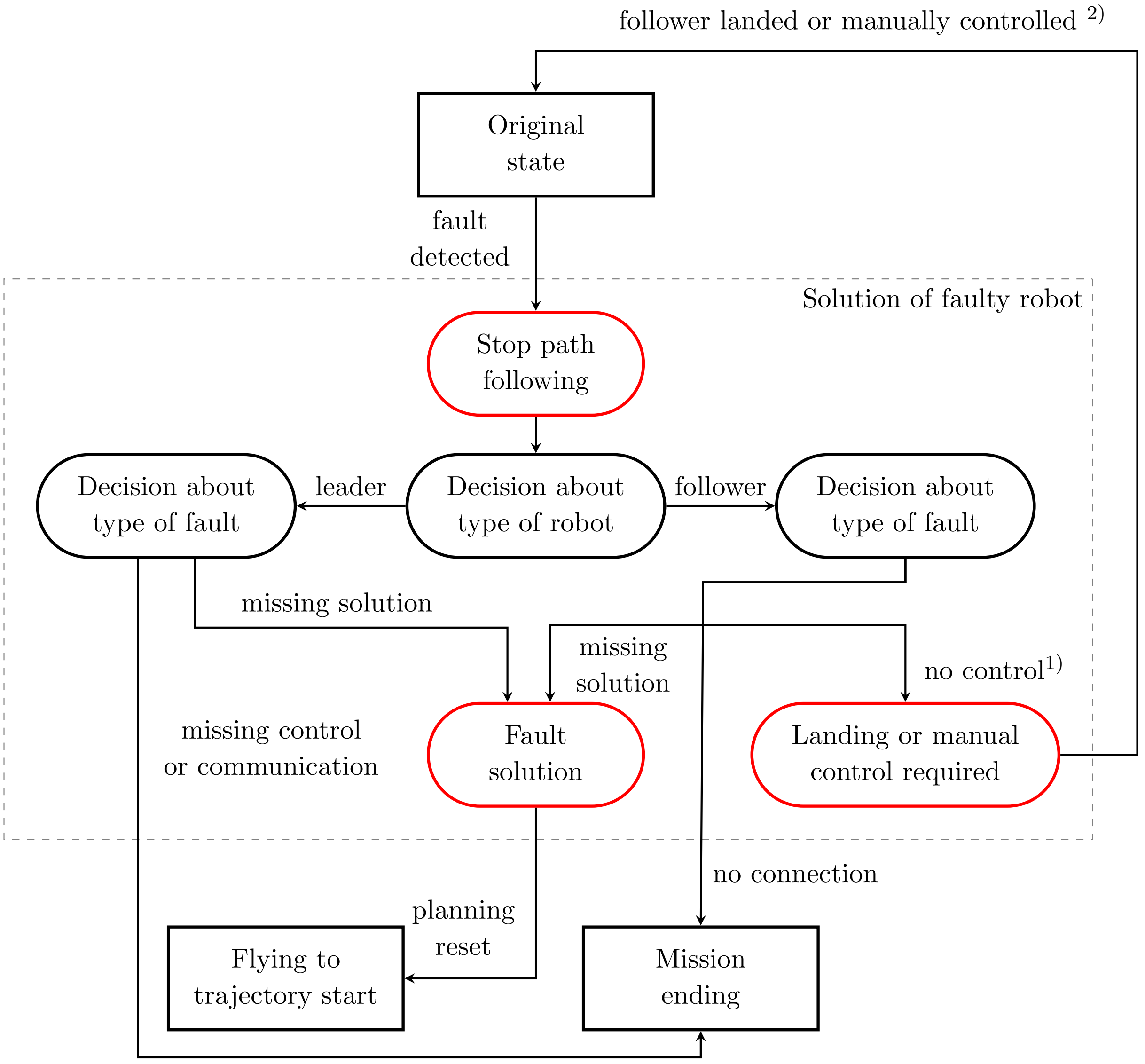

Waiting in final position
landing at initial position required
Flying to initial position
initial position achieved
Waiting in initial position
automatic landing required
Landing
automatic landing required
autonomous control
all robots landed
Mission finished
all robots manually controlled
manual landing required
Manual control required
manual landing required
manual control
Any state
Notify operator about danger
landing request
landing place is not safe
landing place is safe
Stop path following
Decision about safety of landing
Decision about type of control

Figure 4.3: Part of the mission controller responsible for control of landing.


follower landed or manually controlled 2)
Original state
fault detected
Solution of faulty robot
Stop path following
Decision about type of fault
leader
Decision about type of robot
follower
Decision about type of fault
missing solution
missing solution
no control1)
missing control or communication
Fault solution
Landing or manual control required
no connection
planning reset
Flying to trajectory start
Mission ending

[1] no control and following with faulty follower not required

[2] and following mission with faulty follower required

Figure 4.4: Part of the mission controller responsible for handling of faulty operation.

Chapter 5

Reflectance transformation imaging

5.1 Reflectance transformation imaging method

For the experts from the field of restoration and historical science is important to have the high-quality documentation of historical art pieces, which enables their analysis and planning of future restoration works without repetitive direct access to particular artefacts. The most common type of such documentation is a three dimensional model (in case of statues, buildings, etc.) or the set of images with different lighting conditions (in case of statues, but also paintings, frescos and mosaics) which enables to precisely inspect and analyse the surface of the captured object. The drawback of both these methods is that the needed number of taken pictures or scans is proportional to the required resolution measured in the lighting angles or the mesh density.

For the set of images, this drawback can be overcome with a method called Reflectance Transformation Imaging (RTI), which is used for obtaining the representation of an image that enables its displaying under arbitrarily lighting conditions. One type of such representation is a Polynomial Texture Map (PTM), which was originally developed by T. Malzbender [35]. While in the common representation, each pixel of the image has assigned three static values for red, green and blue colour (RGB), within the PTM, the intensity of each colour channel $I_{c,x,y}$, $c \in \{red,\ green,\ blue\}$ of the pixel at position (x, y), is represented by function

$$I_{c,x,y} = f(l_u, l_v), \tag{5.1}$$

where l_u and l_v are elements of lighting vector and the function $f(\cdot)$ is a second-order bi-quadratic polynomial function with varying coefficients $\alpha_{i,c}$ for particular pixels (x, y). Thus the intensity $I_{c,x,y}$ of each colour can be interpreted as

$$I_{c,x,y} = \alpha_{1,c}l_u^2 + \alpha_{2,c}l_v^2 + \alpha_{3,c}l_ul_v + \alpha_{4,c}l_u + \alpha_{5,c}l_v + \alpha_{6,c},\ c \in \{red,\ green,\ blue\}. \tag{5.2}$$

The input of the RTI method is the set of images taken by a static camera under varying known light conditions, where each image in the set has assigned corresponding lighting vector. Dependent on the used method, the lighting vectors are computed from known positions of light sources or from the reflections on the target object placed next to the captured object (so-called H-RTI method [36]). The main advantage of this approach in comparison to capturing the simple set of images is that the PTM can be produced from tens of images and despite this fact it provides the ability to view the captured object under arbitrarily lighting conditions.

The RTI method was widely used within the archaeological, historical, and restoration community in recent years [37, 38, 39, 40, 41], and it also belongs to subjects of interest of researchers in aforementioned scientific fields [42]. Further development of the RTI approach is supported by freely available software for complete RTI procedure provided mostly by Hewlett Packard company [43] and Cultural Heritage Imaging corporation [44]. This software is also used within this thesis for computing and displaying polynomial texture maps.

5.2 RTI scanning implementation

The implementation of the RTI scanning phase can be divided into three tasks - generating a set of desired light positions to achieve equally distributed lighting, determination of the optimal sequence of these positions and plan safe path connecting this sequence. The approaches which are described in the following sections are highly influenced by the requirements of experts from the field of restoration and historical science. Furthermore, common issues and properties of multi-rotor UAVs have to be as well taken into account during their deployment in real-world scenarios.

5.2.1 Set generation

Based on the theory of experts from the field of restoration and historical science, the optimal set of lighting vectors is consisted of equally distributed vectors within the range defined by minimum and maximum lighting angles in horizontal ($\lambda_{h,min}$, $\lambda_{h,max}$) and vertical ($\lambda_{v,min}$, $\lambda_{v,max}$) direction. The second requirement is that the intensity of lighting presented at the scanned object should be the same for all lighting directions.

Given these two requirements and assumption that the intensity of the light source is constant, we can determine that the desired positions of light sources will be distributed on the spherical cap of the sphere with the center located at the position of the scanning object. Therefore, this problem can be defined as the problem of even distribution of points on the sphere, which is not trivial and has an exact solution only for particular cases [45]. However, despite the fact that the exact even distribution of lighting vectors is optimal for obtaining PTM, the deflection from the optimal positions does not lead to a significant reduction of the quality of resulting PTM. Moreover, the exact solution is not optimal from our application point of view, since the shortest path connecting all points leads to an unstructured sequence. Therefore, we have adapted and slightly enhanced an approach used by restorers during the manual acquisition of images for RTI method [46].

Our method for obtaining the set of desired lighting positions uses as inputs the border lighting angles $\lambda_{d,m}$, $d \in \{h, v\}$, $m \in \{min, max\}$, the position of scanned object $P_{OoI}(t)$, the position of camera $P_{cam}(t)$, desired distance between the light and scanned object d_l, and desired number of samples of lighting angles v_s in vertical direction for which holds $v_s \geq 2$. The first step of the method is getting set of samples of vertical lighting angles Λ_v from interval $\langle \lambda_{v,max}, \lambda_{v,min} \rangle$ in such way that they are equally distributed over this interval, $|\Lambda_v| = v_s$, $\min(\Lambda_v) = \lambda_{v,min}$ and $\max(\Lambda_v) = \lambda_{v,max}$. For each sample λ_v from Λ_v is then constructed one spline on which possible positions of the light lie. These splines are parts of the circle and with assumption $P_{OoI} = \{x, y, z\} = \{0,\ 0,\ 0\}$ and heading of camera $\{\psi_{cam}, \zeta_{cam}\} = \{0, 0\}$ can be described by equations

$$\begin{aligned} x_s &= -d_l \cos(\lambda_v) \cos(\lambda_h),\ \ \lambda_h \in \langle \lambda_{h,min},\ \lambda_{h,max} \rangle \\ y_s &= -d_l \cos(\lambda_v) \sin(\lambda_h),\ \ \lambda_h \in \langle \lambda_{h,min},\ \lambda_{h,max} \rangle \\ z_s &= -d_l \tan(\lambda_v). \end{aligned} \tag{5.3}$$

Considering the arbitrary position of scanned object defined by $P_{OoI} = \{x_{OoI}, y_{OoI}, z_{OoI}\}$ and arbitrary heading of camera $\{\psi_{cam}, \zeta_{cam}\}$, the equation (5.3) should be rewritten to

$$\begin{aligned} x_s &= x_{OoI} - d_l \cos(\lambda_v + \zeta_{cam}) \cos(\lambda_h + \psi_{cam}),\ \ \lambda_h \in \langle \lambda_{h,min},\ \lambda_{h,max} \rangle \\ y_s &= y_{OoI} - d_l \cos(\lambda_v + \zeta_{cam}) \sin(\lambda_h + \psi_{cam}),\ \ \lambda_h \in \langle \lambda_{h,min},\ \lambda_{h,max} \rangle \\ z_s &= z_{OoI} - d_l \tan(\lambda_v + \zeta_{cam}). \end{aligned} \tag{5.4}$$

The splines defined by equations (5.3) and (5.4) are graphically illustrated in Figure 5.1.

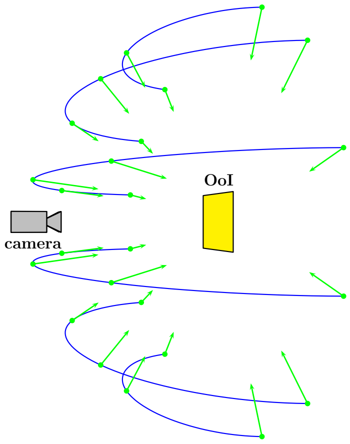


Figure 5.1: The example of the generated set of RTI goals marked with green dots and arrows. The yellow object identifies the scanned object of interest, and the blue curves indicates the horizontal splines, that represent possible positions of RTI goals.

As the next step, the sampling distance s_d is computed based on the equation

$$s_d = \frac{d_l(\lambda_{v,max} - \lambda_{v,min})}{v_s}, \tag{5.5}$$

and so it is equal to the shortest distance between two neighboring splines traveled on the surface of the previously defined spherical cap. With the usage of s_d we can compute the number of lighting positions on previously defined splines as

$$n_s(\lambda_v) = \left\lceil \max\left(\frac{d_l \cos(\lambda_v)(\lambda_{h,max} - \lambda_{h,min})}{s_d}, 1 \right) \right\rceil . \tag{5.6}$$

Then we can simply compute the equally distributed set of sample positions on particular splines from equation (5.4) by substituting for the angles λ_h from the set of angles given by equation

$$\Lambda_h(\lambda_v) = \begin{cases} \left\{ \lambda_{h,min} + \frac{\lambda_{h,max} - \lambda_{h,min}}{2} \right\} & \text{iff} \quad n_s(\lambda_v) = 1, \\ \left\{ \lambda_{h,min} + k\frac{\lambda_{h,max} - \lambda_{h,min}}{n_s - 1} \mid k \in \{0, 1, \ldots, n_s - 1\} \right\} & \text{iff} \quad n_s(\lambda_v) \geq 2. \end{cases} \tag{5.7}$$

Based on the previous considerations, the set of desired lighting positions Λ is defined by the equation

$$\Lambda = \{\Lambda_h(\lambda_v) | \lambda_v \in \Lambda_v\}. \tag{5.8}$$

This set is then reduced and altered by considering the constraints given by the deployment of UAVs. Namely the points that are within the sphere with center at the position of UAV carrying the camera and radius equal to critical avoidance radius $r_{c,a}$ are moved to the surface of this sphere for the preservation of the desired lighting angle.

Furthermore, the whole sets $\Lambda_h(\lambda_v)$ which violates the conditions on minimum or maximum altitude are removed from the set Λ. For resolving other unfeasible positions caused by collisions with obstacles or the occlusion, we propose a method for safe trajectory following which will be described in section 5.2.3. This method tries to get close to the desired positions even if they are unreachable. Such approach is advantageous in comparison to removing all unfeasible points since it attempts to place the light near to its desired position, which preserves the number of lighting positions and helps to avoid situations of absence of "lighting data" from some regions.

Let us note that the equations (5.3), (5.4) and (5.6) has to be used under condition

$$\lambda_{h,min}, \lambda_{h,max}, \lambda_{v,min}, \lambda_{v,max} \in \left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle . \tag{5.9}$$

Nevertheless, this condition is not so restrictive, since it corresponds to the fact that the RTI method does not require lighting which violates any of these angle limitations.

For further usage, the set of RTI positions has to be extended by the starting position of UAV carrying the light (P_{init}) since the RTI scanning procedure is supposed to be part of the scanning mission and the formation should return to its former state after finishing the RTI phase. Therefore, we define the complete set of RTI positions as

$$\Lambda_c = \Lambda \cup \{P_{init}\}. \tag{5.10}$$

5.2.2 Determination of the best sequence

Given the set of desired RTI light positions, we want to find the optimal closed path connecting all the positions in the set Λ_c with respect to certain criterion (e.g. minimum energy, shortest path, minimum time etc.). This problem is usually solved by splitting it into two subproblems - find paths between all pairs of positions from set Λ_c and find the optimal sequence of these paths with respect to certain criterion. The final path is then constructed by connecting previously found paths between pairs of positions in order given by previously found optimal sequence. The problem of this approach is that it can produce the path which does not conform constraints given by kinematic model of real robot. Since the RTI method requires the static scene, the UAV carrying the light should be also static during the process of taking photo of OoI. Therefore, the UAV should stop at every position from the Λ_c. Due to this fact and due to the deployment of holonomic kinematic model described in Section 3.2, presented problem can be rewritten into above mentioned subproblems without any additional restrictions and changes of the resulting optimal solution.

