Graz University of Technology

Institut for Computer Graphics and Vision

PhD Thesis

# Interactive Structure-from-Motion

**Christof Hoppe**

Graz, Austria, May 2014

*Thesis supervisors*

Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof

Univ.-Prof. Dipl.-Ing. Dr.techn. Konrad Schindler

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

# EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

# STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

# Abstract

Structure-from-Motion, a technique to reconstruct 3D information from multiple overlapping 2D images, has reached maturity. Since the processing of thousands of high-resolution unordered images is quite efficient, it is routinely applied for obtaining reconstructions from photo community collections but also for applications like architectural reconstruction, geological surveying and scene documentation.

This thesis focuses on Structure-from-Motion for applications where images can be deliberately acquired for the reconstruction process. Such applications typically have requirements on the accuracy and completeness of the reconstruction. Since these parameters largely depend on the acquired input images, the image acquisition becomes an integral part of the reconstruction process. This provides the opportunity to couple the reconstruction and the image acquisition to guarantee the user's requirements. Therefore, two methods are proposed in this thesis that exploit this opportunity in different ways.

The first approach couples the image acquisition and the reconstruction process tightly by a new closed-loop Structure-from-Motion method that incrementally performs the reconstruction in real-time. The novel surface extraction method that operates on sparse triangulated feature points in real-time and in a fully incremental manner, allows to visualize important reconstruction parameters like the redundancy and the resolution of the reconstruction. This instant feedback on-site allows the user to continuously asses the reconstruction's quality already during the image acquisition and to recognize problems at an early stage. Since this feedback is provided in real-time, the user can adapt the image acquisition to avoid these problems. As a consequence, the presented interactive Structure-from-Motion method greatly improves the reliability of image-based reconstruction in practical applications when the user is in the loop.

The second proposed method, that exploits the controllability of the image acquisition, is a view planning approach that is specifically designed for large-scale, close-range reconstructions from wide-baseline images. The novel method takes the user's requirements on the accuracy, resolution and completeness of the reconstruction into account but also the

requirements of state-of-the-art Structure-from-Motion pipelines on the image data, which are redundancy and a certain spatial distribution of the images to allow wide-baseline feature matching. The method selects a small subset of a large number of potential view points such that the most important parameters of a reconstruction are fulfilled. Experiments show that the proposed view planning approach provides image datasets that are suitable for reliable processing by todays Structure-from-Motion pipelines.

To summarize, the interactive Structure-from-Motion and the novel view planning approach increase the reliability of image-based reconstruction from high-resolution wide-baseline images and therefore, opens new applications for this reconstruction method.

# Kurzfassung

Die 3D Rekonstruktion aus mehreren redundanten Bildern, auch Structure-from-Motion genannt, hat in den letzten Jahren beträchtliche Fortschritte gemacht und somit sind heutige Verfahren in der Lage tausende von ungeordneten und hochaufgelösten Bildern zu verarbeiten. Typische Anwendungsbereiche von Structure-from-Motion sind z.B. die Rekonstruktion von Sehenswürdigkeiten aus Internet-Bilddatenbanken, Archäologie, Geologie und Vermessungskunde.

Diese Dissertation konzentriert sich auf die bildbasierte Rekonstruktion für Anwendungen, in denen Bilder speziell für den 3D Rekonstruktionsprozess aufgenommen werden. Solche Anwendungen stellen oft bestimmte Anforderungen wie z.B. eine hohe Genauigkeit oder die Vollständigkeit, an die Rekonstruktion. Da diese Parameter sehr stark von den verwendeten Bildern abhängig sind, muss die Bildaufnahme in den Rekonstruktionsprozess eingebunden werden damit die finale Rekonstruktion die Erwartungen erfüllen kann. Daher werden in dieser Arbeit zwei neue Verfahren vorgestellt, die die 3D Rekonstruktion und die Bildaufnahme eng miteinander verbinden.

Das erste entwickelte Verfahren verbindet die Bildaufnahme und die Rekonstruktion indem die klassische, sequentielle Rekonstruktionsmethode durch ein neues inkrementelles und echtzeitfähiges Verfahren ersetzt wird. In Verbindung mit einem neuen Algorithmus zur Extraktion von Oberflächen aus wenigen 3D Punkten, das ebenfalls in Echtzeit und inkrementell arbeitet, ermöglicht dieses interaktive Verfahren die Rekonstruktion bereits während der Bildaufnahme durchzuführen. Durch die ständige Visualisierung von wichtigen Rekonstruktionsparametern, wie der Redundanz und der Rekonstruktionsauflösung, können Probleme frühzeitig erkannt werden und der Benutzer kann diese umgehen, indem er seine Bildaufnahmestrategie anpasst. Diese Methode macht die bildbasierte Rekonstruktion zuverlässiger und erweitert damit den Anwendungsbereich dieser Rekonstruktionsmethode.

