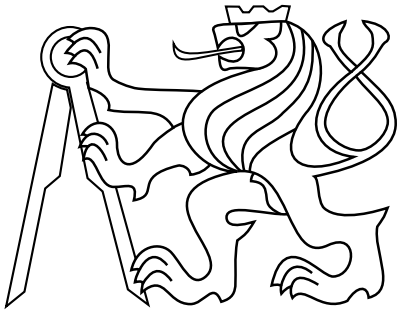
Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems



Autonomous Image Capturing of Large-Scale Objects by an Unmanned Aerial Vehicle

Bachelor's Thesis

Jan Wagner

Prague, January 2023

Study programme: Electrical Engineering and Information Technology
Branch of study: Artificial Intelligence and Computer Science

Supervisor: Ing. Vít Krátky

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Wagner Jan** Personal ID number: **492302**

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

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Autonomous Image Capturing of Large-Scale Objects by an Unmanned Aerial Vehicle

Bachelor's thesis title in Czech:

Autonomní snímání velkoplošných objektů pomocí bezpilotní helikoptéry

Guidelines:

The aim of the thesis is to design a method for the realization of autonomous image stitching by an Unmanned Aerial Vehicle (UAV) in environments with obstacles. The thesis focuses on the determination of proper positions for the image acquisition process of a given object and the application of multi-goal path planning to find paths efficiently connecting the positions. The following tasks will be solved:

- Define the determination of a set of camera positions covering a predefined area of interest as an optimization problem.
- Propose and implement a solution to this problem, considering required overlaps of images, desired resolution of the resulting image, occlusions and obstacle avoidance.
- Apply a multi-goal path planning algorithm to find a path connecting desired camera positions.
- Familiarize yourself with the system of Multi-Robot Systems group for stabilization and control of UAVs [1].
- Verify the proposed algorithm in the Gazebo simulator under ROS using the MRS system and models of historical buildings.
- Prepare a real-world experiment which will be conducted based on the availability of a real multi-rotor helicopter and permission to access a historical building suitable for the realization of the experiment.

Bibliography / sources:

[1] T. Báča, M. Petrlík, M. Vrba, V. Spurný, R. Pěnička, D. Heřt and M. Saska, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," Journal of Intelligent & Robotic Systems 102(26):1–28, 2021.
[2] M. Brown and D. G. Lowe, „Automatic Panoramic Image Stitching using Invariant Features," International Journal of Computer Vision 74:59–73, 2007.
[3] S. Isler, R. Sabzevari, J. Delmerico and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 3477-3484.
[4] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri and L. Seneviratne, "Coverage Path Planning with Adaptive Viewpoint Sampling to Construct 3D Models of Complex Structures for the Purpose of Inspection," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 7047-7054.

Name and workplace of bachelor's thesis supervisor:

Ing. Vít Krátký Multi-robot Systems FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **21.01.2022** Deadline for bachelor thesis submission: **10.01.2023**

Assignment valid until: **30.09.2023**

Ing. Vít Krátký	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

Declaration

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 10. January 2023

Signature: _____

Acknowledgments

Firstly, I would like to express my gratitude to my supervisor Ing. Vít Krátký for his feedback and guidance during the writing of this thesis. I am also thankful to my family for supporting me during my studies. Thank you.

Abstract

The world's cultural heritage has an important role in society. It is important to monitor the condition of cultural heritage, so it is preserved for future generations. A promising approach for monitoring such assets is the deployment of autonomous unmanned aerial vehicles (equipped with sensors) using methods of coverage path planning for path planning in image acquisition tasks. We consider the coverage path planning problem using autonomous unmanned aerial vehicles with the application of image mosaicing for scanning large-scale historical objects for periodic and detailed inspection.

In this thesis, the problem is formally defined and divided into two subproblems, with various constraints corresponding to requirements on image quality and safety during image acquisition process. We propose a solution to this problem based on computing the best feasible camera configuration according to a cost function for cell decomposition of the plane of interest and solving a traveling salesman problem instance, based on the Euclidean distance heuristic constructed from the camera positions. Then a planner is applied, to get collision-free paths connecting the camera positions. The solution is implemented in Python and is tested in the Gazebo simulator under the robot operating system using the MRS UAV system. The achieved results and their analysis are presented alongside identified potential future improvements of the proposed approach.

Keywords Unmanned Aerial Vehicles, Coverage Path Planning, Optimal Camera Placement, Autonomous Image Capturing

Abstrakt

Světové kulturní dědictví má důležitou roli ve společnosti. Je důležité monitorovat jeho stav, aby mohlo být zachováno pro budoucí generace. Slibný přístup pro monitorování těchto objektů je použití metody plánování pokrytí pomocí autonomních bezpilotních prostředků pro sběr fotografií. V této práci uvažujeme problém plánování pokrytí pro skenování velkoplošných historických objektů, za pomoci autonomních bezpilotních helikoptér s využitím pro periodickou a detailní inspekci takových objektů.

V této práci je problém plánování pokrytí formálně definován jako dva podproblémy s ohledem na různá omezení zaručující kvalitu obrázků a bezpečnost v průběhu sběru fotografií a je navrženo řešení. Prezentujeme řešení problému založené na výpočtu nejlepších vyhovujících prostorových orientacích (configuracích) kamery pomocí nákladové funkce pro rozdělení buňek cílené roviny, a řešení instance problému obchodního cestujícího, zkonstruované pomocí heuristiky Euklidovských vzdáleností mezi pozicemi kamer. Poté je aplikován plánovač pro získání bezkolizní cesty spojující pozice kamer. Řešení je implementováno v jazyce Python a testováno v simulátoru Gazebo s využitím robotického operačního systému a systému pro řízení bezpilotních prostředků skupiny MRS. Dosažené výsledky a jejich analýza je prezentována spolu s návrhy na možné vylepšení navrhovaného přístupu.

Klíčová slova Bezpilotní prostředky, Plánování pro pokrytí, Optimální umístění kamery, Autonomní snímání

Contents

List of Figures

Abbreviations

UAV Unmanned Aerial Vehicle

RANSAC RANdom SAmple Consensus

CPP Coverage Path Planning

TSP Traveling Salesman Problem

AGP Art Gallery Problem

OCPP Optimal Camera Placement Problem

SCP Set Cover Problem

WSCP Weighted Set Cover Problem

ILP Integer Linear Program

SIFT Scale Invariant Feature Transform

BFS Breadth First Search

MST Minimum Spanning Tree

IP Integer Programming

ILP Integer Linear Programming

OBB Oriented Bounding Box

LiDAR Light Detection And Ranging

Chapter 1

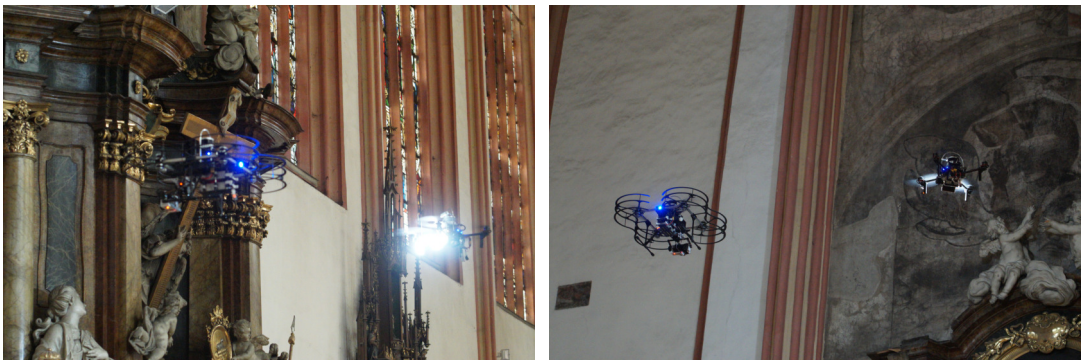
Introduction

The condition of world cultural heritage is degrading with time and the requirements on periodic inspection and digital preservation of cultural assets is on the rise. Historians need a tool to monitor asset's condition regularly and plan for renovation accordingly to prevent unrepairable damage (caused by long-term degradation). Many assets which need to be monitored periodically are very large. The prime examples are frescoes, murals and large paintings in churches. Others are, e.g., altars, statues, paintings and large stained glass windows.