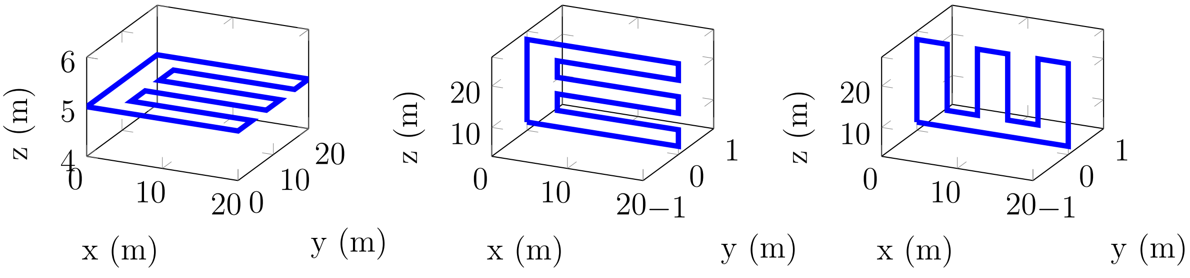
Based on the expected application of the system, we proposed to use the minimum energy as the optimization criterion for solution of the alternative of Travelling Salesman Problem (TSP), which also leads to maximization of possible flight time. The energy required by the multi-rotor UAV for the flying from position P_1 to P_2 can be computed as

$$\text{cost}(P_1, P_2) = \int_{t_{P_1}}^{t_{P_2}} u(\tau) i(\tau) d\tau, \tag{5.11}$$

where t_{P_1} and t_{P_2} are times of start of the flight from position P_1 to position P_2 and time of finishing this flight respectively, and $u(\tau)$ and $i(\tau)$ are a voltage and current measured on the battery output at time τ. To enable the simplification of the computation of energy cost of particular paths, we have conducted the simulations in realistic simulator Gazebo to identify the dependence of the path energy cost on the direction of a flight. During these simulations, a single multi-rotor UAV has flown along the closed trajectories of the E-shape with three different orientations in the space (see Figure 5.2) with velocity $1\,\mathrm{m\,s^{-1}}$. Based on these experiments, which show that the difference in energy consumption during this trajectories is less than $0,1\%$, we can claim that the energy required for flying closed trajectories at low velocity does not significantly depend on the direction of flight. This corresponds with the fact that the most of the energy consumed by slowly flying multi-rotor UAVs is spent on the hovering in the air. Assuming that the UAV will fly with constant velocity during the RTI scanning and that direct paths between particular positions are collision free, we can replace the optimization criterion defined in equation (5.11) by simple computation of Euclidean distance

$$\text{cost}(P_1, P_2) = \sqrt{(p_{1,x} - p_{2,x})^2 + (p_{1,y} - p_{2,y})^2 + (p_{1,z} - p_{2,z})^2}, \tag{5.12}$$

where $p_{i,c}$, $i \in \{1, 2\}$, $c \in \{x, y, z\}$ are particular coordinates of positions P_1 and P_2.



(a) planar (b) horizontal (c) vertical

Figure 5.2: The E-shape trajectory presented with three different orientations used for the experimental determination of dependence of the consumed energy on the direction of flight.

By combining aforementioned observations, we have completely defined our problem of finding optimal sequence of lighting positions as the Euclidean Traveling Salesman Problem, which is a specific type of TSP. The variations of TSP are intensively studied by science community for tens of years, which results in the availability of numerous approaches and solvers for TSPs including the dynamic programming [47], approximation algorithms [48], heuristic algorithms [49] and modern exact solvers [50]. These solvers differ in the quality of solutions as well as in memory and computational time requirements with respect to the size of the problem. Our requirements on the expected number of RTI positions (not exceeding 100) and maximum computational time equal to units of seconds are achievable for most of the approximate solvers and even for some exact solvers like Concorde [50]. However, optimal or nearly optimal solutions provided by any of these solvers are usually hardly predictable for the human being. This fact is the significant disadvantage in case of our application, since the UAVs are supposed to be monitored by human operators who should take over the control of the UAV in case of the occurrence of autonomously unsolvable failures.

The predictable behavior enables early recognition of a failure and thus helps to accelerate reaction to its occurrence by human operator. Due to high requirements on the safety of the system, we have decided to prioritize the predictability of the solution instead of finding the shortest path. Nevertheless, the length of the path affects the flight time, so we are naturally still interested in finding the way how to find a short path while preserving its predictability. For these purpose, we have proposed the following approach for finding the sequence of RTI positions S_p.

As the first step, the current position of the UAV carrying the light is added at the beginning of the sequence list S_p. Then we have to find the pair of the RTI positions (P_{start}, P_{end}) on the vertical boundaries for which holds

$$(P_{start}, P_{end}) = \arg \min_{(P_{i,j}, P_{k,l})} \text{dist}(P_{i,j}, P_{init}) + \text{dist}(P_{k,l}, P_{init}), \tag{5.13}$$

$$\begin{aligned} s.t.\ i =& k+1, \\ (j,l) \in& \{(1,1), (|\lambda_{h,i}|, |\lambda_{h,k}|)\}, \end{aligned} \tag{5.14}$$

where $P_{i,j}$ stands for the RTI position in the i-th row and j-th column, function dist($\cdot$) returns the Euclidean distance between two positions given as arguments, and $\lambda_{h,i}$ stands for the set of RTI positions in the i-th row. The position P_{start} is then added to the sequence of positions S_p. After that all positions on the vertical boundary on the way up to the highest row are added to S_p (for graphical illustration see Figures 5.3a, 5.3b, 5.3c). By these three steps we have achieved one of the corner positions in the highest row. Further, the procedure depends on the number of rows and is described separately.

In the case of even number of horizontal rows, the RTI positions are added line by line with switching the left-right and right-left direction and omitting the points which lie on the same vertical boundary as P_{start} (see Figure 5.3d). After reaching last admissible position in the most bottom row, the remaining points are added from the bottom row up to the P_{end} into S_p (see Figure 5.3e). Finally, the P_{init} has to be added at the end of the P_s to ensure the return to the initial position.

The solution for odd number of horizontal lines is more complicated, although it is based on the solution for even number of rows. Firstly, it is needed to determine which pair of the horizontal lines will be traversed in different manner. This procedure can be described as

$$(h_{odd,1}, h_{odd,2}) = \arg \min_{i,j} |\lambda_{h,i}| + |\lambda_{h,j}|, i, j \in \{1, 2, \ldots, |\Lambda_v|\},\ s.t.\ j = i + 1, \tag{5.15}$$

where $(h_{odd,1}, h_{odd,2})$ is the pair of indices of searched horizontal lines. The equation (5.15) can be interpreted as determining the pair of consequent horizontal lines with the smallest sum of the number of RTI positions. From the way of generation of RTI positions, namely from equation (5.6), it is clear that this pair will be either in the most upper or bottom horizontal row and the corresponding nearest horizontal row. The solution for odd number of rows is the same as in the case of even number of rows until the procedure reaches the pair $(h_{odd,1}, h_{odd,2})$. The particular RTI positions within this pair of rows are traversed in an up-side-down-side manner in such way that we are ending in the opposite side in lower row (see Figure 5.4c). By applying this approach, the last added RTI position lies one row lower at the opposite side of the starting position in the row with index $h_{odd,1}$. Thus, from there, the situation and also the solution is the same as in the case of even number of rows. The alternative sequence is generated by following the positions in row $h_{odd,1}$ to the opposite side, then flying back to the starting side of row $h_{odd,2}$ and again following this row to the opposite side of this row (see Figure 5.5b). In case that this solution for traversing first two rows is shorter than the corresponding original presented sequence, the relevant part

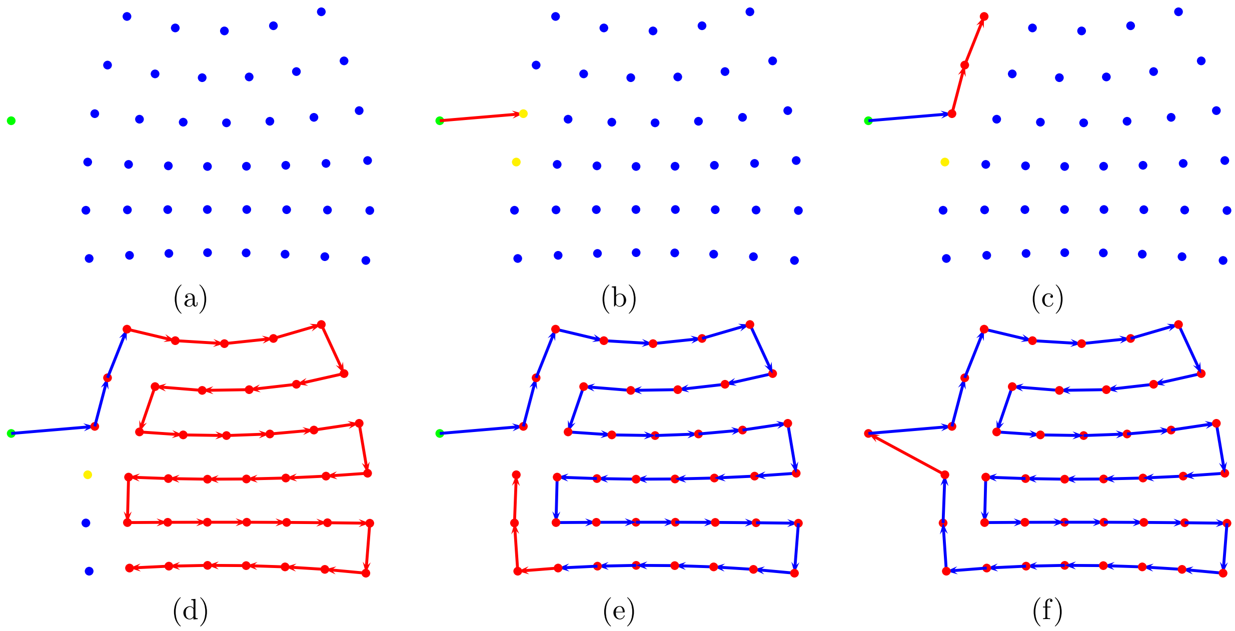


Figure 5.3: Illustration of the procedure of determining the predictable sequence of RTI positions for even number of horizontal rows. The green dot marks the initial state P_{init}, the pair of start and end node within RTI positions (P_{start}, P_{end}) is marked by yellow dots, the blue dots stands for the unvisited RTI positions while the red dots for already visited RTI positions. The arrows show the transitions between particular RTI positions, where the red arrows stand for the transitions added during the last step.

of the sequence is replaced by this z-sequence. The comparison of the solution obtained by applying this approach and the original solution is shown in Figure 5.5.

Although the above described algorithm generates completely predictable sequences of RTI positions, it does not explicitly try to find the sequence resulting in the shortest path. To get some quantitative comparison, we have tested our proposed algorithm against the TSP solver using Lin-Kernighan heuristic (LKH solver), which belongs to most efficient approximate algorithms for TSP. We have generated the testing case of 10000 sets of RTI positions, each with randomly chosen parameters $\lambda_{v,min}, \lambda_{v,max}, \lambda_{h,min}, \lambda_{h,max}, d_l, v_s$, distance between OoI and leader with camera d_{cam}, and initial position of the UAV carrying the light P_{init}. The ranges of particular parameters which were used within RTI test case generation are shown in Table 5.1.

As the aspects of the quality of solutions, we compare time requirements and length of the resulting solution. Regarding the CPU time is our implementation significantly faster. For example for problem with 50 RTI positions, it is approximately 15 times faster than the LKH solver. However, since the total CPU time needed by LKH solver to solve this problem is 5.1ms (computed on the computer with CPU intel CORE i7 8250), this aspect is not so important for our application. More significant parameter from point of view of our application is the difference in length of the paths produced by our algorithm and by

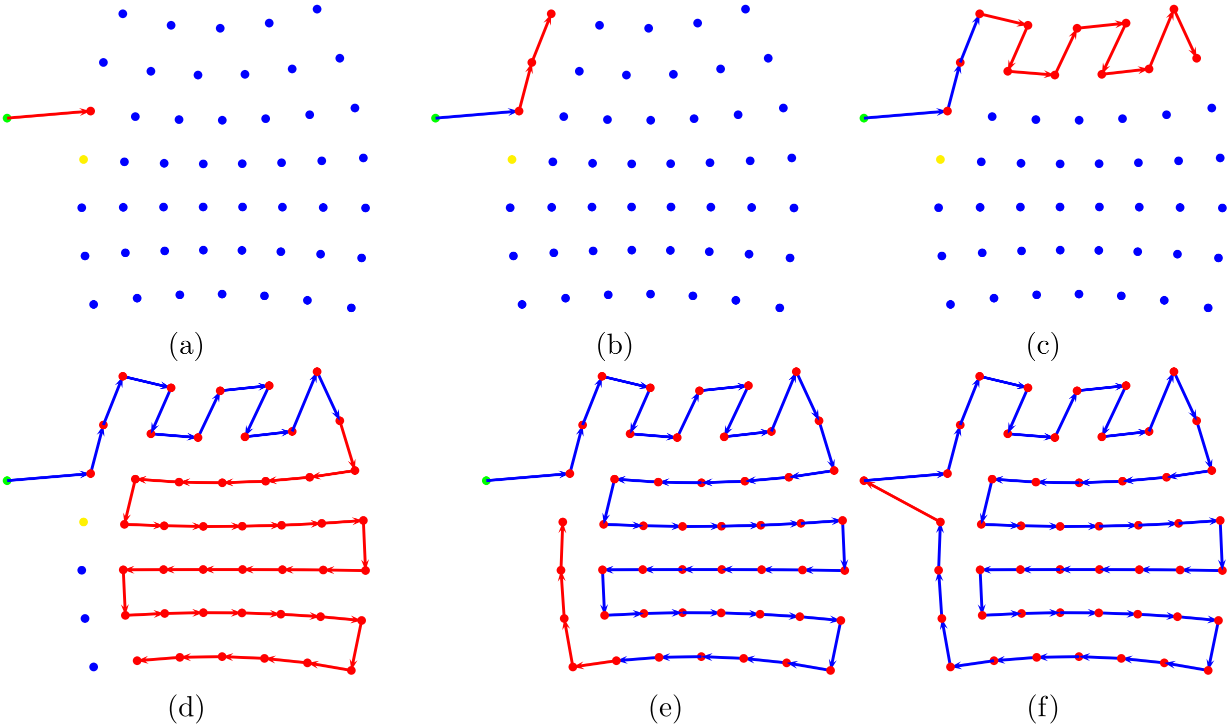
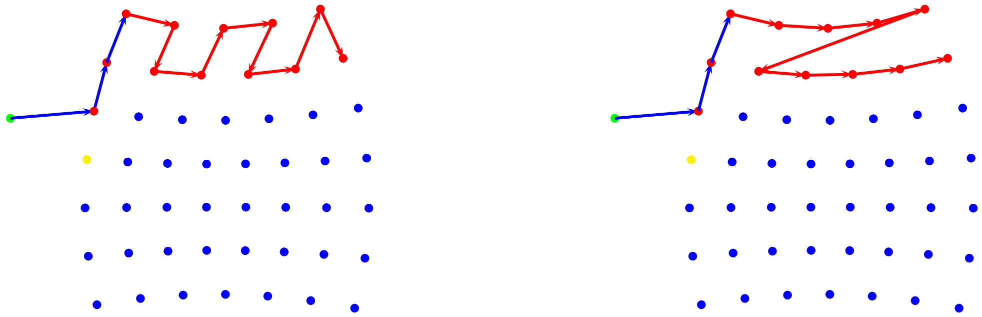


Figure 5.4: Illustration of the procedure of determining the predictable sequence of RTI positions for odd number of horizontal rows. The green dot marks the initial state P_{init}, the pair of start and end node within RTI positions (P_{start}, P_{end}) is marked by yellow dots, the blue dots stands for the unvisited RTI positions while red dots for already visited RTI positions. The arrows show the transitions between particular RTI positions, where the red arrows stand for the transitions added during the last step.