Die zweite vorgestellte Methode berücksichtigt die Möglichkeit, die Aufnahmeorte der Bilder für die Rekonstruktion frei zu wählen, indem sie ein Kameranetzwerk berechnet, das wichtige Parameter der bildbasierten 3D Rekonstruktion berücksichtigt. Im Gegensatz zu vielen existierenden Verfahren wurde diese Methode speziell für hochaufglöste Bilder entwickelt, bei denen korrespondierende Punkte zwischen den Bildern ohne zeitliches Vorwissen gefunden werden müssen. Das Verfahren bezieht daher neben Anforderungen des Nutzers an Genauigkeit, Auflösung und Vollständigkeit auch die Anforderungen des Rekonstruktionsverfahren bezüglich der relativen räumlichen Verteilung der Kamerapositionen in die Berechnung ein. Die mit diesem neuen Verfahren berechneten Kamerapositionen erlauben eine zuverlässige 3D Rekonstruktion mittels der heute gängigen Methoden.

Sowohl die interaktive, echtzeitfähige Rekonstruktionsmethode als auch die Planung eines Kameranetzwerkes machen die bildbasierte Rekonstruktion zuverlässiger und damit attraktiver für neue Anwendungsbereiche.

# Acknowledgments

First and foremost I want to thank Prof. Horst Bischof for giving me the opportunity and the support to do my PhD at the ICG. He convinced me that doing a PhD is a very interesting challenge where you learn a lot.

Furthermore, I want to thank Manfred Klopschitz, Arnold Irschara, Matthias Ruether, Stefan Kluckner and Katrin Santner, a.k.a Katrin Pirker, for introducing me in the world of 3D reconstructions from images and who helped me to develop new ideas. Manfred pushed me several times to find a better solution for a problem. Special thanks go to Matthias for discussing problems about geometric relations many times and for proofreading parts of this thesis.

I'm very thankful for the help of Michael Donoser who supported my work and who taught me in scientific writing. Furthermore, I learned a lot from his distant view to my research. Furthermore, his continuous commitment for the reading group, which is a source for new ideas and important for extending the view on other topics on computer, is admiring. Thank you very much also for proofreading the thesis!

I also want to thank Christian Reinbacher and David Ferstl for starting each workday with a big cup of great coffee. The collaboration with my colleagues from the Aerial Vision Group and with my students helped me a lot with their input and discussions. All people at the institute make the ICG to a place where it is a pleasure to work and to develop new ideas but also to enjoy the social components of work.

Working on a PhD thesis brings much joy but also sometimes frustrations. For both reasons, everyone needs someone who shares these feelings with you. My spouse Hanna has always been there for me in all situations. I'd like to thank her for supporting me at the low points but also for sharing my elations about new findings in abstract computer vision problems.

There are also some guys I'd like to mention because they brought me to Graz: Theresa Rienmueller, Michael Reip, Christof Rath, David Monichi, Mate Wolfram, Christoph Zehentner and Stefan Gspandl. Before I met them at a Robocup competition, I did not even knew that Graz is located in Austria.

Finally, I'm deeply grateful for the support of my family, my parents Franz and Irmgard who always believe in me and supported my decisions, my brothers Matthias and Andreas and my sister Lisa for all the things I learned from them and all the time they took care of me.

This work was supported by the Austrian Research Promotion Agency (FFG) FIT-IT project CONSTRUCT (830035) and the FFG project UAV Vermessung (3770439).