The means of digital preservation vary. The most widely used methods are 3D laser scanning, photogrammetry, and photography in the visible spectrum. However, the quality of the results of photo-based methods is inherently dependent on the distance from the object, the angle between the photographed surface and lighting conditions. This problem can be addressed by using technologies such as Unmanned Aerial Vehicles (UAVs), kites, balloons and planes. These techniques enable capturing images from places inaccessible to human photographers without building an external infrastructure, such as scaffolding, or using mobile platforms. UAV-based techniques also significantly reduce the risk of damaging the asset during building such an infrastructure and the elapsed time from the proposal to the result. Many popular assets are also very frequently visited by sightseers, rendering timely methods unfeasible, due to imposing unacceptable long-term constraints on the regular use of the asset.

In recent years Unmanned Aerial Vehicles (UAVs), have been used in the heritage sector (and others, such as the industrial sector [3]) to monitor the condition of the roofs and artifacts within the exteriors of historical structures [13]. The UAV can be equipped with various sensors, such as high-resolution cameras, thermal cameras and Light Detection And Rangings (LiDARs), allowing for detailed analysis and acquisition of high-quality still images and motion videos. The advantages of autonomous UAVs are many. Autonomous board control allows for more precise planned maneuvers, which would be difficult for a human pilot. The piloting system also allows multi-robot cooperation [6] (see Fig. 1.1), which is nearly impossible for human pilots at scale.

This thesis focuses on designing a method for the realization of autonomous image stitching using UAVs in an environment with obstacles. This would allow for simple mission planning of image acquisition, with image mosaic resolution and obstacle-wise safety guarantees for the scanned environment and the equipment used. We propose a four-stage method, that localizes the plane of interest based on the provided oriented bounding box, decomposes the plane into cells, then finds the best camera configuration (according to loss function) which also satisfies constraints on image quality, completeness and connectedness, constructs an approximate traveling salesman problem instance with Euclidian distance heuristic and then finds a collision-free shortest path based on the TSP tour result.



(a) formation 1 (b) formation 2

Figure 1.1: Formations of unmanned autonomous vehicles at church of Saint Maurice in Olomouc performing image acquisition task

Chapter 2

Related work on coverage path planning

The Coverage Path Planning (CPP) is a problem of constructing a path visiting (covering) all desired volumes or nodes in space such that the robot does not collide with the environment (obstacles). The applications are mostly in robotics, such as vacuum cleaning robots, painter robots, various UAV performing environment mapping or image mosaics [2], demining robots [14], lawn mowers, automated harvesters and inspection of complex structures. One of the earliest works on CPP [30] defines requirements for CPP in a flat 2-dimensional environment, which applies to the 3-dimensional version of the problem:

1. Robot must move through or cover all the points of interest completely,
2. robot should cover the region without overlapping paths,
3. robot should use simple motion trajectories (straight lines, circles) for simplicity in control,
4. robot must avoid all obstacles,
5. an optimal path is desired under available conditions.

The CPP is closely related to the Art Gallery Problem (AGP) from computational geometry. AGP is a problem of placing stationary guards/watchmen in an environment (called an art gallery), in such a way, that all parts of the gallery can be observed by at least one guard. Each guard can see to an unlimited distance in any direction. Václav Chvátal proved in [36] that only $\lfloor \frac{n}{3} \rfloor$ guards are needed to cover an art gallery represented by a simple polygon. By [29] is AGP NP-hard, so mostly approximate solutions are applicable in larger instances. A similar problem to AGP is Optimal Camera Placement Problem (OCPP), which is a more tightly constrained variant of AGP. Given a surveillance area (2D or 3D) and set of feasible camera locations and orientations (from now on, we will refer to this as camera configuration), find camera configurations that optimize a predefined objective and satisfy a set of application-specific constraints. This problem has several applications, such as area surveillance, target tracking, photogrammetry or traffic and crowd analysis [10]. This problem is often formulated as Set Cover Problem (SCP) (Sec. 4.9). While AGP and OCPP are a substantial part of CPP, they only focus on the coverage part of the problem. Common approach to CPP is to apply methods for solving the AGP or OCPP and then use multi-goal path planning algorithms, such as Traveling Salesman Problem (TSP) (Sec. 4.8), to solve the problem of finding a path that visits all the camera configurations.

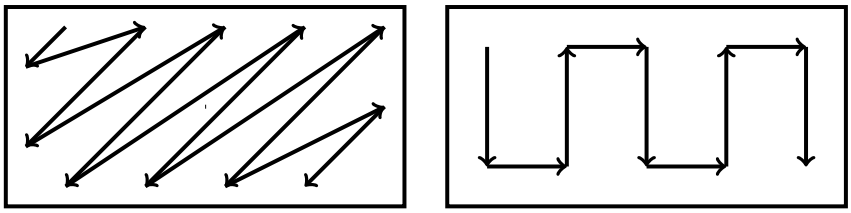
2.1 Knowledge aquisition

CPP can be divided into two types according to the manner of knowledge acquisition. Offline with a priori knowledge of the environment, and online, without the a priori knowledge.

Offline algorithms can typically produce better solutions. However, for some applications acquiring a priori knowledge about the environment is not possible.

2.2 Cellular decomposition

Cellular decompositions are the core of most CPP algorithms[1]. Cellular decomposition is a technique for dividing polygons into sections, which are then used to solve the AGP. The simplest technique is the exact cellular decomposition. Exact cellular decomposition decomposes the space into simple, uniform, nonoverlapping regions. By [14], two regions are adjacent if they share a common boundary. This adjacency graph is then used for path planners to construct an exhaustive walk on this adjacency graph, often by simple space-filling curve paths. For example the zigzag path or the left/right turn path (see Fig. 2.1). Note that these techniques do not necessarily consider the exhaustive walk as a circular sequence as in TSP. Another type of decomposition is trapezoidal decomposition. Trapezoidal decomposition



(a) zigzag path (b) left/right turn path

Figure 2.1: Example of zigzag and left/right turn path.

works by slicing the polygon into successive slices at vertices in slices parallel to some of the edges of the polygon or other lines. This produces many cells that could be intuitively merged without the need to deviate or discontinue the space-filling into some positioning maneuvers. This technique was used by authors of [19].

Problems of trapezoidal decomposition are addressed in Choset & Pignon's proposition of the Boustrophedon decomposition [27], which is very similar to trapezoidal decomposition. Instead of slicing at every vertex, the slice occurs only at so-called critical vertices, which they have defined as vertices, in which the slice can be cut both above and below.

2.3 Cellular decomposition using set cover

Cell decomposition can be solved with mathematical programming (SCP). However, the results are not necessarily better. For instance, [17] achieved better results with naive cellular decomposition than by specifying the problem as SCP.

2.4 Optimization criterions

In many applications, the relevant metric for determining the optimal path is the path length. The reason for this is mostly that longer path results in longer duration of the exper-

[1] One of the algorithms that does not rely on cellular decomposition is [15].

iment and larger battery aspect of robots performing the experiment. This can be in many applications very critical metric due to the limited capacity of robots performing the experiment.

In [24] the authors have formulated a criterion based on the number of turns[2] in a path, arguing that sharp direction changes can be costly, especially for ground-based robots. They argue that in some cases the expense of turns in robotic motion (stopping, deaccelerating, turning and accelerating in a new direction) has a more significant impact on the duration than the traveled distance. However, for some applications like UAV, the length of a path is not the only variable that influences power consumption. For example altitude differences (elevation) or the weather is a big factor in power consumption. Publications [12] and [17] have formulated the optimization problem to minimize the energy (and other resources) needed to perform an experiment.