Parameter	minimum value	maximum value
v_s	2	10
$\lambda_{v,min}$	$-\pi/2\,\mathrm{rad}$	$-\pi/6\,\mathrm{rad}$
$\lambda_{v,max}$	$\pi/6\,\mathrm{rad}$	$\pi/2\,\mathrm{rad}$
$\lambda_{h,min}$	$-\pi/2\,\mathrm{rad}$	$-\pi/6\,\mathrm{rad}$
$\lambda_{h,max}$	$\pi/6\,\mathrm{rad}$	$\pi/2\,\mathrm{rad}$
d_{cam}	$3\,\mathrm{m}$	$15\,\mathrm{m}$
d_l	$3\,\mathrm{m}$	$15\,\mathrm{m}$

Table 5.1: Ranges of particular parameters used within generation of testing set for comparison of our predictable solution of TSP with solution provided by the LKH solver.

LKH solver. Considering the length of path as the comparison value, our solution is better or equals to LKH solution in 9% of test samples and is not longer by more than 50% in 98% of test samples. The ratio of the worst case scenario among all test samples and its corresponding solution provided by LKH solver is 1.83. More detailed results of this



(a)

(b)

Figure 5.5: Comparison of two different solutions of the RTI sequence determination problem. The green dot marks the initial state P_{init}, the pair of start and end node within RTI positions (P_{start}, P_{end}) is marked by yellow dots, the blue dots stands for the unvisited RTI positions while the red dots for already visited RTI positions. The arrows show the transitions between particular RTI positions, where the red arrows stand for the transitions added during the last step.

comparison are shown in Figure 5.6. Based on this statistical comparison, we can claim that our predictable solution is sufficiently good for our application and the predictability is worth the additional length of the path.

5.2.3 Trajectory generation and tracking

Once the sequence of RTI positions is determined, the UAV carrying the light needs to be safely navigated through the environment to sequentially visit all desired locations. To achieve this goal, we took the advantage of our system for trajectory tracking originally proposed in [1] and its enhancement described in Section 3.4.

The desired trajectory for the RTI phase Tr_{RTI} is generated by simple sampling of the direct straight path between consequent RTI positions with the sampling distance d_{RTI}, without considering any obstacles. The desired velocity is influenced by parameter d_{RTI} and can be expressed as

$$v_{RTI} = \frac{d_{RTI}}{T_s}, \tag{5.16}$$

where T_s is the sampling period of the trajectory. Nevertheless, this desired trajectory is usually unfeasible due to the presence of obstacles and does not conform all necessary conditions to achieve good lighting. Therefore, similarly to approach described in Section 3.4, we define the trajectory tracking during the RTI phase as an optimization task with a vector of decision variables $\mathcal{U}_j$ over the time horizon of length NT_s.

The objective function J_{RTI} is similar to $J_{j,p}$ defined in equation (3.17), only the

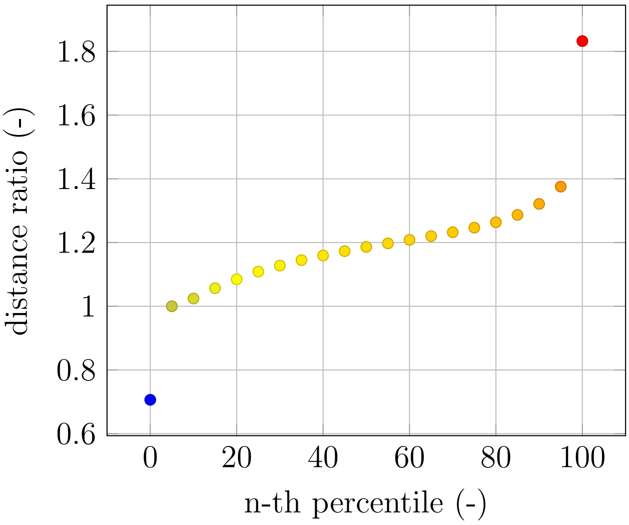


Figure 5.6: Comparison of length of TSP solution obtained from the LKH solver and our predictable solution

distance $d_{j,FoV}(\cdot)$ used in component $J_{j,occlusion}$ is computed according to set of equations

$$\begin{aligned}
d_{xy}(k) &= \sqrt{(x_L(k) - x_j(k))^2 + (y_L(k) - y_j(k))^2},\\
\alpha_{diff,h}(k) &= \left|\text{atan2}(y_j(k) - y_L(k), x_j(k) - x_L(k)) - \varphi_L(k)\right|,\\
\alpha_{diff,v}(k) &= \left|\text{atan2}(z_j(k) - z_L(k), d_{xy}(k)) - \xi_L(k)\right|,\\
\beta_{diff,h}(k) &= \min\{\alpha_{diff,h}(k), \pi - \alpha_{diff,h}(k)\},\\
\beta_{diff,v}(k) &= \min\{\alpha_{diff,v}(k), \pi - \alpha_{diff,v}(k)\},\\
d_{FoV,xy}(k) &= d_{xy}(k)\sin\left(\beta_{diff,h}(k) - \frac{AoV_h}{2}\right),\\
d_{FoV,z}(k) &= \sqrt{d_{xy}(k)^2 + (z_L(k) - z_j(k))^2}\sin\left(\beta_{diff,v}(k) - \frac{AoV_v}{2}\right),\\
d_{j,FoV}(k) &= \sqrt{d_{FoV,z}(k)^2 + d_{FoV,xy}(k)^2} - r_d,
\end{aligned} \tag{5.17}$$

where $d_{FoV,xy}(\cdot)$ is the distance to the nearest vertical border of FoV, $d_{FoV,z}(\cdot)$ is the distance to the nearest horizontal border of FoV, $\alpha_{diff,h}(\cdot)$ and $\alpha_{diff,v}(\cdot)$ stands for the angle between the nearest vertical respectively horizontal border of the FoV and connecting line between leader and j-th follower, and r_d marks the radius of the j-th robot. $\beta_{diff,h}(k)$ and $\beta_{diff,v}(k)$ are equivalent to $\alpha_{diff,h}(k)$ and $\alpha_{diff,v}(k)$ angles, but beside the FoV of the camera it considers also the FoV of the virtual camera pointed in the exactly opposite direction than the real camera. With this alteration, the $J_{RTI}(\cdot)$ penalizes not only the occlusion caused by the UAV carrying the light but also the shadows visible in the FoV caused by lighting from behind the leader, which is important for obtaining proper images for later RTI image processing. The graph of an example of the dependence of values of $J_{j,occlusion}$

of the follower for the RTI phase on the x, y coordinates is shown in Figure 5.7. Since the corresponding constraint function $g_{j,occlusion}(\cdot)$ applies also the distance $d_{j,FoV}(\cdot)$, another changes of the $g_{j,occlusion}(\cdot)$ are not needed and it is applied in its original form presented in equation (3.32). The solution of the optimization task is obtained in the similar manner as in the case of the optimization task for trajectory tracking described in Chapter 3.

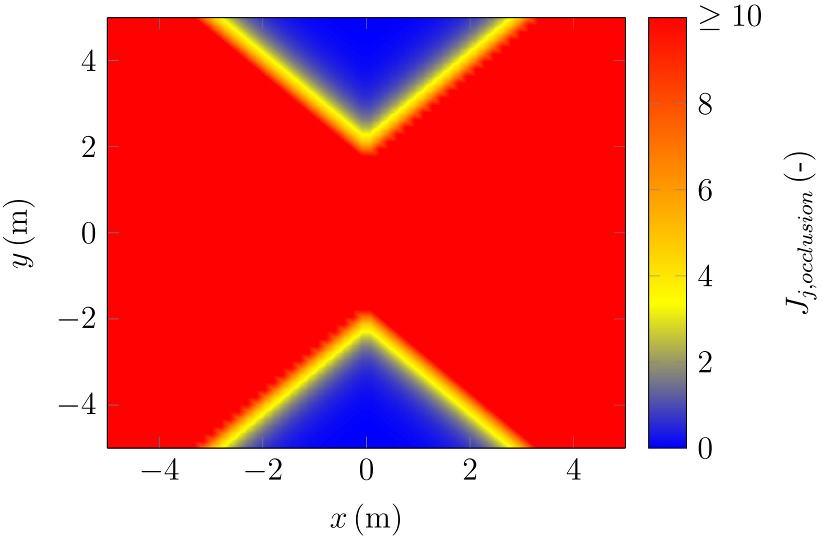


Figure 5.7: Graph of the objective function for penalization of the occlusion during the RTI scanning phase. The position of the camera for which the graph is generated is $P_{cam} = (0,\, 0)^T$.

To achieve the precise light conditions, the UAV carrying the light is supposed to hover at the desired position during taking the photo. This requirement is introduced into our system by addition of the certain number of particular RTI positions as the transition points of Tr_{RTI} after each fly-over to the next RTI position. The number of this repetitions N_H depends on the required time for stabilization at the desired RTI positions and it can be computed as

$$N_H = \left\lceil \frac{T_{stab}}{T_s} \right\rceil, \tag{5.18}$$

where T_{stab} stands for the required time for stabilization.

The whole RTI scanning procedure is started by the leader, which provides the information about requested RTI scanning to the mission controller that initializes the RTI scanning phase for the necessary UAVs. After that, the UAV carrying the light starts flying the desired trajectory Tr_{RTI}. It starts in the initial position and flies to the first RTI position, where it stabilizes itself. After expiration of time T_{stab}, the photo is taken by the UAV equipped with camera and the UAV with light flies to the next RTI position. This sequence of actions is repeated until the last RTI position is reached. Then the UAV with light flies back to its initial position P_{init} and the RTI scanning phase is finished.

5.3 Image post-processing

In the previous sections, we have described the procedure of obtaining the set of images of a certain object together with the information about the direction of lighting, with the usage of two UAVs, one equipped with a camera and the second one equipped with a light. This process is motivated by the RTI method, which has the requirements on the static scene and the constant intensity of lighting. Nevertheless, both these conditions can be hardly achieved by mounting the camera and lights on the UAVs hovering in the air.

The precision of placement of camera and lights depends on the precision of localization in the environment and the quality of the positional controller. Even in ideal conditions the positional error will be still nonzero and is naturally increasing with the imprecision of localization. The common ways of indoor localization usually have the uncertainty in the measurement in tens of centimeters, which equals to possible displacement of camera and lights. Based on our experiments and discussion with experts in the field of restoration and historical science, we will assume that the displacement of light, which results in the imprecision of lighting angle and the desired intensity of light, does not degrade the resulting image. Moreover, the only known way how to solve the problem of displacement of light is to achieve lower localization error.

While the displacement of light can be tolerated within a certain range, the displacement of the camera during the process of taking images causes complete degradation of resulting PTM representation of the image even for displacements equal to a few centimeters. The causes of such displacement during RTI phase are shown in Figure 5.8. However, in contrast to light displacement, the camera displacement can be solved by post-processing the set of images so that they will fit to each other and will look like they were taken by the static camera. This process is called the image registration, and various approaches to solving this problem were already developed [51, 52, 53].

The specificity of this application is that particular images are taken by substantially different lighting conditions and in extreme case, two images do not have to contain any common features. Hence, several publicly available implementations of the image registration algorithms failed in the registration process of images taken during the RTI scanning phase. Since the solution of this task is not part of this thesis, to enable the presentation of results of the RTI scanning, the problem is solved by simulation of a static camera.

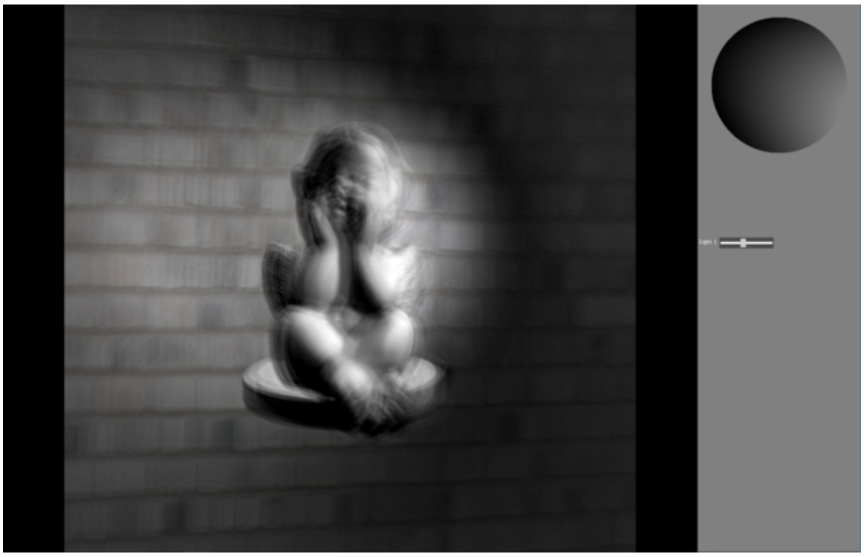


Figure 5.8: Example of the PTM representation of the image obtained from the onboard camera without any post-processing.

Chapter 6

Experimental results

To verify the functionalities, desired properties, and correctness of the system described in the previous sections, we have performed numerous tests within the realistic simulator Gazebo under ROS and also one experiment in a real-world scenario. In the following sections, we present the results of two experiments performed in the simulator and the real-world experiment. The first presented experiment in the simulation is aimed at the verification of the complete system for formation control including the mission controller. The intention of the second presented experiment is to verify the proposed approach for the RTI method and its implementation.

6.1 Complex experiment

The experiment presented in this section aims at the verification of the proposed system in the challenging environment, which consists of a narrow gate, a long hall with pictures, and the room with an altar, several statues, and pictures (Figure 6.1). This environment and the initial scanning plan was chosen to present the complex example of expected deployment of the system. The illustration of one complete mission also enables the demonstration of the functionality of the mission controller. Since the operation of the mission controller is described in Chapter 4 in details, the following description of the experiment mentions only its interventions that deviates from the normal operation. Nevertheless, the whole mission is accomplished fully autonomously without any involvement of human operators.