# Contents

# List of Figures

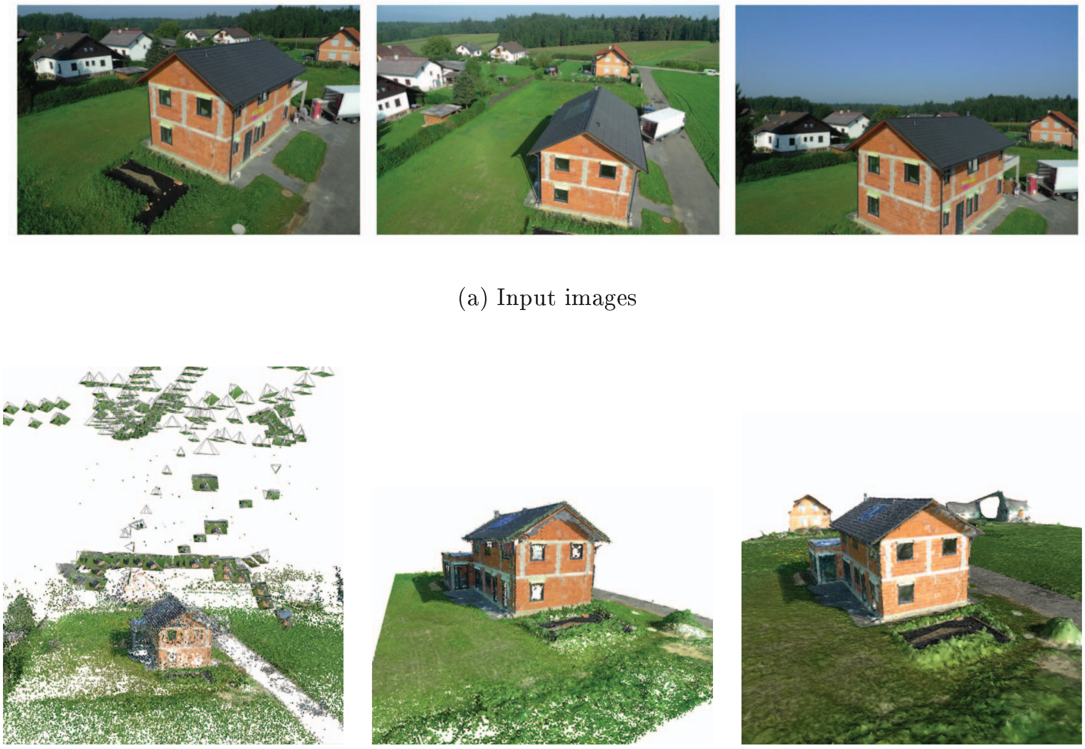
# Chapter 1

# Introduction

Our life takes place in a 4-dimensional world: three spatial and one time-related dimension. Nowadays, we are able to capture a two-dimensional projection of our physical world and the time dimension easily with photographs or videos. However, capturing all three spatial dimension in a dense manner is a hard task and is an unsolved issue. For special applications like the 3D capturing of indoor environments or the 3D reconstruction of microscopic objects, specialized hardware devices are available. But right now, there exists no universal device for capturing our world in all three spatial dimensions.

As a human, we mainly derive 3D information visually from two basic principles: stereo-vision and monocular cues. The strongest cue for depth perception is the disparity information that is derived from our two eyes. But also depth cues that can be derived from a single view like motion, shading and occlusions are important for our depth perception.

These biological principles triggered fundamental research in computer vision and resulted in sophisticated methods for reconstructing depth information from stereo-images as well as from moving monocular cameras. The latter technique is well-known in computer vision as Structure-from-Motion (SfM) or Structure-and-Motion. The reconstruction from stereo-images requires a specialized hardware setup whereas SfM can be calculated from a set of images acquired by widely used consumer-grade cameras. The goal of SfM is to recover camera poses as well as the 3D scene structure given only a set of 2D images that are partially redundant. Figure 1.1 shows the typical input and output of a SfM processing pipeline.



(a) Input images

(b) Sparse reconstruction

(c) Dense reconstruction

(d) Meshed result

Figure 1.1: Structure-from-Motion example. (a) A set of overlapping images is used to determine the camera positions and a sparse scene representation as shown in (b). Given this information, typically the point cloud is (c) densified and finally (d) a triangular surface mesh is extracted.

Fundamental research on the geometric relation between 2D images such as the calculation of the relative motion has been already done at the beginning of the 20th century and became popular with the development of computational hardware. This research phase is summarized in the book of Hartley and Zisserman [31]. The enormous computational complexity, the absence of robust image features and missing robust image registration methods prevented SfM of being practically usable for large-scale high-resolution scene reconstruction in the past. With the development of robust features like the Scale Invariant Feature Transform (SIFT) [62] and the continuously growing computational power, SfM became again very popular in the 2000's where pipelines have been presented that were able to recover the scene structure of thousands of unordered images completely automatically [2]. In this scenes, images acquired by tourists are used to reconstruct important sights. These approaches benefit from the ultra-high redundancy of the set of input images.