[2]The criterion number of turns is a relaxed version of the sum of subregion altitudes covering a region.

Chapter 3

Problem formulation

Given a predefined area of interest (oriented bounding box $(min_{x,y,z}, max_{x,y,z}, \varphi)$), camera intrinsics, minimal image overlap ratios, and point cloud representing the environment, choose a subset of camera positions (camera extrinsics) $C^* \subseteq C$ from all possible configurations $C = \{(\mathbf{p}, \theta) \,|\, \mathbf{p} \in \mathbb{R}^3, \theta \in \langle 0, 2\pi)\}$, such that it satisfies the following constraints[1]

- $\mathbf{P_1}$ *Prohibited regions:* no camera position is closer to any obstacle than ϵ,
- $\mathbf{P_2}$ *View angle:* no image has been taken under view angle greater than θ,
- $\mathbf{P_3}$ *Resolution:* the resolution of any image taken is at least ϕ,
- $\mathbf{P_4}$ *Focus:* all points in all images taken are in the depth of field of the camera (in focus),
- $\mathbf{P_5}$ *Visibility:* in none of the images taken there is an object occluding the plane of interest,
- $\mathbf{P_6}$ *Overlaps:* for every image taken, there exists at least one other image with geometrical overlap with it, with at least ρ_x and ρ_y overlap ratios in direction of axis x and y in its local frame
- $\mathbf{P_7}$ *Completeness:* the collection of images taken captures all relevant parts of the plane of interest,
- $\mathbf{P_8}$ *Connectedness*[2]*:* every image is connected[3] to all other images,

and $\sum_{c^* \in C^*} \mathrm{f}(c^*)$ is minimal. Here, $\mathrm{f}(c^*)$ is an image cost function that denotes the quality of the given camera configuration. Formally this problem can be defined as a constrained optimization problem

$$C^* = \begin{array}{ll} \underset{X \subseteq C}{\text{minimize}} & \sum_{x \in X} \mathrm{f}(x), \\ \text{subject to} & \bigwedge_{i=1}^{8} \mathbf{P_i}. \end{array} \tag{3.1}$$

To the camera configurations $C^* = \{c_1^*, \ldots, c_k^*\}$ where $c_i = (\mathbf{p}_i, \theta_i)$ for $i \in \{1, \ldots, k\}$, apply multi-goal path planning, to find a minimum length path connecting camera configurations C_i^*. Formally, this problem can be formulated as

$$\pi^* = \underset{\pi \in \Pi}{\text{minimize}} \sum_{i=1}^{|C^*|-1} \mathrm{path}(C_{\pi_i}^*, C_{\pi_{i+1}}^*), \tag{3.2}$$

where Π is set of all permutations of indices of C^* and path is a function returning length of the shortest collision-free path connecting two configurations.

[1] These constraints are a more strict version of constraints defined in [31]. For this application is needed to capture all the relevant parts of the plane of interest and the captured images have to be connected to enable later image stitching.

[2] This is needed to connect all taken images into an image mosaic.

[3] Images s and d are said to be connected if they overlap, or there exists a sequence of images $[\mathrm{I}_0, \ldots, \mathrm{I}_k]$ such that s and I_0 are connected, I_j and I_{j+1} are connected $\forall j \in \{0, \ldots, k-1\}$ and I_k and d are connected.

Chapter 4

Preliminaries

4.1 Random sample consensus

Model fitting is a common problem in computer science. It is a problem where data need to be either filled in the gaps between samples with a good guess, or they need to be extrapolated based on the trend of the data. There are many techniques for approaching this task, and one that is very common is RANdom SAmple Consensus (RANSAC) [20]. RANSAC is a randomized algorithm, which takes the minimum number of points to build up the fitted model, counts the samples within a margin of error and then chooses the best result among all iterations. Because of this, RANSAC is fairly resistant to outliers. It is common practice to further refine the obtained model by other fitting techniques (e.g., the least squares method [33]).

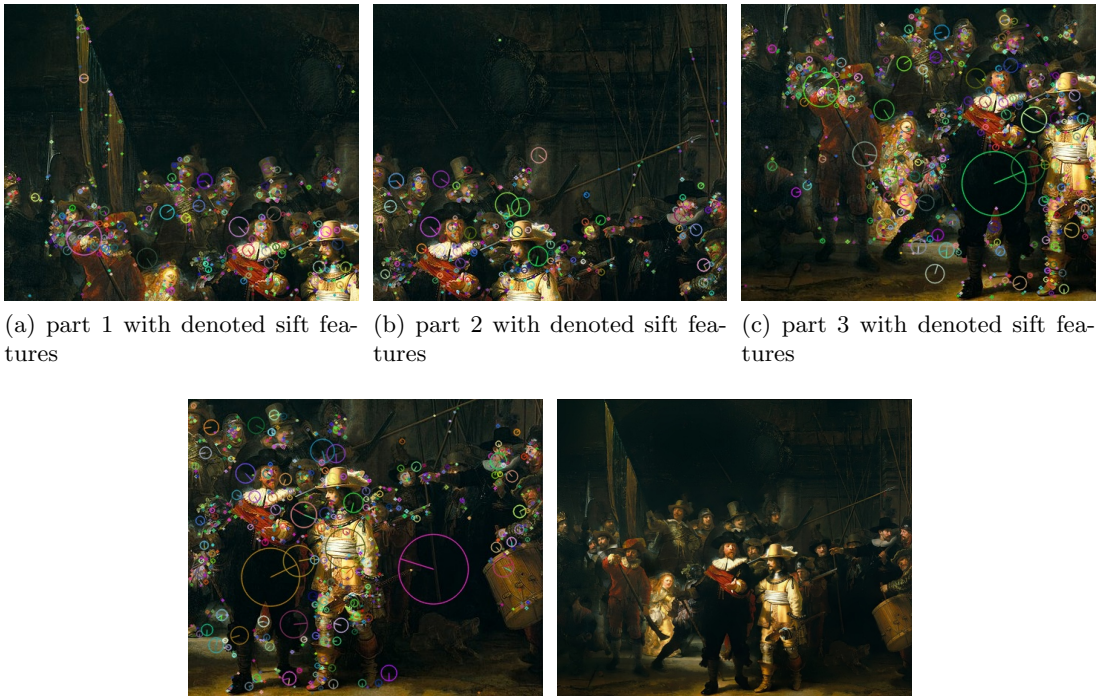
4.2 Image mosaicing

Photographing large-scale objects in great detail has its difficulties. The larger the object, the farther away the camera must be to capture the entire object. However, this decreases the ratio of pixels per m^2 in the image corresponding to the surface of the object. This problem can be partially avoided by using a camera with a higher resolution. However, such an approach is impractical in cluttered environments, because moving farther away is not always feasible due to possible occlusions and collisions. An alternative solution for general environments is to apply image mosaicing. Image mosaicing is a technique for the spatial composition of images into a composite image. The composition of the image is done using dominant feature pairs between two images and using them to compute a homography transform defining the transition between the images.

4.3 Image features

For some time, computer vision researchers have been looking for a way to algorithmically resolve if two images capture the same object. The consensus has taken a direction toward a technique called image features[1], which are specific configurations of pixels that can be recognized through a set of images. Some basic features include edges and corners, which are obtained by a convolution of specific operators and then by further processing of the result. Simple operators for edge and corner detection have their advantages, but they do

[1] If we exclude neural networks because they have to be trained. The practicality of image features is, that they do not need training.



(a) part 1 with denoted sift features (b) part 2 with denoted sift features (c) part 3 with denoted sift features

(d) part 4 with denoted sift features (e) image stitched from parts 1-4

Figure 4.1: Image sectioned into 4 parts and then stitched together [4]

not scale well and are use-specific. Some pictures simply do not contain edges or corners due to the nature of the geometry in the scene captured.