The specified scanning plan contains the RTI scanning procedure and also passages that requires two light sources. Hence the employed formation consists of three UAVs - leader with the camera and two followers carrying the lights. The leader has assigned the index 1, the follower flying on his left side has the index 2 and the follower flying on the right side is allotted the index 3. These indices correspond to the priorities of particular robots in the formation, which are used in part of the objective function $J_{j,trajectories}$, which

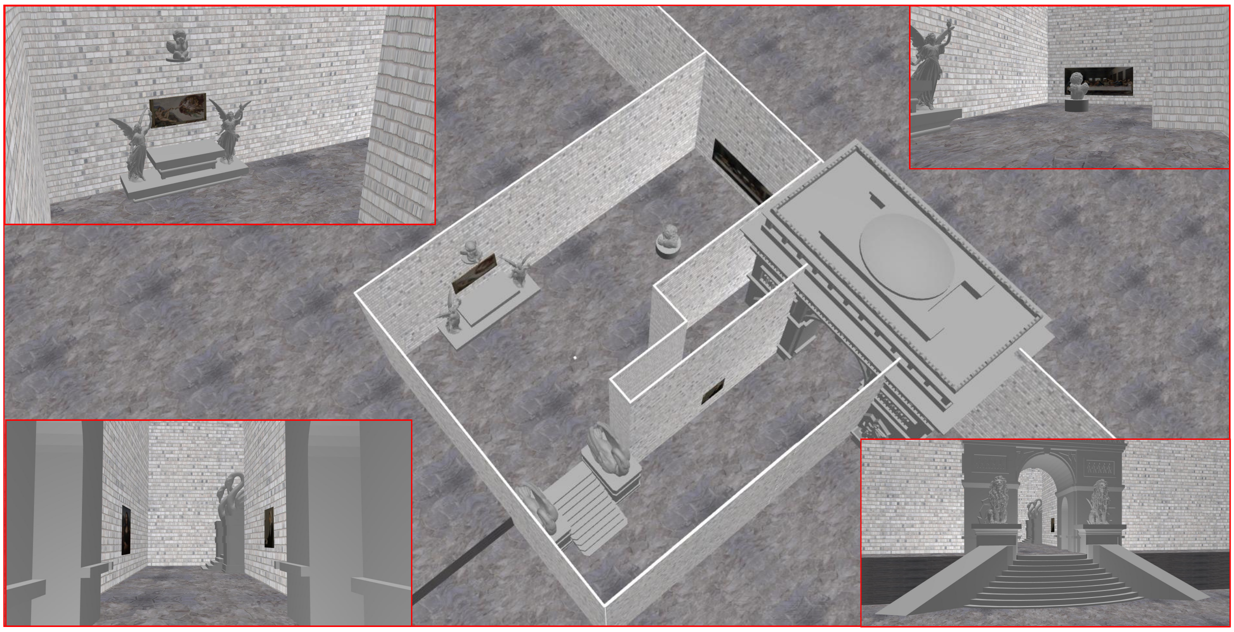


Figure 6.1: The simulation environment used in the experiment described in Section 6.1.

penalizes possible collisions with trajectories of other robots. Thus, the leader, as the robot with the lowest index and so highest priority, is not influenced by planned trajectories of other robots, the follower with index 2 avoids only planned trajectory of the leader and the follower with index 3 avoids trajectories of all robots in the formation.

The formation starts in front of the gate, on the level below the desired scanning area. After completion of the take off procedure and flight to initial position, it follows the trajectory leading through the first narrow passage, which is formed by the gate. Since the formation is not scanning any OoI in this part of the trajectory, the strict formation leader-follower scheme is applied for computation of desired trajectories of followers. The values of parameters used within this scheme are presented in Table 6.1 together with other constants used for the definition of the mission and setup of the formation control system.

After flying through the gate, the mission plan includes the scan of the picture on the right side of the hall and within a few seconds the scan of the picture on the opposite side of the hall. This part of the mission plan requires the fast change of the desired orientation and thus enables to show the advantage of the leader-follower scheme with virtual OoI. Thanks to this scheme, the whole formation rotates in the compact shape while providing the desired lighting in the predefined distance from the camera. The behavior of the formation during this phase of the experiment is graphically illustrated in Figure 6.2.

In the next part of the mission plan, the leader-follower scheme is set to the strict formation and the formation is lead through the very narrow passage with width 5.3 m formed by stairs surrounded by statues. According to parameters, which define the desired

Constant	Value	Brief description	Constant	Value	Brief description
general			objective function coefficients		
t_s	$0.2\,\mathrm{s}$	sampling time	α	$1\,\mathrm{m}^{-2}$	position deviations
N	15	prediction horizon	β	$0.1\,\frac{\mathrm{s}^2}{\mathrm{m}^2}$	positional controls
n	2	control horizon	γ	0.01	obstacles collisions
$r_{d,l}$	$1.5\,\mathrm{m}$	detection radius	δ	0.01	occlusion
$r_{d,f}$	$2.0\,\mathrm{m}$	avoidance radius	η	0.01	trajectories collision
$r_{a,l}$	$0.75\,\mathrm{m}$	detection radius	ζ	$1\,\mathrm{rad}^{-2}$	orientation deviations
$r_{a,f}$	$1.0\,\mathrm{m}$	avoidance radius	κ	$0.1\,\frac{\mathrm{s}^2}{\mathrm{rad}^2}$	orientation controls
kinematic constraints					
$\xi_{j,min}$	$-90°$	min. pitch angle	$\xi_{j,max}$	$90°$	max. pitch angle
$v_{x,min}$	$-1\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. x-velocity	$v_{x,max}$	$1\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. x-velocity
$v_{y,min}$	$-1\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. y-velocity	$v_{y,max}$	$1\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. y-velocity
$v_{z,min}$	$-1\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. z-velocity	$v_{z,max}$	$1\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. z-velocity
$\omega_{j,min}$	$-0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. yaw rate	$\omega_{j,max}$	$0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. yaw rate
$\varepsilon_{j,min}$	$-0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. pitch rate	$\varepsilon_{j,max}$	$0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. pitch rate
leader-follower scheme					
$d_{p,2}$	-2.0 m	dist. direct	$d_{p,3}$	-2.5	dist. direct
$d_{q,2}$	1.0 m	dist. sideways	$d_{q,3}$	-1.0	dist. sideways
$d_{r,2}$	0.0 m	dist. ascent	$d_{r,3}$	0.0 m	dist. ascent

Table 6.1: Overview of the values of particular constants used for the complex experiment presented in Section 6.1. The brief descriptions provided in the table are used to remind the meaning of particular symbols and they are correctly defined in the previous sections. The values of parameters with the j index are applied to all robots in the formation, while the values of parameters with the l or f index are applied only to leader or followers respectively.

shape of the strict formation and parameters that described the detection and avoidance radii, the width of the formation with consideration of radius of UAV and its avoidance radius is 5.5 m. With consideration of detection radius, the width of the formation is even 7.5 m. Thus, the flight through this passage without a change of the formation shape is unfeasible and also the solutions with small changes are significantly penalized due to the proximity of obstacles. Therefore the UAVs fly through the corridor in sequence, which is shown in Figure 6.3.

After the successful flight through this narrow corridor, the formation enters the large room in which it flies to the next OoI, the statue. The mission plan requires its scan under precisely achieved predefined lighting conditions, so when the formation achieves the position in the proximity of the OoI, it switches the leader-follower scheme to lighting setup given by equations (3.1) and (3.2). Afterwards scanning of this OoI, the formation is required to fly to the next OoI, and so the leader-follower scheme is again changed to the strict formation. During the flight between these OoIs, the program for formation control of follower with index three crashes. This fault is detected by the mission controller, and so it pauses the planning methods of all other UAVs in the formation and starts to solve the faulty follower. As the fault does not disable the autonomous control of the UAV and the mission setup requires the following of the mission also in the case of presence of faulty follower, the autonomous landing procedure of the follower is started. After its successful landing, it is removed from the list of formation members and the planning of remaining UAVs is reactivated.

The remaining two UAVs fly with the use of the leader-follower scheme with virtual OoI to the next OoI, which is the altar. After reaching the desired position for scanning of this OoI, the formation travels to the next OoI, which is the statue above the altar for which the mission plan requires to perform the RTI scanning. The analogy of this part of the mission is described in the next section in details. With the finishing of the RTI scanning procedure, the formation achieves all desired scanning goals of the mission, and they are required to fly back to the starting position. Since the formation consists only of two UAVs, the flight through the narrow corridors is easier. Nevertheless, after the turn back to the hall, the mission controller detects insufficient precision of localization of the follower with index 2 and consequently, the planning of the leader is paused. Since this fault disables the safe autonomous control of the UAV, its manual control is required. When the mission controller detects that the UAV is manually controlled, the fault is supposed to be resolved, and the planning of the leader is reactivated. Next, the leader follows the trajectory to the initial position, where it autonomously lands and the mission is completed.

The trajectories of all robots in the formation together with the corresponding orientations of the camera and lights are shown in Figure 6.4, Figure 6.5, and Figure 6.6. The course of the values of particular objective functions during the mission are presented in Figure 6.7 and the graph of applied control inputs to kinematic model of the leader are shown in Figure 6.8.

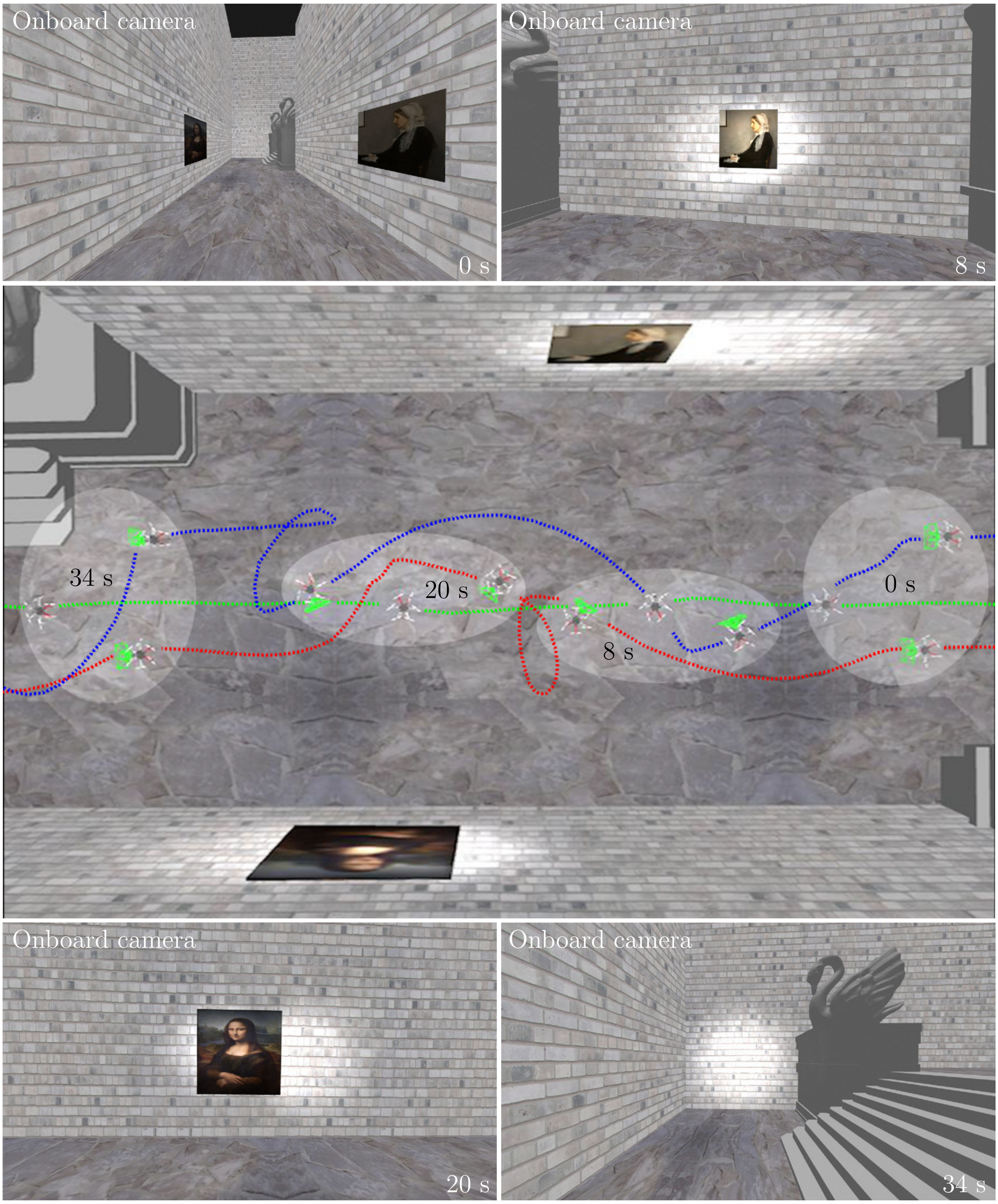


Figure 6.2: Trajectories of particular UAVs in the formation during the transition between two OoIs with the use of leader-follower scheme with virtual OoI. The upper and bottom pictures show the images taken by the onboard camera carried by the leader (green dotted line). The green pyramids in the pictures mark the light sources. The times written in the figure are relative to the simulation time of the first presented position of the formation.

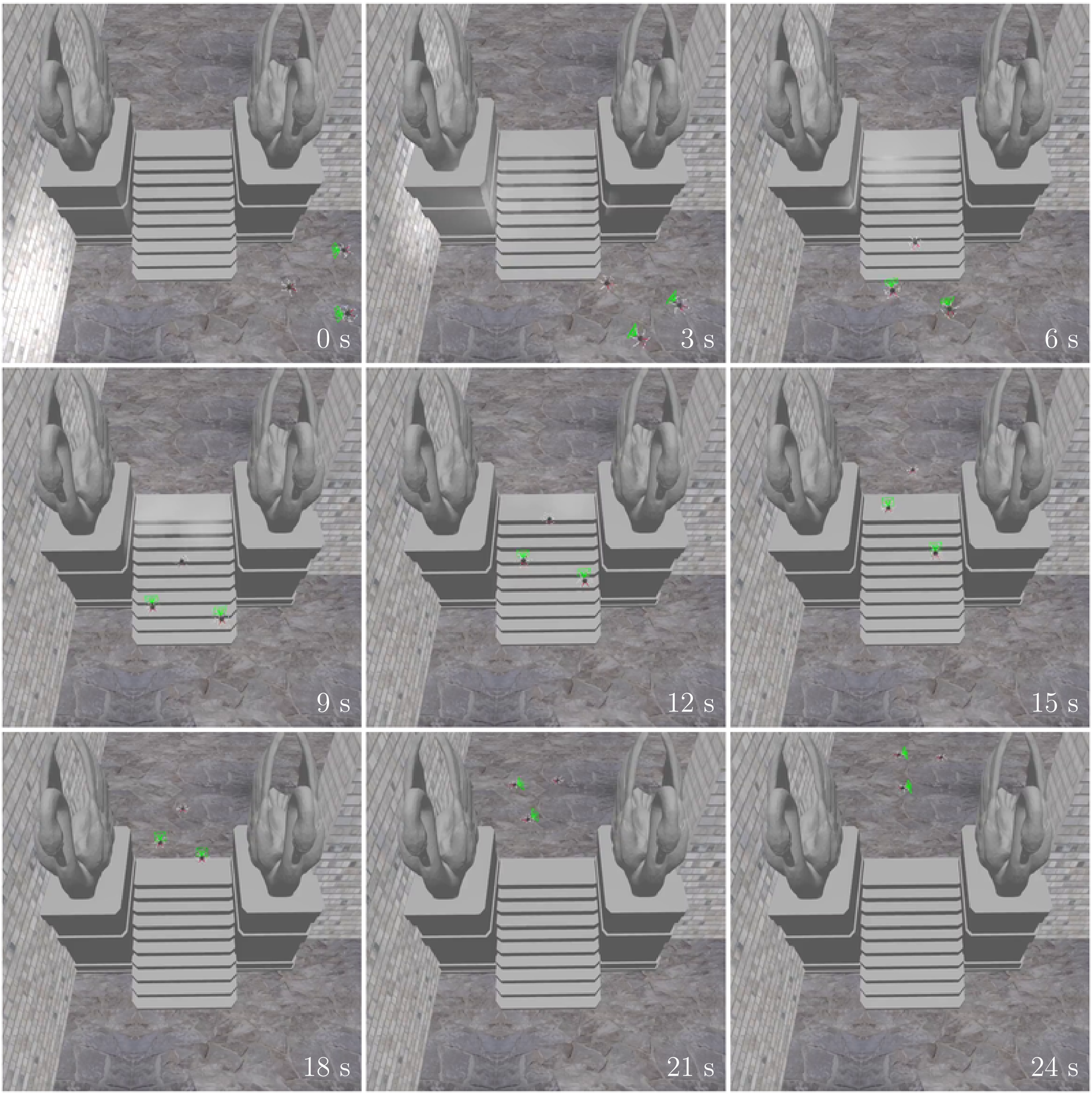
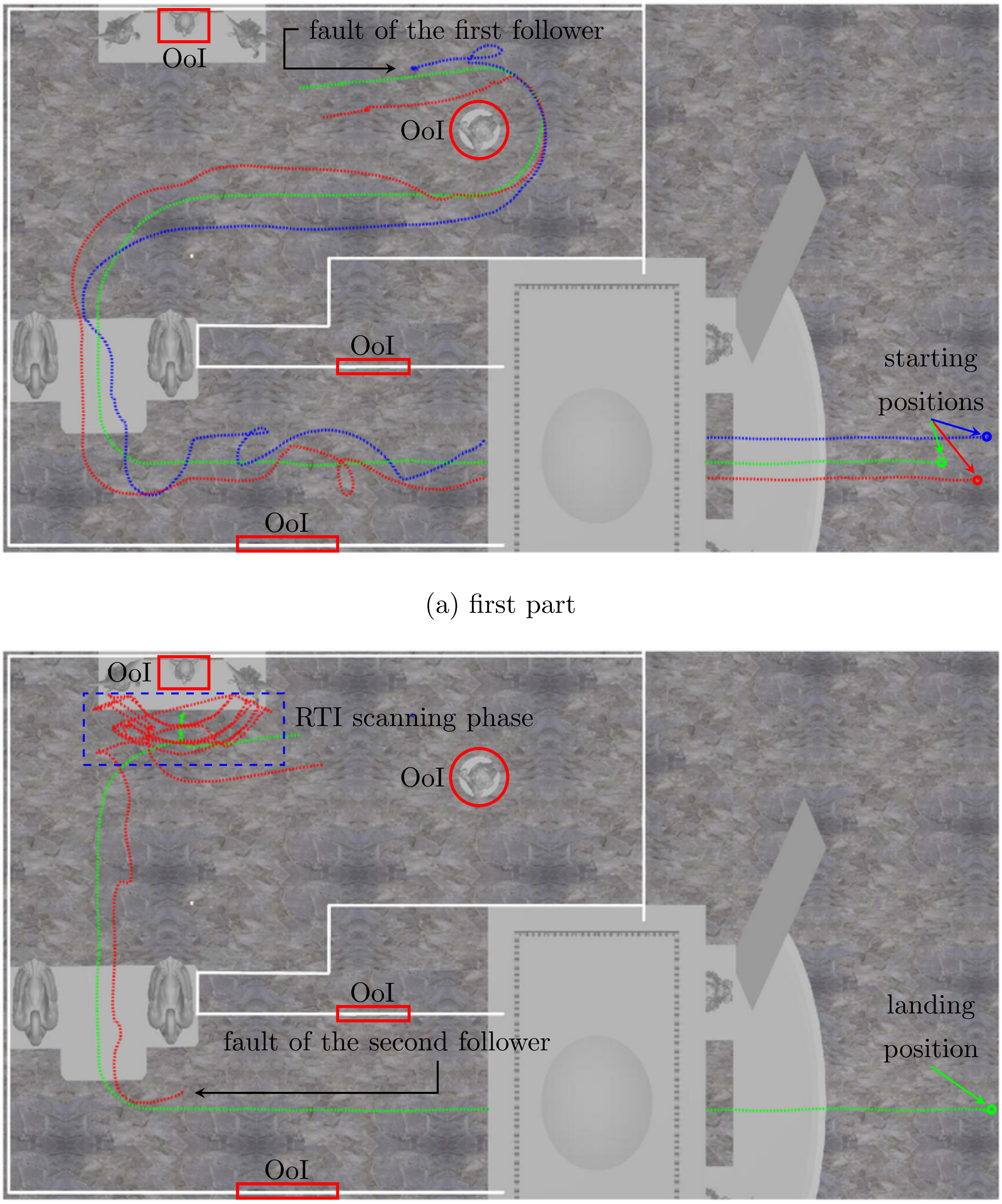


Figure 6.3: Snapshots from the simulation of the formation flying through the narrow corridor during the experiment presented in Section 6.1. The time in particular pictures is relative to the simulation time corresponding to the first picture in the sequence. The green pyramids in the pictures mark the light sources.



(a) first part

(b) second part

Figure 6.4: Trajectories of robots during the complex experiment presented in Section 6.1. The experiment shows the scanning of several OoIs, one RTI scanning procedure and includes also the faults of the followers. The green dots mark the trajectory of the leader, the red dots mark the trajectory of the follower with index 2 and the blue dots mark the trajectory of the follower with index 3. The video of the experiment is available at `http://mrs.felk.cvut.cz/kratky2019thesis`.

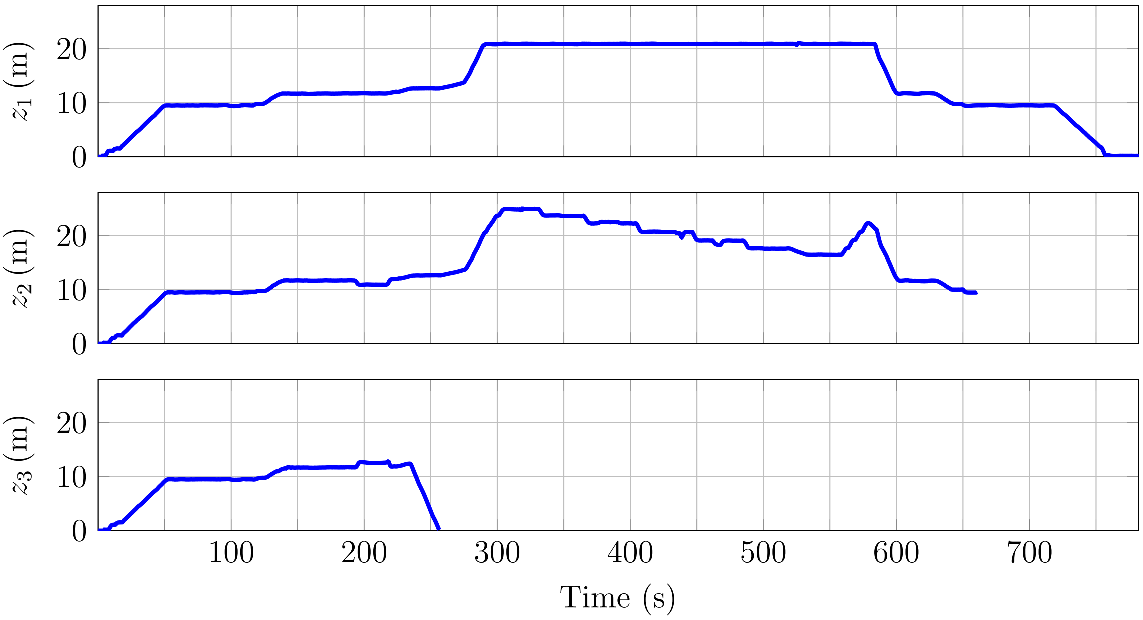


Figure 6.5: The z coordinate of the trajectories of robots during the complex experiment presented in Section 6.1. The earlier end of the graphs for z_2 and z_3 are caused by detected failures of the corresponding UAVs during the mission.

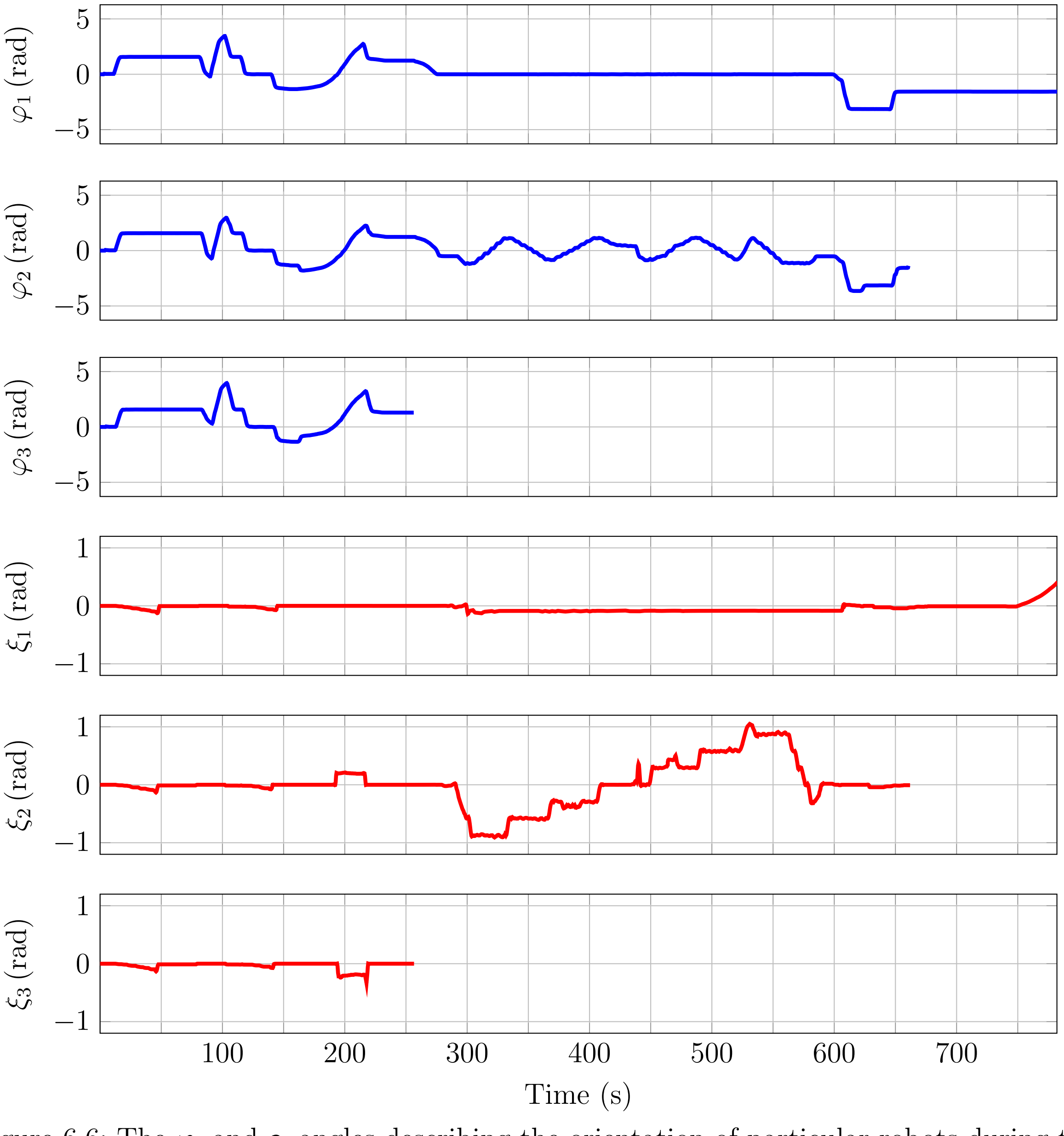


Figure 6.6: The φ_i and ε_i angles describing the orientation of particular robots during the complex experiment presented in Section 6.1. The earlier end of the graphs for φ_2, φ_3, ε_2 and ε_3 are caused by detected failures of the corresponding UAVs during the mission.

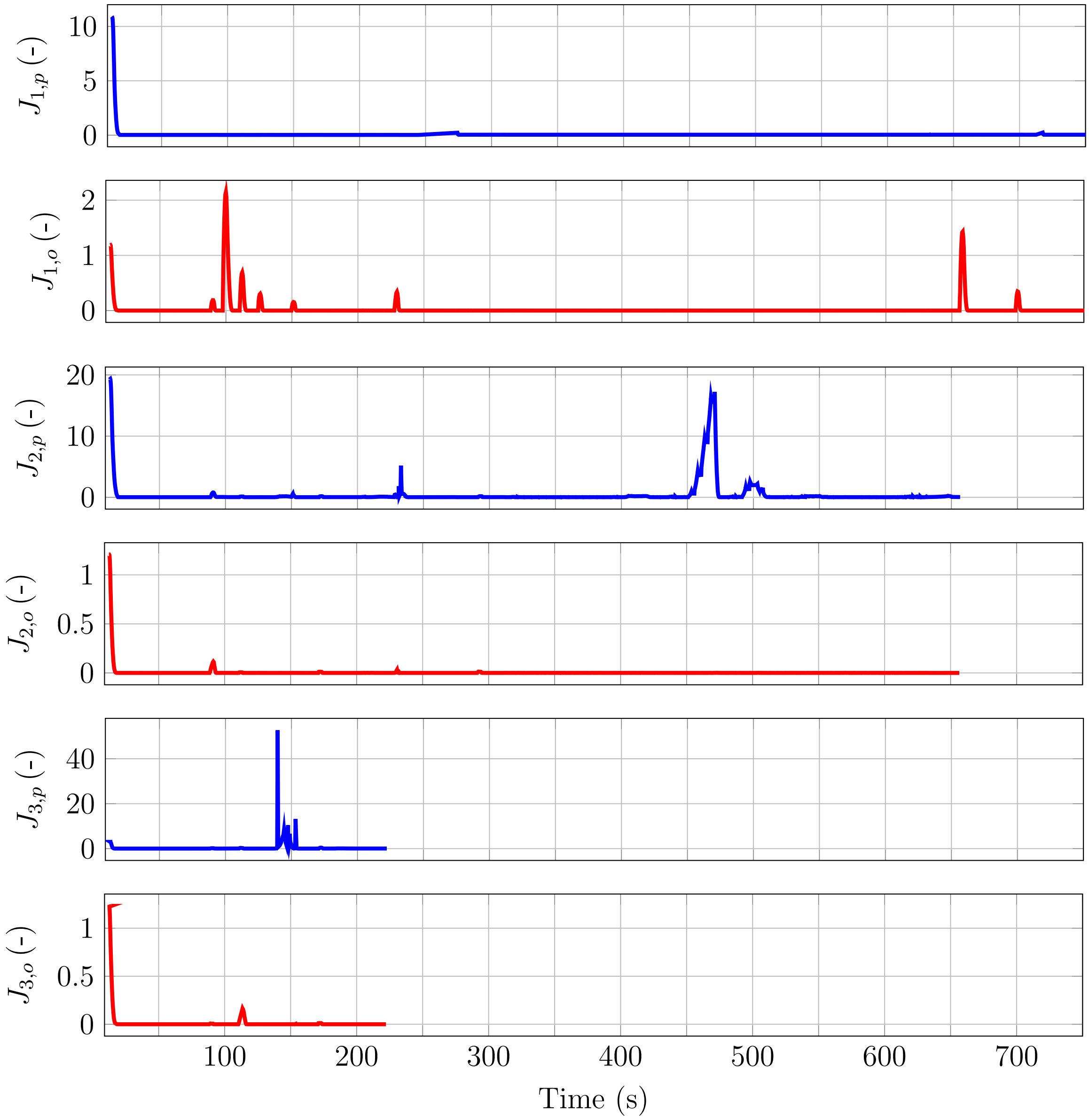


Figure 6.7: The values of the objective functions $J_{j,p}$ and $J_{j,o}$ of particular robots during the complex experiment presented in Section 6.1. The earlier end of the graphs for $J_{2,p}$, $J_{3,p}$, $J_{2,o}$ and $J_{3,o}$ are caused by detected failures of the corresponding UAVs during the mission.