Many of the classic feature-related problems are solved by the Scale Invariant Feature Transform (SIFT) detector [26]. SIFT features[2] are scale-invariant and orientation-invariant. This enables more complex features, that can be tracked within the images more reliably. Extracting SIFT interest points (see Fig. 4.1) is done by a difference of scaled Gaussians (approximation of NLoG operator) using $s^i\sigma$ ($\mu = 0$), forming $W_{x,y,j}$ array and finding local extrema in the array. This results in weak extrema interest point candidates. To obtain only strong candidates, thresholding is applied, resulting in $S_{x,y,i-1}$ array. The scale invariance is achieved by scaling the feature blob[3] according to the scale of matching extrema. Orientation invariance is achieved by constructing a histogram from gradient vectors of pixels within the feature blob. The maximum of the histogram is then called a principal orientation and can be used to undo the effects of rotation.

4.4 Homography

Homography (or a general projective transform) is a 3x3 matrix with 8 degrees of freedom that describes a transformation between two planes. It is also often referred to as a perspective transformation. The result of applying this transformation to image results in an illusion like the image was taken from a different point of view. However, a homography

[2]There are also other popular feature detectors, such as [16], [22].

[3]A feature blob is called the immediate neighborhood of a feature point, proportionate in size to s.

cannot introduce new details into the image, so the resulting image will not have the same level of detail as if it were taken from a different position. This effect will be stronger for larger perspective changes. The homography matrix is defined as

$$\mathrm{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \tag{4.1}$$

Matrix H is generally normalized so that $\sum_{i \in I} \sum_{j \in J} h_{i,j}^2 = 1$. To transition from one perspective of an image s to another image d, where the pixels at x, y are using the corresponding subscripts s and d, we have to apply the homography to homogeneous coordinates. In other words, we need to consider the following expression

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} z_d x_d \\ z_d y_d \\ z_d \end{bmatrix} = \mathrm{H} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \tag{4.2}$$

for every pixel to transform the entire image. Homographies are closed under composition, so the transformation between perspectives of images s and d can be computed as composition of two intermediate transformations $\mathrm{H}_{s,i}$ and $\mathrm{H}_{i,d}$ as follows

$$\mathrm{H}_{s,d} = \mathrm{H}_{i,d} \mathrm{H}_{s,i}. \tag{4.3}$$

To compute a homography, at least $i = 4$ related pairs[4] from the source image to the destination image are needed. Then a possibly overdetermined linear system can be arranged.

$$A\mathbf{h} = \begin{bmatrix} x_s^1 & y_s^1 & 1 & 0 & 0 & 0 & -x_d^1 x_s^1 & -x_d^1 y_s^1 & -x_d^1 \\ 0 & 0 & 0 & x_s^1 & y_s^1 & 1 & -y_d^1 x_s^1 & -y_d^1 y_s^1 & -y_d^1 \\ & & & & & \vdots & & & \\ x_s^i & y_s^i & 1 & 0 & 0 & 0 & -x_d^i x_s^i & -x_d^i y_s^i & -x_d^i \\ 0 & 0 & 0 & x_s^i & y_s^i & 1 & -y_d^i x_s^i & -y_d^i y_s^i & -y_d^i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{4.4}$$

The homography is then found as a solution to the linear system (4.4), which can be solved as the constrained least-squares problem

$$\begin{array}{lll} \underset{\mathbf{h}}{\text{minimize}} & ||A\mathbf{h}||^2 = & 0, \\ \text{subject to:} & ||\mathbf{h}|| = & 1. \end{array} \tag{4.5}$$

4.5 Shortest path search

Considering a weighted graph, the shortest path between two nodes is defined as a sequence of edges that form a line and the sum of their weight is minimal among all existing lines connecting the given nodes. This problem can be tackled by many algorithms, one of them being Breadth First Search (BFS). However, more efficient algorithms have been proposed, such as Dijkstra's algorithm or A*, which are more efficient in exploration. A* with its heuristic-based search is a perfect fit for the search on large uniformly weighted graphs.

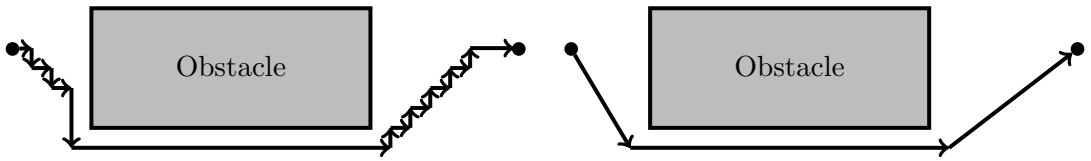
[4] i is an index of the feature pair. $i \gg 4$ is desired for a more robust estimate, resulting in an overdetermined linear system, which can be solved by constrained least squares.

4.6 A* algorithm

The A* algorithm is a variation of the Dijkstra algorithm, but instead of prioritizing the exploration of edges that results in a minimal sum of weights, the exploration criterion is g(node) + h(node), where g(node) is the true cost of the path from starting node to current node and h(node) is the heuristic function which estimates the distance from the current node to the goal node. This helps to better direct the search effort of the algorithm and to explore more efficiently. However, for A* to yield optimal paths, the heuristic (heuristic function) needs to have two properties. The first property is admissibility. A heuristic is admissible if it does not overestimate the true cost of reaching a goal. The second property is consistency. A heuristic is consistent if for any node N, its neighbors P and the heuristic function f(·), holds $h(N) \leq \text{dist}(N, P) + h(P)$ and $h(G) = 0$, where G is a goal node. Consistency is a stronger property than admissibility, so if a heuristic is consistent, it is also admissible. But it is not guaranteed that the admissible heuristic is also consistent.

4.7 Any angle path planning

A convenient way of constructing a graph for the search for the shortest path in an environment is to decompose the environment into cells equal in size. However, such discretization often yields the pathfinding algorithms to produce jagged paths (see Fig. 4.2), which are needlessly longer, than a straight line between two nodes. One approach to dealing with this problem is any angle pathfinding, where the search space is not exhaustive[5]. For example, there can be a valid edge from node A to node B, even if it is not explicitly stated in the graph. One such algorithm is Theta*. Theta* works similarly to A*, but a parent node does not need to be directly connected with the current node, as long as there is a clear line of sight[6]. This results in shorter and more straight paths.



(a) path style found by A* (b) path style found by Theta*

Figure 4.2: The path style of A* (Fig. 4.2a) and the path style of Theta* (Fig. 4.2b).

4.8 The traveling salesman problem

The traveling salesman problem represents the task of finding the optimal Hamiltonian cycle within a weighted graph. It is a very relevant problem with various applications, such as logistics, planning or PCB manufacturing. The formal definition is as follows.

[5]The meaning of a non-exhaustive graph is, that not all valid paths are present. In the real world, it is not practical to include all possible valid edges in a graph. We only consider adjacent nodes to be connected.

[6]Two objects have a clear line of sight if there is no object occluding visual contact between them. By considering a collision-free path instead of vision contact, we are introducing shorter paths into the graph.