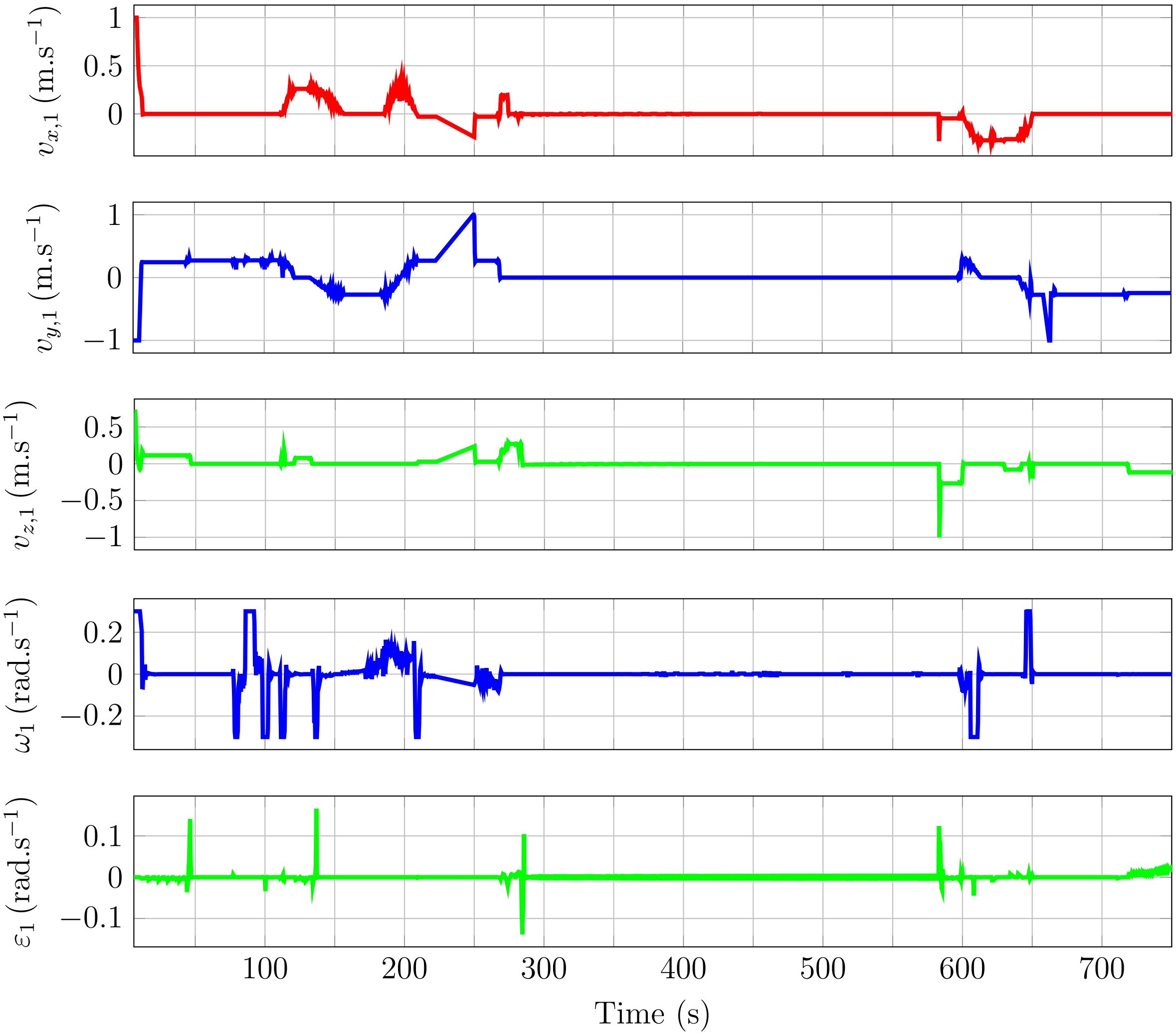


Figure 6.8: The values of control inputs applied to leader during the complex experiment presented in Section 6.1.

6.2 RTI experiment

The experiment for verification of the RTI method implementation was performed in the realistic robotic simulator Gazebo in the environment inspired by the interiors of churches. In this experiment, only two UAVs (one equipped with the camera and the second one equipped with light) are deployed. They are used for performing the RTI scanning procedure of the statue situated above the altar in height inaccessible for humans without the use of scaffolding or a mobile platform.

The experiment was performed with the following setup. The camera was set in the position which is 4 meters directly in front of the scanned object, the desired distance of the light source from the scanned object d_l was set to 5.5 m and the desired number of RTI samples in vertical direction $v_s = 7$, the desired sampling distance of the RTI trajectory was set to $d_{RTI} = 0.1\,\text{m}$ and the required stabilization time $T_{stab} = 3\,\text{s}$. The values of all parameters and coefficients are shown in Table 6.2 and Table 6.3. This setup leads to the generation of 56 RTI positions and the resulting trajectory of the overall length of 110.55 m. Together with the stops at each RTI position required for the stabilization, the estimated time of flight following the generated trajectory is 358.8 s. The set of generated points together with the trajectory flown by the follower carrying the light are shown in Figure 6.9.

In compliance with the theory presented in Chapter 5, the follower stops at each reachable RTI position and waits until the image is taken by the leader. In this way, the system collects 56 images of the scanned object under various lighting conditions during the presented experiment. All these images are shown in Figure 6.10. Based on these images and the file containing the information about corresponding lighting directions, the PTM representation of the image is computed with the use of program PTM Fitter. The problem, which was already mentioned in Section 5.3, is that the camera mounted on the UAV hovering in the air is not static with respect to the scanned object and thus the images are shifted, rotated, and differently scaled. These transformations between particular images cause a blur in the resulting PTMs. The difference between the PTM representation obtained from the unregistered images and the PTM representation obtained from the images taken by properly registered images is illustrated in Figure 6.11.

Since the solution of the image registration problem is beyond the goals of this thesis, we have simultaneously collected the images from the virtual static camera placed at the desired position of the onboard camera, to be able to present the results of implemented RTI method. The resulting PTM representation of the scanned object obtained from the images taken by the static camera is illustrated in Figure 6.12. In this figure, the scanned image with different lighting settings is shown within the screenshots from the program for the view of the PTMs. The main advantage of obtaining the PTM from the set of images is that the image can be displayed under arbitrarily lighting conditions. Since this result can be hardly presented within the printed work, it is demonstrated in the video available at `http://mrs.felk.cvut.cz/kratky2019thesis`.

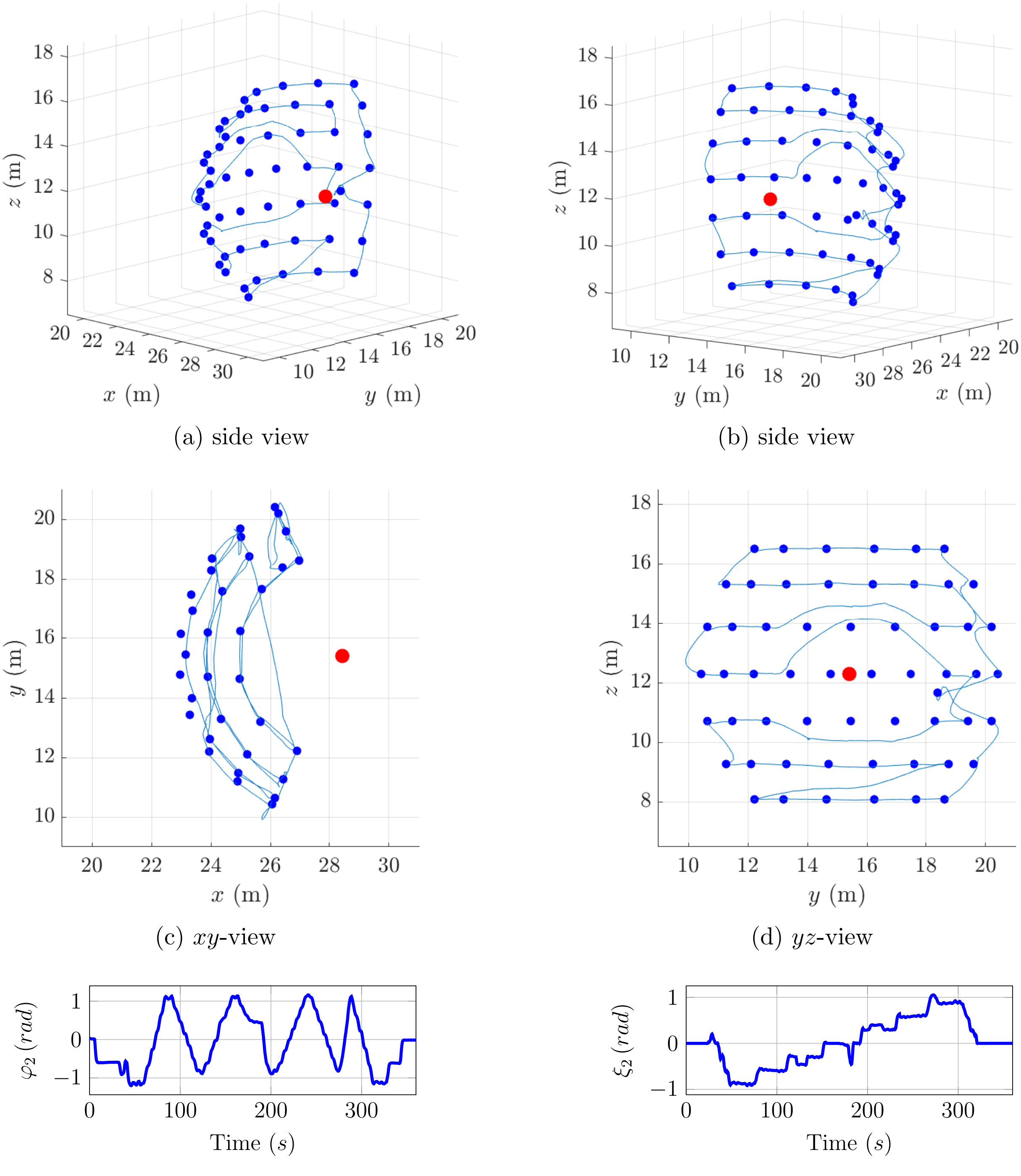
Constant	Value	Brief description	Constant	Value	Brief description
general			objective function coefficients		
T_s	$0.2\,\mathrm{s}$	sampling time	α	$1\,\mathrm{m}^{-2}$	position deviations
N	15	prediction horizon	β	$0.2\,\frac{\mathrm{s}^2}{\mathrm{m}^2}$	positional controls
n	2	control horizon	γ	0.3	obstacles collisions
$r_{d,1}$	$2.5\,\mathrm{m}$	detection radius	δ	0.05	occlusion
$r_{d,2}$	$3.0\,\mathrm{m}$	avoidance radius	η	0.6	trajectories collision
$r_{a,1}$	$1.5\,\mathrm{m}$	detection radius	ζ	$1\,\mathrm{rad}^{-2}$	orientation deviations
$r_{a,2}$	$2.0\,\mathrm{m}$	avoidance radius	κ	$0.4\,\frac{\mathrm{s}^2}{\mathrm{rad}^2}$	orientation controls
kinematic constraints					
$\xi_{j,min}$	$-90°$	min. pitch angle	$\xi_{j,max}$	$90°$	max. pitch angle
$v_{x,min}$	$-1\,\frac{\mathrm{m}}{\mathrm{s}}$	min. x-velocity	$v_{x,max}$	$1\,\frac{\mathrm{m}}{\mathrm{s}}$	max. x-velocity
$v_{y,min}$	$-1\,\frac{\mathrm{m}}{\mathrm{s}}$	min. y-velocity	$v_{y,max}$	$1\,\frac{\mathrm{m}}{\mathrm{s}}$	max. y-velocity
$v_{z,min}$	$-1\,\frac{\mathrm{m}}{\mathrm{s}}$	min. z-velocity	$v_{z,max}$	$1\,\frac{\mathrm{m}}{\mathrm{s}}$	max. z-velocity
$\omega_{j,min}$	$-0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. yaw rate	$\omega_{j,max}$	$0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. yaw rate
$\varepsilon_{j,min}$	$-0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	min. pitch rate	$\varepsilon_{j,max}$	$0.25\,\frac{\mathrm{rad}}{\mathrm{s}}$	max. pitch rate

Table 6.2: Overview of the values of particular constants used for the RTI experiment presented in Section 6.2. The values of parameters with the j index are applied to all robots in the formation. The brief descriptions provided in the table are used to remind the meaning of particular symbols and they are correctly defined in the previous sections.

Constant	Value	Brief description
T_{stab}	0.2 s	RTI sampling time
d_{RTI}	0.1 m	RTI sampling distance
v_s	7	number of vertical samples
d_l	5.5 m	desired lighting distance
$\lambda_{h,min}$	$-65°$	minimum horizontal lighting angle
$\lambda_{h,max}$	$65°$	maximum horizontal lighting angle
$\lambda_{v,min}$	$-50°$	maximum vertical lighting angle
$\lambda_{v,max}$	$50°$	maximum vertical lighting angle

Table 6.3: Overview of the values of particular constants, connected with the RTI scanning procedure, used for the RTI experiment presented in Section 6.2. The brief descriptions provided in the table are used to remind the meaning of particular symbols and they are correctly defined in the previous sections.

It is hard to evaluate the quality of the resulting PTM based on some measurable properties. Thus, its quality has to be assessed by subjective considerations of experts who are the expected users of the produced PTMs. Based on their opinion, we can claim that the precision of achieved lighting is sufficient, and the set of generated RTI positions forms sufficiently good coverage of lighting positions to produce the good-quality PTM.



(a) side view

(b) side view

(c) xy-view

(d) yz-view

(e) yaw - follower

(f) pitch - follower

Figure 6.9: The generated RTI positions and the trajectory flown by the follower carrying the light during the RTI scanning procedure. The blue dots mark particular RTI positions, the red dot marks the position of scanned object and the blue line shows the trajectory of the follower. The video of the experiment is available at `http://mrs.felk.cvut.cz/kratky2019thesis`.

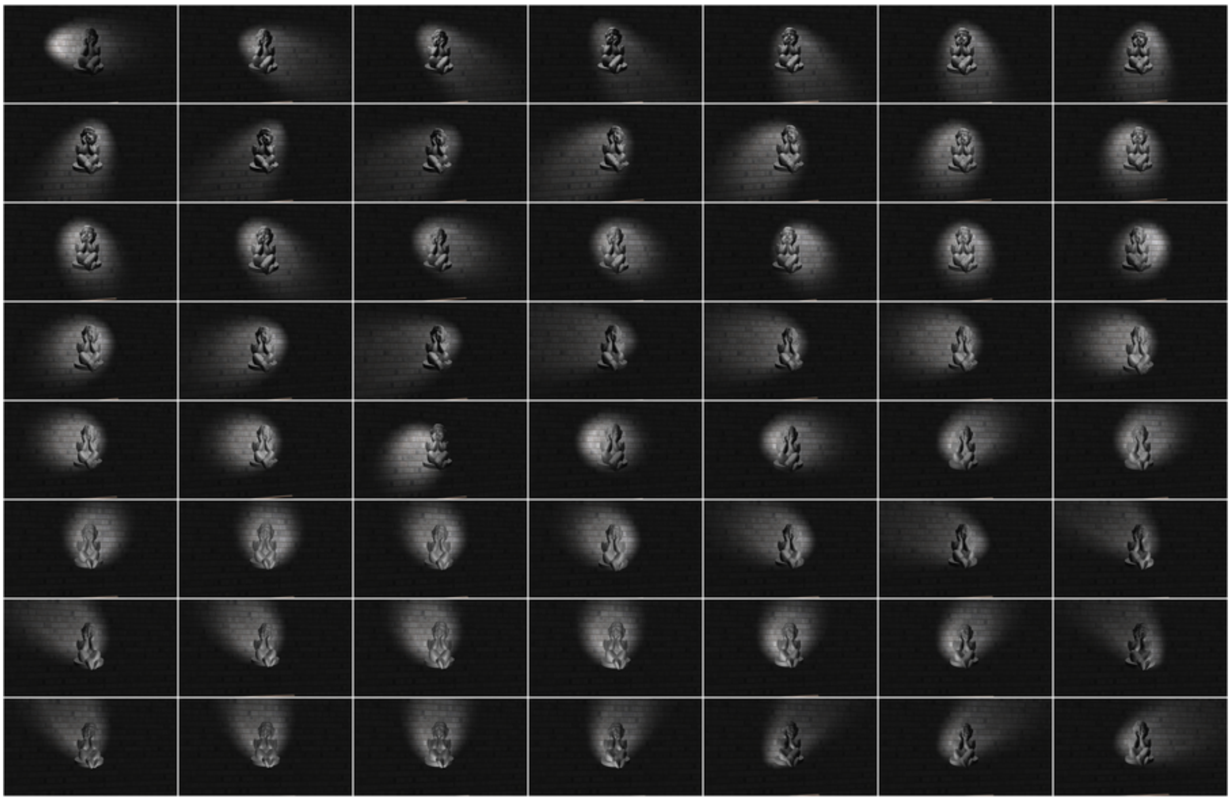
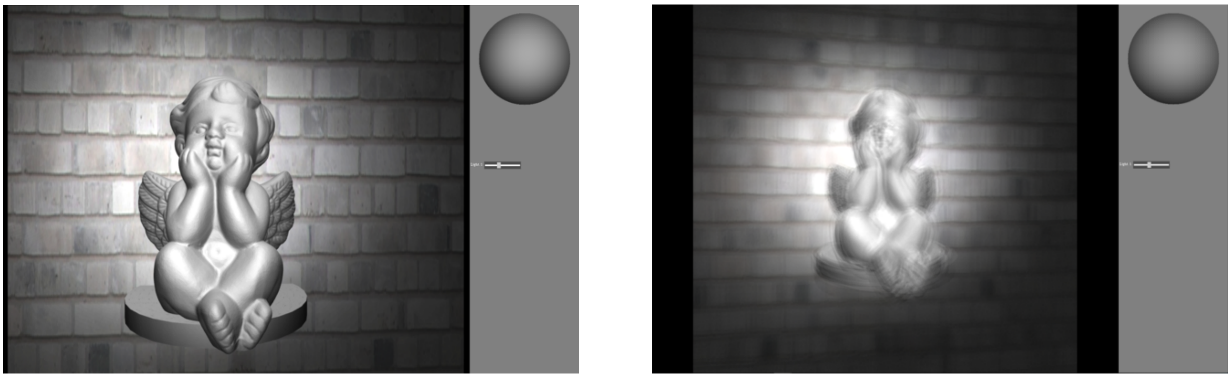


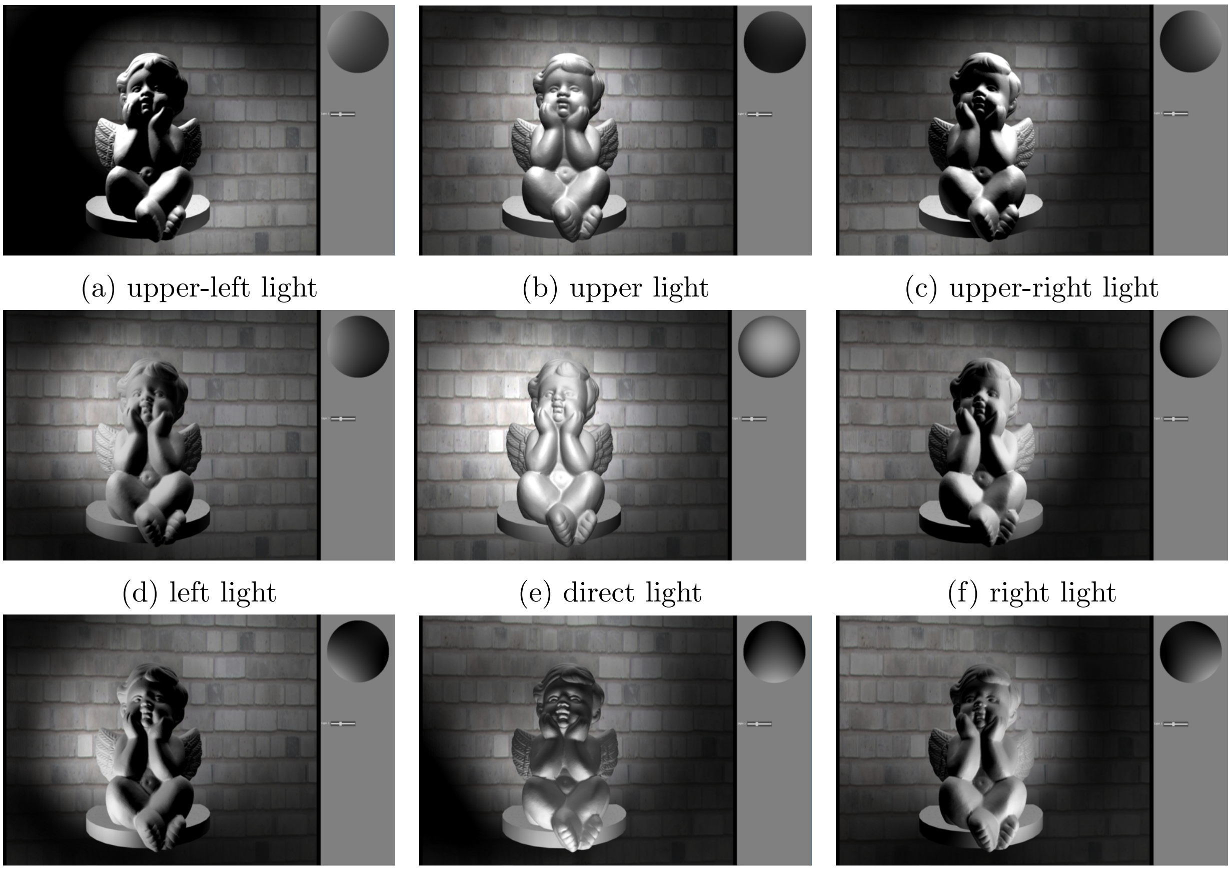
Figure 6.10: The set of images taken by the onboard camera mounted on the leading UAV during the RTI experiment described in Section 6.2.



(a) static camera

(b) onboard camera

Figure 6.11: Comparison of PTMs representations of the image of scanned object obtained from the properly registered images (simulated by static camera) (a) and from the onboard camera without any post-processing (b).



(a) upper-left light

(b) upper light

(c) upper-right light

(d) left light

(e) direct light

(f) right light

(g) bottom-left light

(h) bottom light

(i) bottom-right light

Figure 6.12: Presentation of PTM representation of the images of scanned object obtained from the images taken during the RTI experiment performed in the realistic simulator Gazebo. The program PTMViewer is used for the view of the PTM.

6.3 Real-world experiment

After successful testing of the proposed approach for performing RTI scanning in the simulation, the real-world experiment was conducted. The aim of this experiment, which is described in this section, was to determine the applicability of the proposed and implemented approach to RTI method in the real-world environment with the use of available UAVs of the Multi-robot Systems group at Faculty of Electrical Engineering at Czech Technical University in Prague.

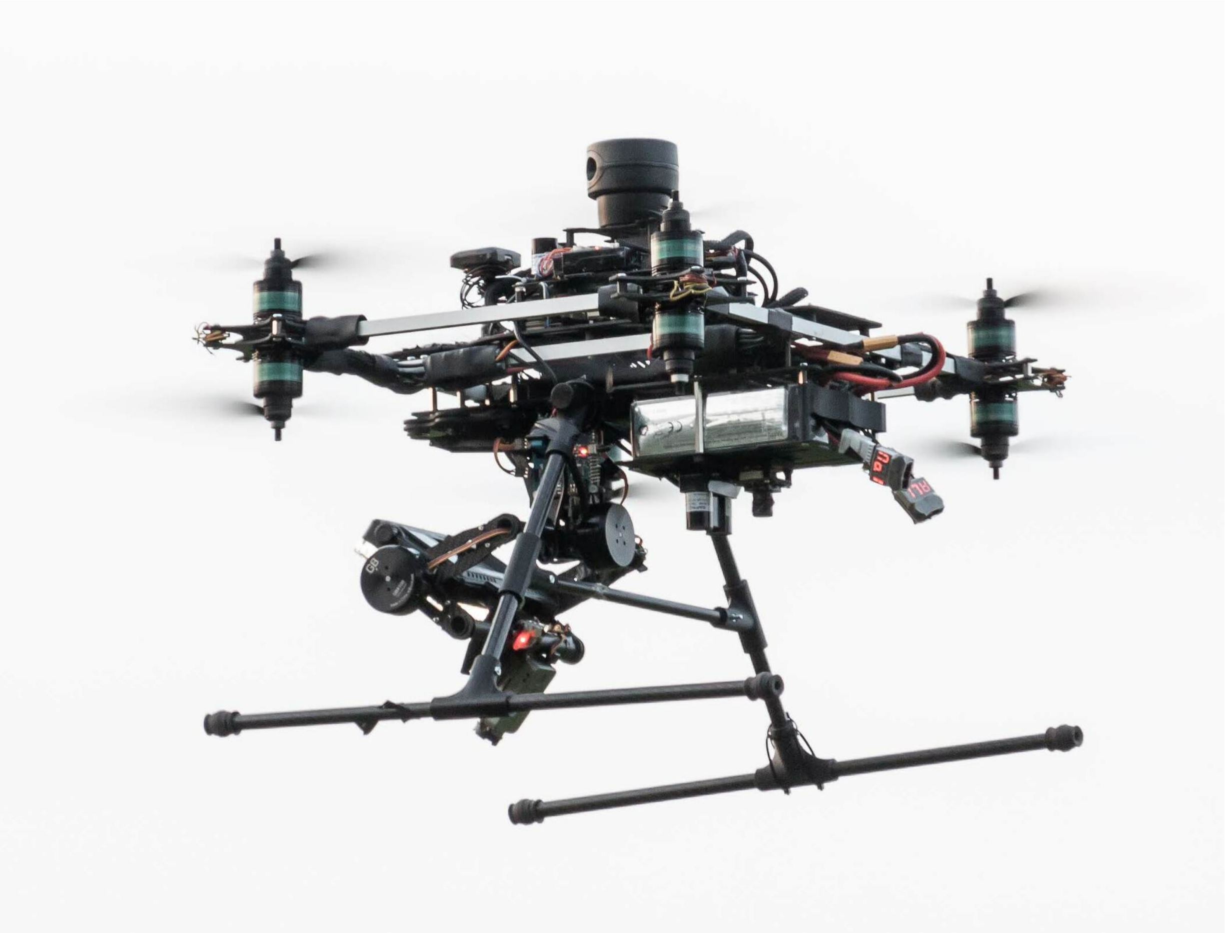


Figure 6.13: Specialized platform developed within Multi-robot Systems group for scanning of historical buildings interiors during its deployment in the experiment presented in Section 6.3.

Since the UAVs which are equipped with necessary sensors for indoor localization and holders for cameras and lights were not available, it was decided to perform the experiment in the outdoor environment. The UAV with the DJI F450 frame equipped with the high-resolution camera was deployed as the leader. The specialized platform for scanning of interiors of buildings, developed within Multi-robot Systems group was used as the follower carrying the light. This platform (Figure 6.13) is equipped with the gimbal, which is prepared for stabilization of a camera. Nevertheless, during our experiment, we have used it as the holder for the light, and its orientation was controlled by the formation control system.

Both UAVs were localized with the use of Global Positioning System (GPS). Although the Real-time Kinematic GPS (RTK GPS) was available, we assume that to achieve similar precision of localization as the RTK GPS is challenging for all available onboard indoor localization systems and therefore the localization based on the simple GPS will be better outdoor equivalent for the indoor localization methods.

The mission plan for the experiment consists of the 6 m long flight of the leader, which scans one OoI (outdoor gas heater) and in the middle of the trajectory, the RTI scanning of this object is required. The overview of constants that define the desired RTI scanning procedure is presented in Table 6.4. The setup of the experiment is illustrated in Figure 6.14.

Constant	Value	Brief description
T_s	0.2 s	RTI sampling time
d_{RTI}	0.1 m	RTI sampling distance
v_s	5	number of vertical samples
d_l	4.5 m	desired lighting distance
$\lambda_{h,min}$	$-70°$	minimum horizontal lighting angle
$\lambda_{h,max}$	70°	maximum horizontal lighting angle
$\lambda_{v,min}$	$-10°$	maximum vertical lighting angle
$\lambda_{v,max}$	70°	maximum vertical lighting angle

Table 6.4: Overview of the values of particular constants connected with the RTI scanning procedure and used for the outdoor RTI experiment presented in section Section 6.3. The brief descriptions provided in the table are used to remind the meaning of particular symbols and they are correctly defined in the previous sections.

Although the best conditions for performing RTI scanning is absolute dark so that the light source carried by the follower is the only source of the light in the experimental scenario, first tests were conducted during the day which enables the monitoring of the experiment by human operators. Already during these tests, the precision of localization was found as insufficient. The problem was not only the precision of estimation of the position but also the determination of the correct orientation. During the experiment, the estimations of position and orientation of the follower were so imprecise that the light source pointed out of the desired OoI. Thus, the experiment was ended after a few minutes due to the danger of destroying the hardware. The sequence of images from the camera that captures the whole experimental scene is provided in Figure 6.15. The video from the experiment is available at `http://mrs.felk.cvut.cz/kratky2019thesis`.

Despite the fact that the experiment successfully tests only partially the correctness

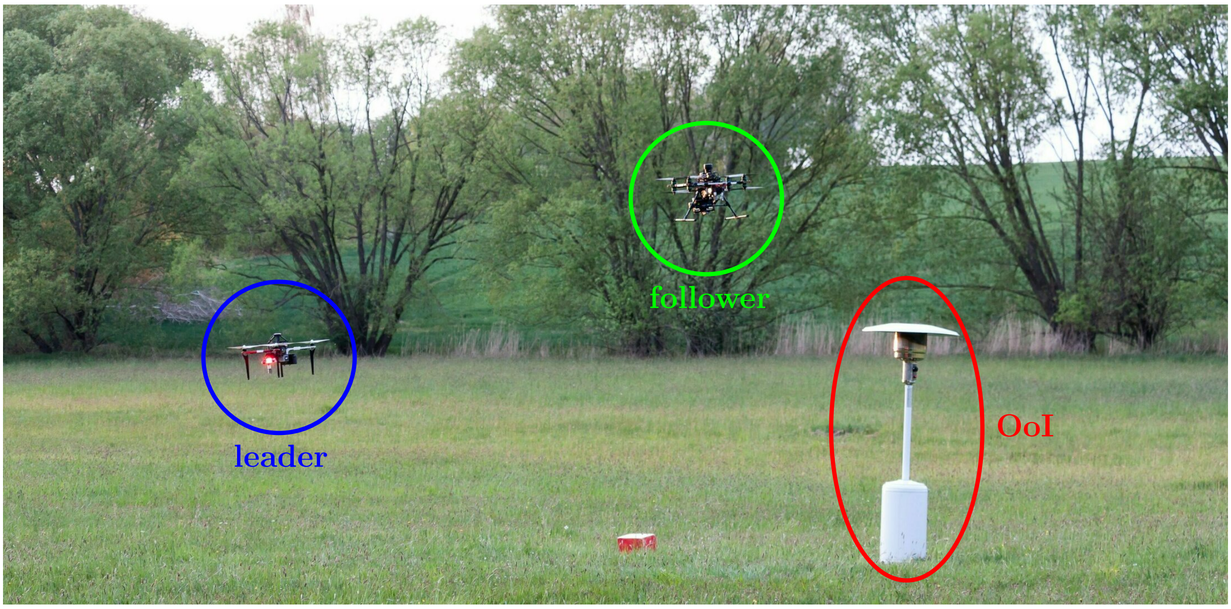


Figure 6.14: The real-world scenario used within the experiment described in Section 6.3.

of the implementation of the mission controller for its deployment on the real UAVs and the first few steps of the RTI scanning procedure, it shows the importance of the precision of localization for this application. Even though the precision of indoor localization methods should be sufficient for the estimation of position, the small error in position and orientation estimation can lead to the presence of OoI on the edge of the camera FoV instead of its center. This situation can be improved by introducing the detection of the desired OoI in the image and adjustment of the orientation of camera or light based on this observation, which is one of the proposed future works of this project.