Definition 4.8.1 (The traveling salesman problem)**.** Given a set of n integers $N = \{1, \ldots, n\}$ called cities, weighted edges denoted as $w_{i,j}$[7] from every city i to every city j, find a permutation $\pi^* = [i \,|\, \forall i \in N]$ called a tour, such that $\sum_{i=1}^{n-1} w_{\pi_i^*, \pi_{i+1}^*}$ is minimal. For practical purposes, we will define Integer Linear Programming (ILP) in a slightly different manner. Let $x_{i,j} = 1$ if edge from i to j is part of the tour, $x_{i,j} = 0$ otherwise.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} w_{i,j} x_{i,j}, \\
\text{subject to} \quad & \sum_{i=1, i \neq j}^{n} x_{i,j} = 1, && \forall j \in N, \\
& \sum_{j=1, j \neq i}^{n} x_{i,j} = 1, && \forall i \in N, \\
& \sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{i,j} \leq |Q| - 1, && \forall Q \subsetneq \{1, \ldots, n\}, |Q| \geq 2, \\
& x_{i,j} \in \{0, 1\}, && \forall i, j \in N.
\end{aligned}
\tag{4.6}
$$

The TSP is NP hard [34]. This implies that finding an optimal solution in a reasonable time is possible only for smaller instances with specialized solvers, such as Concorde [23]. To tackle larger instances, we have to yield the goal of finding an optimal solution and focus on heuristic algorithms, which are capable of finding an approximate solution in a reasonable time. One of the approximate solutions is the Nearest-Neighbor heuristic [28], which chooses a starting city and forms a Hamiltonian cycle by always picking the nearest neighbor as the next node. A very similar approach is the greedy heuristic, which always picks the shortest edge while rejecting edges that would form a cycle or would result in the node being of degree 3. Although these methods are generally fast and easy to implement, there are several algorithms with better performance. Such as the Christophedes method [28], [39], which uses Minimum Spanning Tree (MST), as the initial step, then removes odd-degree vertices by adding edges of perfect matching[8], resulting in a multigraph[9], where every node has an even degree of edges. From this intermediate result, the Eulerian path[10] is constructed by iterating over the edges, adding the connected vertices, and ignoring the vertices that are already included. The next class of algorithms for solving TSP are iterative improvement methods such as k-opt improvements [18] and random swaps, randomized approaches such as simulated annealing [32] or ant colony optimization [21]. One of the best known heuristic algorithms is Lin-Kernighan [37]. This method is based on iteratively computing the gain sequence, which is obtained by swapping elements of two subgraphs. Then the swaps that would maximize the cumulative gain are chosen, repeating these two steps until there are no more swaps, which would result in a positive gain.

4.9 The set cover problem

One of Karp's 21 problems [38], is the SCP. In general, it is a problem of reducing the number of actions (or resources being used), while achieving the same goal. Formal definition

[7] The edge weights $w_{i,j}$ are often formulated as a (distance) matrix, whose elements are the weights. The column and row of the element in the matrix represents the i and j respectively.

[8] Perfect matching is a problem of choosing edges, such that all vertices are covered.

[9] Multigraph is a graph that has multiple parallel edges.

[10] Not to confuse with the hamiltonian cycle, which seeks to connect all vertices. Eulerian path needs to use all edges forming a cycle.

is

Definition 4.9.1 (Set cover problem)**.** Given a set U (called the universe) and collection S_i of n sets, such that $\bigcup_{i=1}^{n} S_i = U$, find set J such that $\bigcup_{j \in J} S_j = U$ and $\sum_{i=1}^{n} x_i$ is minimal. Let $x_i = 1$ if S_i is chosen as part of the solution (if $i \in J$) and $x_i = 0$ otherwise. The ILP formulation of this problem is is

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} x_i, \\ \text{subject to} \quad & \sum_{i:e \in S_i} x_i \geq 1, \qquad \forall e \in U, \\ & x_i \in \{0,1\}, \quad \forall i \in \{1,\ldots,n\}. \end{aligned}$$

Let us propose an example. A civil engineer is faced with the task of building fire stations in a new city. The city has n districts (the universe U). Each city district needs a fire station within driving distance of 10 minutes from the district center. However, there is only a limited number of places where is it possible to build a fire station (the collection of S_i). Each place is within 10 minutes from at least one district center $|S_i| > 1$. The task is then to choose such combination (J) of fire stations, that all city districts are covered ($\bigcup_{j \in J} S_j = U$), while minimizing the number of fire stations built. A similar example can be made for the Weighted Set Cover Problem (WSCP), where the only modification is, that each fire station needs a different amount of resources (w_i) to be built and the minimized function is then $\sum_{j \in J} w_j x_j$, where $x_j = 1$ if the fire station j is chosen, $x_j = 0$ otherwise.

Definition 4.9.2 (Weighted set cover problem)**.** Given a set U (called the universe) and collection S_i of n sets such that $\bigcup_{i=1}^{n} S_i = U$ and weights $w_i \in \mathbb{R}$ corresponding to each S_i, find set J where $\bigcup_{j \in J} S_j = U$, such that $\sum_{j \in J} w_j x_j$ is minimal. Let $x_i = 1$ if S_i is chosen as part of the solution (if $i \in J$) and $x_i = 0$ otherwise. The Integer Programming (IP) formulation is

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} w_i x_i, \\ \text{subject to} \quad & \sum_{i:e \in S_i} x_i \geq 1, \qquad \forall e \in U, \\ & x_i \in \{0,1\}, \quad \forall i \in \{1,\ldots,n\}. \end{aligned}$$

Chapter 5

Proposed methodology

The problem formulated in (3.1), is essentially Coverage Path Planning (CPP). In all generality, the CPP is hard to solve. By [14] common approach for tackling this problem is to divide this problem, into two subproblems. Finding the optimal camera configurations C^* and finding the shortest path between them. The first problem is an Art Gallery Problem (AGP) or Optimal Camera Placement Problem (OCPP). The second problem is TSP. In our method, we will use the same approach. Our method considers the following data as input

- the point cloud of the environment, obtained by prior scanning with LiDAR,
- an OBB, which specifies the area in the point cloud, that is to be photographed,
- the UAV camera sensor intrinsics,
- the overlap ratios ρ_x and ρ_y of the images taken,
- the distance from obstacles ϵ, which is to be considered as prohibited regions,
- the size of voxels used for voxelization of the environment.

The proposed approach can be divided into 4 stages. Extraction of the plane of interest, cellular decomposition of the plane of interest, determination of the camera configurations C^* and UAV path construction.

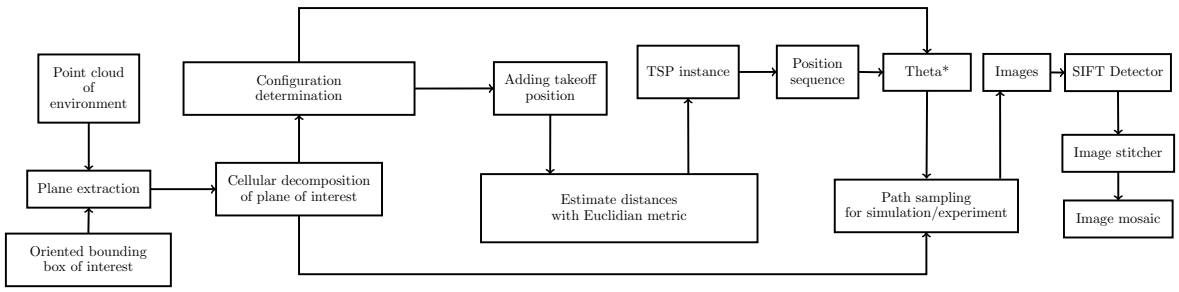


Figure 5.1: A flowchart of the solution pipeline

Plane extraction (as in extracting plane equation) is applied to separate relevant points (inliers), based on which the plane of interest is estimated. The plane equation is crucial for the determination of satisfaction of conditions $(\mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4)$ defined in Chapt. 3.

Cellular decomposition is the process of breaking a plane into n regions S_i, such that $\bigcup_{i=1}^{n} S_i \approx U$, where U is the plane of interest. We will denote an optimal decomposition, where P_6 - P_8 constraints are satisfied, as S^*.

To determine the C^* set (the configurations corresponding to the S^*), we are searching for camera placements, for each corresponding decomposed cell (camera view), such that P_1-P_5 are satisfied and $\mathrm{f}(c^*)$ where $c^* \in C^*$ is minimal.

To find the shortest tour, a metric TSP instance has to be constructed with distance metric $\mathrm{d}(a, b)$, which returns the length of the shortest collision-free path between a and b.

5.1 Extraction of the plane of interest

The plane of interest is not explicitly specified. So we need to extract the plane from the point cloud provided. Using the bounding box for filtering the point cloud, we obtain a subset of points in which the plane of interest lies. However, the points can be noisy. To robustly extract a plane from noisy data, we use the RANSAC algorithm. The RANSAC algorithm will give us a set of inlier points, outlier points and estimated plane parameters. To further estimate the plane, we use the least squares method on the inlier set (shifted by centroid) to solve the plane equation

$$\begin{bmatrix} x_1 & y_1 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}. \tag{5.1}$$

5.2 Cellular decomposition of the plane of interest

The inlier points, do not need to be necessarily distributed in rectangular shapes, which would allow for easy decomposition. The assumption is, that the shape can be any general polygon with holes[1]. The problem of choosing a decomposition of the plane of interest to n rectangles S_i (the views of the camera), such that $\bigcup_{i=1}^{n} S_i \approx U$, of different sizes, where P_6-P_8 holds (and are integer linear constraints), and the number of such rectangles is minimal, could be specified as the SCP. However, to obtain the optimal decomposition of camera views (rectangles) S_i^*, we would have to formulate this problem as geometric set cover[2] since any discretization can result in a potentially missed optimal solution. However, the geometric set cover is a very hard problem and the complexity of this formulation largely outweighs the benefit of the optimal solution in our method (if the exact optimal solution were to be found). A simpler approach is to discretize the problem into regular SCP (as mentioned). The size of the SCP instance depends on how sparsely is the state space discretized, which directly influences the computation time. Since there is additional complexity in formulating the conditions P_6-P_8 as integer linear constraints, we have decided to relax this problem, by using classic exact cellular decomposition, such that the constraints P_6 and P_8 will be implicitly satisfied. Since the resulting set will be no longer optimal, we denote it S' instead, with corresponding best camera configurations C'. However, the disadvantage of such approach is that such relaxation may result in in no camera spatial configurations, which will be able to satisfy all the remaining constraints in some instances of cluttered environments.

5.3 Determination of the camera spacial configurations

The camera spacial configurations C'_i, in which a corresponding cell S'_i is captured, can be many. In order to minimize f(·) in (3.1), one could use formal approaches like discrete or continuous mathematical programming. If P_1-P_8 were to be formulated as integer linear constraints, and f(·) would be used for determining weights in WSCP, there would be an implicit relationship between the camera configuration C_i and resulting camera view S_i. This

[1] A hole is a non-intersecting polygon inside another polygon.

[2] Similar problem to Set Cover, but instead of possible choices from a collection of sets S_i, we only have possible geometric shapes, in which we have to enclose all elements of the universe U.

would allow for optimizing both the number and the quality of pictures needed to be taken at the same time. However, the main problem with the application of mathematical programming is the complexity of formulating the constraints.

We propose to construct a set of points in front of each $S'_i \in S'$ and then filter them so that $\bigwedge \mathrm{P}_i$ for $i \in \{1, 2, 3, 4, 5\}$ is true. Since P_6 - P_8 are ensured from the cellular decomposition, all the candidates now satisfy all the constraints. Then if evaluated according to their quality $\mathrm{f}(\cdot)$ and sorted in ascending order. Then the first is the best candidate C'_i for the corresponding S'_i. For the $\mathrm{f}(\cdot)$ function, we have used a function composed of two other functions. The distance penalty function $\mathrm{d}(\cdot)$ is defined as

$$\mathrm{d}(\mathbf{x}) = |p - \mathrm{distance}(\mathbf{x}, \mathbf{g})|, \tag{5.2}$$

where $\mathbf{x} \in \mathbb{R}^3$ is a candidate position of the camera, p is the desired distance from $\mathbf{g} \in \mathbb{R}^3$, which is the centroid of corresponding cell S'_i obtained by cellular decomposition of the plane of interest. The angle penalty function is given by

$$\mathrm{a}(\mathbf{y}) = \frac{|\mathbf{n} \cdot \mathbf{y}|}{|\mathbf{n}|\,|\mathbf{y}|}. \tag{5.3}$$

where $\mathbf{y}$ is vector from the centroid $\mathbf{g}$ of the corresponding cell in decomposition to position $\mathbf{x}$ and $\mathbf{n}$ is normal of the plane of interest. The cost function $\mathrm{f}(\cdot)$ is then defined as

$$\mathrm{f}(\mathbf{x}) = \mathrm{d}(\mathbf{x}) - \mathrm{a}(\mathbf{x}). \tag{5.4}$$

5.4 The shortest UAV path construction

Given the set of camera spacial configurations C'' obtained as described in the previous section, the problem of finding a path connecting these configurations can be formulated as symmetric TSP[3] or metric TSP[4] (Def. 4.8.1). A TSP instance is represented by a distance matrix. However, in order to construct a distance matrix that would truly represent the distance, that the UAV would have to fly, we would have to weigh the paths (edges) connecting the camera positions by the length of the shortest non-colliding path. The shortest path can be found by pathfinding algorithms, such as (adjusted) Theta*[5]. Since the state space is potentially large, we generate the graph on the fly by adding nodes and edges of neighboring voxels (generally 26, since a voxel has 26 neighbors in 3D). The property of resulting paths, not colliding[6] with the environment is ensured by not expanding nodes, which would result in such proximity. The check for proximity is done with KD-Trees [35]. However, the number of paths, we would have to find is $\frac{n(n-1)}{2}$ paths. The pathfinding algorithms would need to explore potentially large (almost uniform) graphs since the number of nodes obtained by the expansion of one voxel is 26. In order to reduce this complexity, we can construct an approximate distance matrix by considering only the Euclidean distance between two nodes. This gives us an approximate solution of the real-world TSP, which is then used to find collision-free paths with Theta*.

[3] In symmetric TSP, the distance between cities A and B is always the same, as the distance between B and A.

[4] In metric TSP, the distance between cities is (often) Euclidean distance and always satisfies the triangle inequality. It is a special case of graph TSP.

[5] The Theta* adjustment is by changing the clear line of sight condition, by collision-free flythrough.

[6] According to P_1 the UAV should not be closer than ϵ to any obstacle at any given time.

Chapter 6

Simulation results

For the simulation, we have chosen the MRS System [5]. For the image collection phase, we have employed a quadrotor of type T650 equipped with a front-facing RGB camera. As a simulation environment, we have chosen the point cloud of the Church of Saint Maurice in Olomouc. In the point cloud, we have chosen two areas of interest and for each one we performed a simulation.

1. A stained glass window The Sacred Heart of Jesus,
2. the pair of stained glass window Inspiration of St Francis de Sales and St Francis de Sales hands over the Rule of Order.

6.1 Simulation 1: A stained glass window

For the first simulation, we have chosen a stained glass depicting The Sacred Heart of Jesus (see Fig. 6.1). We have used parameters $\mathrm{FOV_x} = \frac{\pi}{6}$, $p = 5$ (see Table 6.1). The decomposition has resulted in 20 cells and a feasible camera configuration was found for all of them. However, the images captured (see Fig. 6.7) from the simulation were not successfully stitched into a mosaic (see Fig. 6.3). It is probably due to one row of images not being able to stitch and thus disconnecting the rest.

6.2 Simulation 2: Pair of stained glass windows

For the second simulation, we have used a pair of stained glass windows (the windows of Inspiration of St Francis de Sales and St Francis de Sales hands over the Rule of Order, see Fig. 6.4). We have used parameters $\mathrm{FOV_x} = \frac{\pi}{6}$, $p = 6$ (see Table 6.1). The plane decomposition for this simulation resulted in 27 cells (see Fig. 6.5) and a feasible camera configuration was found for all of them. The images taken by the camera (see Fig. 6.7) are very colorful and have high contrast and many SIFT features have been found. The images were successfully stitched into a mosaic. However, the result is not a quality image mosaic. This is likely because we have used an autonomous stitcher from OpenCV [25] and stitching with a more controlled method is required.