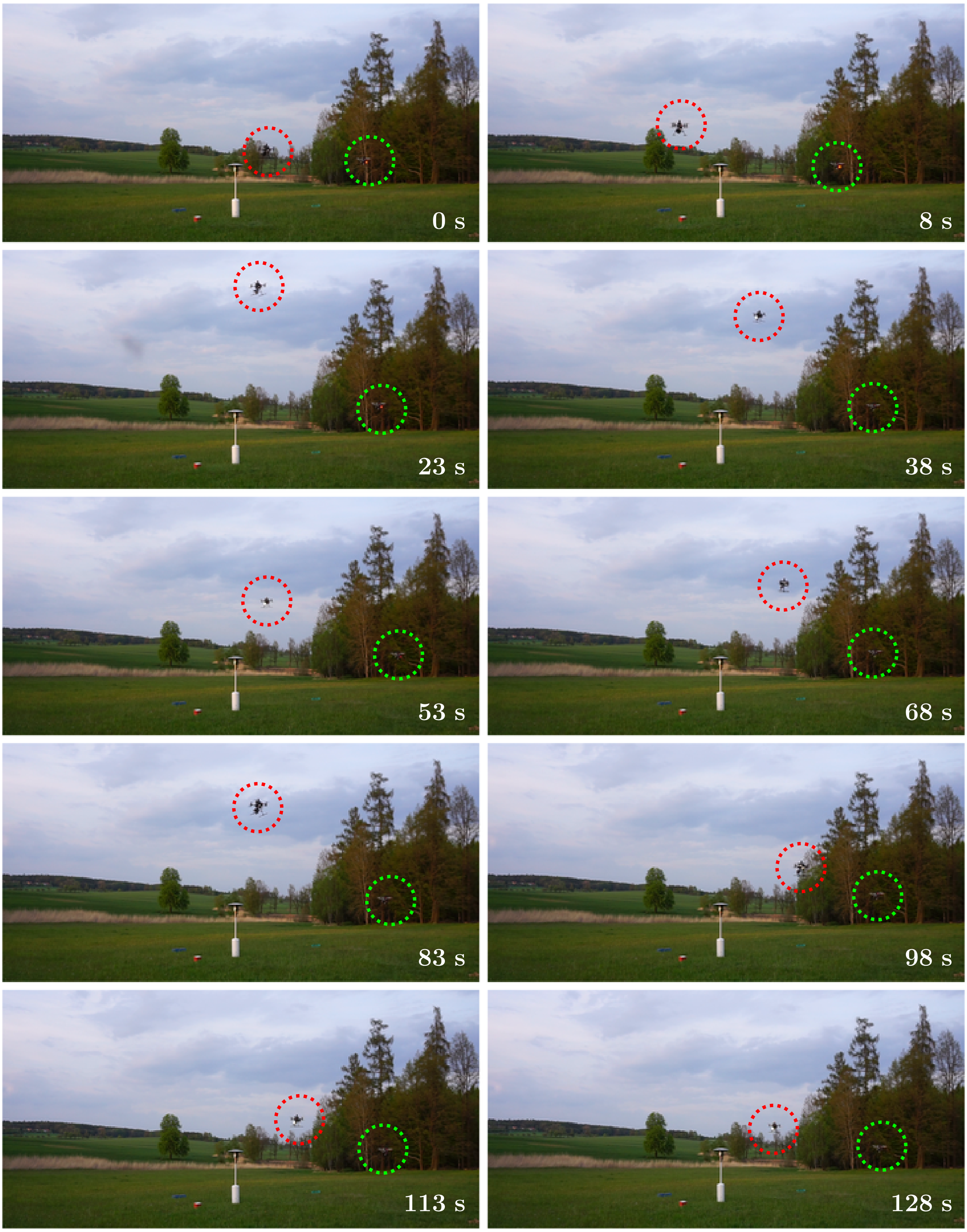


Figure 6.15: The sequence of images of the experimental scene taken by static camera during the experiment presented in Section 6.3. The green and red circles highlight the leader carrying the camera and the follower carrying the light respectively.

Chapter 7

Conclusion

In this thesis, we have designed and implemented a system for safe autonomous documentation of historical building interiors based on a usage of the cooperative formation of multi-rotor UAVs. The system makes use of the method for formation control originally described in [1] and [18], which was further extended within this work. The safety features of the system are provided by the mission controller, which controls the cooperation of particular UAVs in the formation, implements proper reactions on the occurrence of particular faults and failures, and also provides the interface for the operators who monitor the course of the mission.

The verification of the system was realized in various experimental scenarios in realistic robotic simulator Gazebo. The system is also prepared for a real-world experiment with UVDAR [15], the method for relative localization. Although the UAVs with necessary equipment for their deployment in an indoor environment (including the UVDAR system) were not available, the real-world experiment was conducted in a modeled outdoor environment. Unfortunately, the experiment was not successfully completed due to the insufficient precision of localization. Nevertheless, the results from this experiment and also from simulations are presented in Chapter 6 in details and they are available in the multimedia form at `http://mrs.felk.cvut.cz/kratky2019thesis`.

All of the tasks were successfully completed according to the thesis assignment:

- The system for stabilization of formations of UAVs in the task of cooperative filming was extended and enhanced in Chapter 3.
- The sources of potential system failures were identified and the system which ensures safe carrying out of the mission was designed and implemented in Chapter 4.
- The Reflectance Transformation Imaging method used for documentation of historical buildings was implemented in Chapter 5.
- Verification of the system was performed within various scenarios in realistic simulator Gazebo.

- The system is prepared for real-world experiment with UVDAR, which was not conducted due to the unavailability of the multi-rotor helicopters.

The proposed system enables to perform the scanning of historical artefacts by the implementation of three commonly used lighting methods applied by restorers and historians - three points lighting, method using raking light, and reflectance transformation imaging method. Further, it is integrated into the system for UAV control developed by Multi-robot Systems group at Faculty of Electrical Engineering at Czech Technical University in Prague and is prepared for its employment in the task of documentation of hardly accessible places in historical buildings.

Bibliography

[1] M. Saska, V. Krátký, V. Spurný, and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, September 2017.

[2] F. Kovacs, "Documentation of cultural heritage; techniques, potentials, and constraints," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W7, pp. 207–214, 2015.

[3] R. Al-Ruzouq, *Photogrammetry for Archaeological Documentation and Cultural Heritage Conservation.* April 2012.

[4] V. Badenko, D. Zotov, and A. Fedotov, "Hybrid processing of laser scanning data," *E3S Web of Conferences*, vol. 33, p. 1047, January 2018.

[5] R. Zlot, M. Bosse, K. Greenop, Z. Jarzab, E. Juckes, and J. Roberts, "Efficiently capturing large, complex cultural heritage sites with a handheld mobile 3d laser mapping system," *Journal of Cultural Heritage*, vol. 15, no. 6, pp. 670 – 678, 2014.

[6] P. S. Blaer and P. K. Allen, "Data acquisition and view planning for 3-d modeling tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 417–422, October 2007.

[7] K. Themistocleous, M. Ioannides, A. Agapiou, and D. Hadjimitsis, "The methodology of documenting cultural heritage sites using photogrammetry, uav and 3d printing techniques: The case study of asinou church in cyprus," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9535, June 2015.

[8] Y. HU, Y. CHEN, and Z. WU, "Unmanned aerial vehicle and ground remote sensing applied in 3d reconstruction of historical building groups in ancient villages," in *International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*, pp. 1–5, June 2018.

[9] Y. Jo and S. Hong, "Three-dimensional digital documentation of cultural heritage site based on the convergence of terrestrial laser scanning and unmanned aerial vehicle

photogrammetry," *ISPRS International Journal of Geo-Information*, vol. 8, p. 53, January 2019.

[10] A. Gulec Korumaz, M. Korumaz, G. Tucci, V. Bonora, W. Niemeier, and B. Riedel, "Uav systems for documentation of cultural heritage," in *ICONARCH I - International Congress of Architecture - Innovative Approaches in Architecture and Planning*, pp. 419–430, October 2014.

[11] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *International Conference on Computer Vision (ICCV)*, 2017.

[12] N. Hallermann, G. Morgenthal, and V. Rodehorst, "Vision-based monitoring of heritage monuments – unmanned aerial systems (uas) for detailed inspection and high-accurate survey of structures," in *Studies, Repairs and Maintenance of Heritage Architecture - STREMAH*, July 2015.

[13] I. Hyman, "Three point lighting: Learn how to use the key, fill, and back lights," 2011, accessed March 21, 2019.

[14] The National Gallery, "Raking light," 2019, accessed April 5, 2019.

[15] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of uavs using ultraviolet led markers," in *International Conference of Unmanned Aircraft System (ICUAS)*, 2018.

[16] V. Walter, N. Staub, A. Franchi, and M. Saska, "Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2637–2644, July 2019.

[17] T. Krajník, M. Nitsche, J. Faigl, T. Duckett, M. Mejail, and L. Přeučil, "External localization system for mobile robotics," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, November 2013.

[18] V. Krátký, "Model predictive control and stabilization of formation of quadrotors filming in dark conditions," bachelor's thesis, Czech Technical University in Prague, May 2017.

[19] M. Saska, V. Spurný, and V. Vonásek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, 2015.

[20] V. Spurný, T. Báča, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 998–1003, August 2016.

[21] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, “Coordination and Navigation of Heterogeneous MAV & UGV Formations Localized by a ”hawk-eye”-like Approach Under a Model Predictive Control Scheme,” *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.

[22] D. Meagher, “Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer,” October 1980.

[23] Y.-T. Su, J. Bethel, and S. Hu, “Octree-based segmentation for terrestrial lidar point cloud data in industrial applications,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 113, pp. 59 – 74, 2016.

[24] A. Fetzner, C. Frese, and C. Frey, “A 3d representation of obstacles in the robots reachable area considering occlusions,” in *41st International Symposium on Robotics*, pp. 1–8, June 2014.

[25] J. Chen, T.Liu, and S. Shen, “Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1476–1483, May 2016.

[26] D. Stipanovic, P. Hokayem, M. Spong, and D. Siljak, “Cooperative avoidance control for multiagent systems,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, September 2007.

[27] Gurobi Optimization, “Gurobi Optimizer,” 2019, accessed April 24, 2019.

[28] IBM Corporation, *IBM ILOG CPLEX Optimization StudioCPLEX User's Manual*, 2016, accessed April 24, 2019.

[29] J. Mattingley and S. Boyd, “Cvxgen: A code generator for embedded convex optimization,” *Optimization and Engineering*, vol. 13, March 2012.

[30] A. Wächter and L. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, pp. 25–57, March 2006.

[31] S. G. Johnson, “The nlopt nonlinear-optimization package,” accessed April 24, 2019.

[32] P. E. Gill, W. Murray, and M. A. Saunders, “User's guide for snopt version 7: Software for large-scale nonlinear programming,” December 2015.

[33] C. T. Lawrence and J. L. Zhou, “User's guide for cfsqp version 2,” January 1997.

[34] T. Moulard, B. Chrétien, and E. Yoshida, “Software tools for nonlinear optimization — modern solvers and toolboxes for robotics,” *Journal of the Robotics Society of Japan*, vol. 32, pp. 536–541, January 2014.

[35] T. Malzbender, D. Gelb, and H. Wolters, “Polynomial Texture Maps,” in *Annual conference on Computer Graphics and Interactive Techniques*, pp. 519–528, ACM, 2001.

[36] A. Cosentino, “Macro photography for reflectance transformation imaging: A practical guide to the highlights method,” *e-conservation Journal*, no. 1, pp. 70–85, 2013.

[37] H. Mytum and J. R. Peterson, “The application of reflectance transformation imaging (rti) in historical archaeology,” *Historical Archaeology*, vol. 52, pp. 489–503, June 2018.

[38] D. Selmo, F. Sturt, J. Miles, P. Basford, T. Malzbender, K. Martinez, C. Thompson, G. Earl, and G. Bevan, “Underwater reflectance transformation imaging: a technology for in situ underwater cultural heritage object-level recording,” *Journal of Electronic Imaging*, vol. 26, no. 1, pp. 1–18, 2017.

[39] J. Miles, M. Pitts, H. Pagi, and G. Earl, “New applications of photogrammetry and reflectance transformation imaging to an easter island statue,” *Antiquity*, vol. 88, no. 340, p. 596–605, 2014.

[40] H. Mytum, K. Chapman, J. R. Peterson, and A. Cross, “Reflectance transformation imaging (rti) : Capturing gravestone detail via multiple digital images,” in *Association for Gravestone Studies Quarterly*, pp. 3–10, 2017.

[41] J. Valcarcel Andrés and M. Osca Pons, “Applications of reflectance transformation imaging for documentation and surface analysis in conservation,” *Internetional Journal of Conservation Science*, no. 4, pp. 535–548, 2013.

[42] D. Saunders, R. Collmann, and A. Borda, “Reflectance transformation imaging and imagej: Comparing imaging methodologies for cultural heritage artefacts,” in *Conference on Electronic Visualisation and the Arts (EVA)*, (Swindon, UK), pp. 350–357, BCS Learning & Development Ltd., 2017.

[43] Hewlett Packard company, “Polynomial texture mapping,” 2009, accessed March 5, 2019.

[44] Cultural Heritage Imaging, “Reflectance Transformation Imaging,” 2019, accessed March 5, 2018.

[45] E. B. Saff and A. B. J. Kuijlaars, “Distributing many points on a sphere,” *The Mathematical Intelligencer*, vol. 19, pp. 5–11, December 1997.

[46] M. Dellepiane, M. Corsini, M. Callieri, and R. Scopigno, “High quality ptm acquisition: Reflection transformation imaging for large objects,” in *International Symposium on VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage* (M. Ioannides, D. Arnold, F. Niccolucci, and K. Mania, eds.), pp. 179–186, Eurographics Association, November 2006.

[47] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," in *ACM National Meeting*, (New York, NY, USA), pp. 71.201–71.204, ACM, 1961.

[48] J. Monnot, V. T. Paschos, and S. Toulouse, "Approximation algorithms for the traveling salesman problem," *Mathematical Methods of Operations Research*, vol. 56, pp. 387–405, January 2003.

[49] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[50] D. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, "Concorde TSP solver," 2016, accessed March 26, 2019.

[51] R. Matungka, Y. F. Zheng, and R. L. Ewing, "Image registration using adaptive polar transform," *IEEE Transactions on Image Processing*, vol. 18, pp. 2340–2354, October 2009.

[52] Z. Wu and A. Goshtasby, "Adaptive image registration via hierarchical voronoi subdivision," *IEEE Transactions on Image Processing*, vol. 21, pp. 2464–2473, May 2012.

[53] J. Liang, Z. Liao, S. Yang, and Y. Wang, "Image matching based on orientation–magnitude histograms and global consistency," *Pattern Recognition*, vol. 45, no. 10, pp. 3825 – 3833, 2012.

Appendices

CD Content

Names of all root directories on CD are listed in Table 1.

Directory name	**Description**
thesis	the thesis in pdf format
thesis_sources	latex source codes
source	source codes of the implemented solution
videos	videos from experiments

Table 1: CD Content

List of abbreviations

Abbreviations used in this thesis are listed in Table 2.

Abbreviation	**Meaning**
AoV	Angle of View
CPU	Central Processing Unit
CTU	Czech Technical University
ETSP	Euclidean Traveling Salesman Problem
FEE	Faculty of Electrical Engineering
F&F	Faults and Failures
FoV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LKH	Lin-Kernighan Heuristic
MPC	Model Predictive Control
MRS	Multi-Robot Systems
OoI	Object of Interest
PTM	Polynomial Texture Map
ROS	Robot Operating System
RTI	Reflectance Transformation Imaging
RTK GPS	Real-Time Kinematic GPS
TSP	Traveling Salesman Problem
UAV	(multi-rotor) Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

Table 2: Lists of abbreviations