Table 6.1: Simulation parameters and duration

Simulation	FOVx (rad)	p	Num. of cells	Position calculation (s)	Path calculation (s)
Simulation 1	$\frac{\pi}{6}$	5	20	9.2	6.5
Simulation 2	$\frac{\pi}{6}$	6	27	7.5	62.2

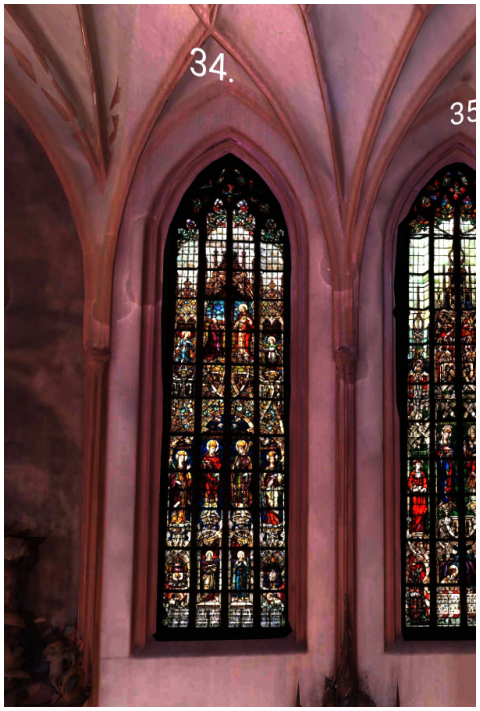
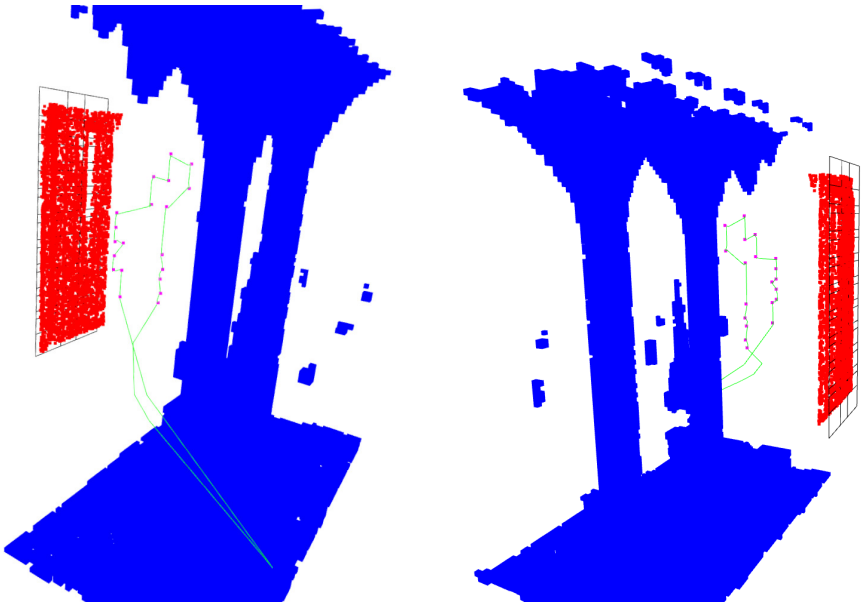


Figure 6.1: Views of a stained glass window with a motive of The Sacred Heart of Jesus, in the MRS group 3D model web viewer



(a) left side view of visualization of simulation 1 (b) right side view of visualization of simulation 1

Figure 6.2: Visualizations of camera configurations resulted from simulation 1 in Open3D [11]. In this image, the red points are points separated by OBB of interest, the blue points are obstacles, the purple points are camera configurations computed by simulation, the black lines are cells of exact cellular decomposition and the green lines represent the path between the points and the starting point.

Figure 6.3: Image mosaics from captured images of simulation 1. There are not present all images captured. It is probably due to one row of images not being able to stitch to the others and thus disconnecting the rest.

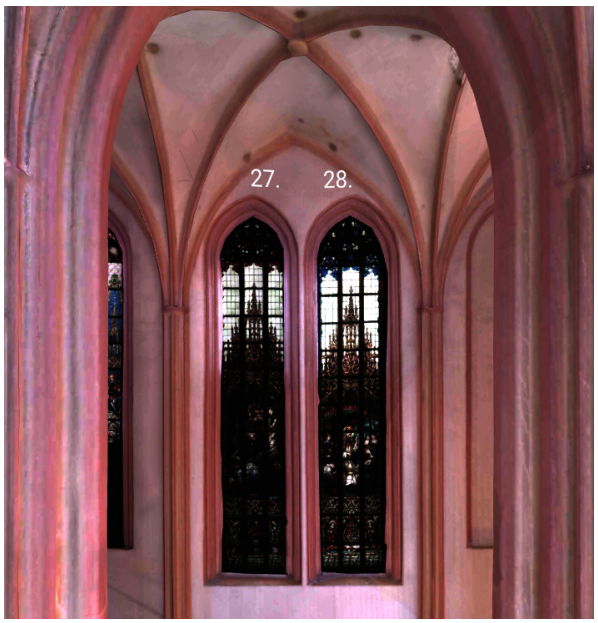
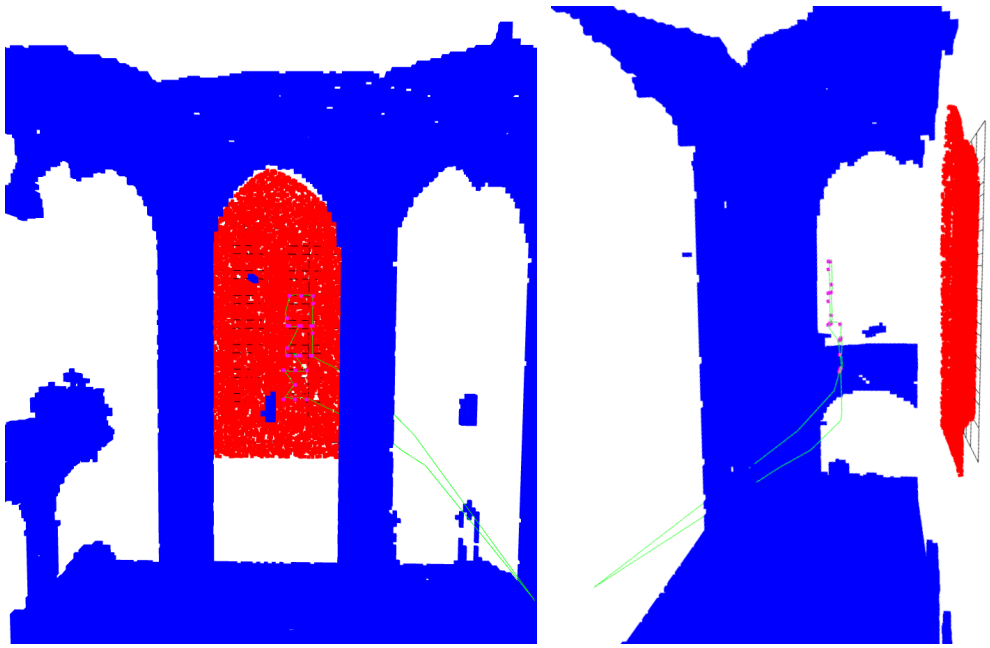


Figure 6.4: Views of a pair of stained glass windows in the church of Saint Maurice in Olomouc, in the MRS group 3D model web viewer



(a) front view of visualization of simulation 1 (b) right side view of visualization of simulation 1

Figure 6.5: A visualization of camera configurations resulted from simulation 2 in Open3D [11]. In this image, the red points are points separated by OBB of interest, the blue points are obstacles, the purple points are camera configurations computed by simulation, the black lines are cells of exact cellular decomposition and the green lines represent the path between the points and the starting point.

Figure 6.6: Image mosaics from captured images of simulation 2. The result is not a quality image mosaic. This is likely because we have used an autonomous stitcher from OpenCV [25] and stitching with a more controlled method is required.

Figure 6.7: Captured images of simulation 2

Chapter 7

Real-world experiment verification

For a real-world experiment, we have chosen a stained glass window depicting The Sacred Heart of Jesus (see Fig. 6.1) and pair of stained glass windows (the windows of Inspiration of St Francis de Sales and St Francis de Sales hands over the Rule of Order, see Fig. 6.4) in church of Saint Maurice in Olomouc. The output of our proposed algorithm is ready for use in real-world experiments. The output of our method is a CSV file that contains the sequence of relative 3D positions (relative to the origin of processed point cloud) and yaw values, which describe the non-colliding path (see Fig. 7.1 and Fig. 7.2) for the UAV to fly through the found camera configurations C''. At each camera configuration, there are stop requests included by padding values for the UAV to stabilize and take a picture. The output is meant for the MRS UAV system [5], coupled with a system designed for safe deployment of UAVs in autonomous documentation of historical buildings [1], [6]–[9]. However, the testing site at the church of Saint Maurice in Olomouc is currently unavailable for us, so we can not perform a real-world experiment.

(a) front view of visualization of simulation 1 (b) side view of visualization of simulation 1

Figure 7.1: Visualization of path resulting from simulation 1 in RViz

(a) right side view of visualization of simulation 2 (b) left side view of visualization of simulation 2

Figure 7.2: Visualization of path resulting from simulation 2 in RViz

Chapter 8

Future work

We have proposed an approach for the solution of the defined problem in real-world scenarios. However, within the design of the method, we have identified several possible improvements, that we would like to target in our future work. The cost function defined in (5.4) can be improved to fully reflect the differences in the real object resolution/pixel density (pixels per m^2). The cost function should also reflect properly the loss in pixel density resulting from taking an image under an angle.

We have proposed the exact decomposition for our method. However, we mentioned, that formulating the proposed conditions (if possible) on a camera configuration as integer linear constraints of WSCP would allow for an implicit relationship between the feasibility of a solution and the cost of an optimal solution. However, the formulation of such conditions is out of scope of this thesis.

Local optimization techniques or iterative techniques could improve the plane decomposition in scenarios, where some of the cells of the exact decomposition are unfeasible. This would result in a more robust algorithm that would disqualify fewer cells, which would lead to larger coverage by the proposed method.

In Chapt. 3, we have specified, that there should be some minimal geometrical overlap between images. However, as we have seen in the results from simulations (see Sec. 6.1 and Sec. 6.2), this is not always enough to make a quality stitched image from the collection. To provide a better guarantee, that the resulting images are stitchable, cellular decomposition that takes into consideration the image features as a criterion in the decomposition would be advantageous.

Chapter 9

Conclusion

In this work, we have proposed a method for the generation of camera configurations for autonomous image capture using autonomous unmanned vehicles for monitoring the interiors of historical objects. We have identified the underlying problem ad coverage path planning problem, recognized it as a composition of the optimal camera placement (art gallery problem) and traveling salesman problem and formulated it as a constrained optimization problem. We formulated a 4-stage methodology for solving the problem and implemented it in Python. We have used the Gazebo simulator and the MRS system to verify the algorithm. The output of our proposed algorithm is ready for use in a real-world experiment. However, due to the unavailability of the church of Saint Maurice for the realization of autonomous UAV flight, the experiment could not be performed.

References

[1] J. Deckerová, J. Faigl, and V. Krátký, "Traveling salesman problem with neighborhoods on a sphere in reflectance transformation imaging scenarios," *Expert Systems with Applications*, vol. 198, p. 116 814, 2022.

[2] J. K. Gómez-Reyes, J. P. Benítez-Rangel, L. A. Morales-Hernández, E. Resendiz-Ochoa, and K. A. Camarillo-Gomez, "Image mosaicing applied on uavs survey," *Applied Sciences*, vol. 12, p. 2729, 2022. DOI: 10.3390/app12052729.

[3] I. Jang, S. Lim, and H. Jeon, "Change detection in unmanned aerial vehicle images for industrial infrastructure rooftop monitoring," in *2022 19th International Conference on Ubiquitous Robots (UR)*, 2022, pp. 274–279.

[4] Wikipedia contributors, *The night watch — Wikipedia, the free encyclopedia*, [Online; accessed 4-January-2023], 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=The_Night_Watch&oldid=1112148288.

[5] T. Baca, M. Petrlik, M. Vrba, *et al.*, "The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, pp. 1–28, 2021.

[6] V. Krátký, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, "Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021.

[7] V. Krátký, P. Petráček, T. Nascimento, *et al.*, "Safe documentation of historical monuments by an autonomous unmanned aerial vehicle," *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, pp. 738/1–16, 2021.

[8] V. Krátký, P. Petráček, V. Spurný, and M. Saska, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020.

[9] P. Petráček, V. Krátký, and M. Saska, "Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2078–2085, 2020.

[10] J. Kritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar, "On the optimal placement of cameras for surveillance and the underlying set cover problem," *Applied Soft Computing*, vol. 74, pp. 133–153, 2019.

[11] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[12] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015, pp. 111–117.

[13] E. Candigliota and F. Immordino, "Low altitude remote sensing by uav for monitoring and emergency management on historical heritage," in *ANIDIS Congress, Padova*, vol. 30, 2013.

[14] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[15] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4126–4133.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571.

[17] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked uavs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[18] K. Helsgaun, "General k-opt submoves for the lin–kernighan tsp heuristic," *Mathematical Programming Computation*, vol. 1, no. 2, pp. 119–163, 2009.

[19] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *Journal of Field Robotics*, vol. 26, no. 8, pp. 651–668, 2009.

[20] O. Chum and J. Matas, "Optimal randomized ransac," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1472–1482, 2008.

[21] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized tsp problem," *Progress in Natural Science*, vol. 18, no. 11, pp. 1417–1422, 2008.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, 2006, pp. 404–417.

[23] C. C. Applegate Bixby. "Concorde-03.12.19." (2003), [Online]. Available: `https://www.math.uwaterloo.ca/tsp/concorde.html` (visited on 05/01/2023).

[24] W. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 27–32.

[25] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[26] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[27] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*, 1998, pp. 203–209.

[28] M. Jünger, G. Reinelt, and G. Rinaldi, "Chapter 4 the traveling salesman problem," in *Network Models*, ser. Handbooks in Operations Research and Management Science, vol. 7, 1995, pp. 225–330.

[29] T. Shermer, "Recent results in art galleries (geometry)," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1384–1399, 1992.

[30] Z. L. Cao, Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic systems*, vol. 5, no. 2, pp. 87–102, 1988.

[31] C. Cowan and P. Kovesi, "Automatic sensor placement from vision task requirements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 407–416, 1988.

[32] C. C. Skiscim and B. L. Golden, "Optimization by simulated annealing: A preliminary computational study for the tsp," Institute of Electrical and Electronics Engineers (IEEE), Tech. Rep., 1983.

[33] S. M. Stigler, "Gauss and the Invention of Least Squares," *The Annals of Statistics*, vol. 9, no. 3, pp. 465 –474, 1981.

[34] C. H. Papadimitriou, "The euclidean travelling salesman problem is np-complete," *Theoretical Computer Science*, vol. 4, no. 3, pp. 237–244, 1977.

[35] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, 509–517, 1975.

[36] V Chvátal, "A combinatorial theorem in plane geometry," *Journal of Combinatorial Theory, Series B*, vol. 18, no. 1, pp. 39–41, 1975.

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

[38] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*. 1972, pp. 85–103.

[39] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees: Part ii," *Mathematical programming*, vol. 1, no. 1, pp. 6–25, 1971.

Appendix A

Figure 1: Captured images of simulation 1

Appendix B

List of attachments

- `source_code.zip` Source code for the implementation.