This work triggered research on the scalability and reliability of SfM for images that are not deliberately acquired for the purpose of 3D reconstruction. However, many applications realized so far, are based on images that are especially captured for the reconstruction process. Such applications are for example scene documentation, geological surveying or archaeological documentation. Typically, these applications have specific requirements on the reconstruction's quality. Two major requirements are *completeness* and *accuracy*.

Both requirements, completeness and accuracy, are primarily dependent on the set of input images and therefore, the image acquisition is crucial for the whole reconstruction process. We observed that it is difficult for an expert user and even more for a non-expert user to obtain an image set that fulfills the user's needs. Therefore, many reconstructions completely fail or they do not fit to the user's expectations. Therefore, we investigate in this thesis methods that support the user in the image acquisition process to obtain an image dataset that leads reliably to an accurate and complete 3D reconstruction using SfM.

In the rest of this chapter, we first motivate our work by showing examples for typical applications that are realized with SfM. Secondly, we analyze the challenges that have to be solved to realize this applications.

## 1.1 The Future of SfM and its Challenges

These days SfM pipelines are able to automatically reconstruct a 3D scene of hundreds of images within reasonable time. Since these pipelines are designed to handle large-scale unordered image datasets with very high redundancy, they often do not meet the demands of application scenarios where images acquired deliberately for the reconstruction process. In order to analyze the differences, we first give examples for the contemplated applications. Secondly, we also motivate our work by a new class of image acquisition devices, namely Micro Aerial Vehicles (MAVs) that expand the range of SfM applications. Finally, the challenges of the new devices in combination with the aforementioned applications lead to the main research question that is addressed in this thesis.

### 1.1.1 Applications

The capability to capture the world in 3D using only a set of redundant 2D images is beneficial for many applications. In contrast to special 3D acquisition devices like laser scanners, the acquisition of images with a standard consumer-grade camera is very cheap. Todays cameras deliver high-quality images and are easy to handle such that even non-expert users have the possibility to build high-resolution 3D reconstructions. In the following, we give a showcase of potential applications for SfM.

One potential application for automated SfM is the documentation of scenes in 3D. Typical domains are the documentation of archaeological excavations [78, 98], traffic accidents or construction site monitoring [25, 49]. In [110], we describe a system for construction site monitoring in 4D. Here, images of the construction site are acquired regularly which are subsequently used for 3D reconstruction. Figure 1.2 shows a typical result of two reconstructions obtained at different points in time. The advantage of a full 3D reconstruction over a documentation using only images is twofold: first, the scene itself is available in 3D which can be used for sophisticated visualization techniques like Augmented Reality or as input for further processing steps like the detection of changes over time. And second, the spatial position of the camera poses is available. This facilitates the navigation within thousands of images and it also gives the spatial context of the image.

Another application where SfM is used is geological surveying where the task is to create a map of a geological formation. The size of the geological formation ranges from a few meters to several kilometers when thinking of open pit mining. Here, SfM can

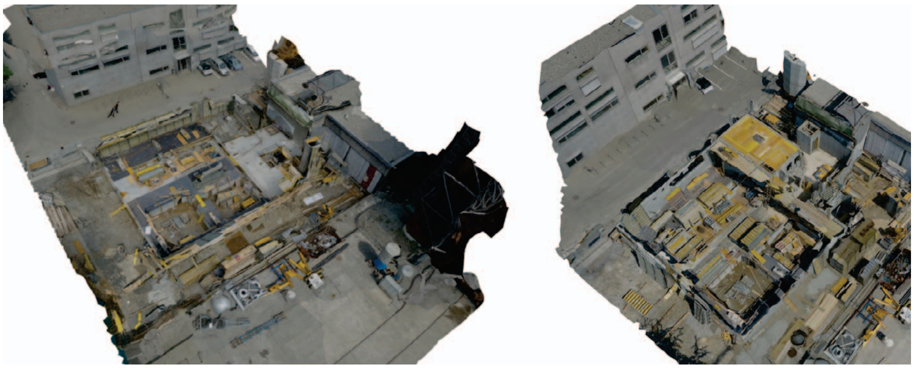


Figure 1.2: Example of the construction site monitoring application. Two reconstructions at different points in time are used to calculate the progress.

support classic geological surveying by delivering a detailed 3D reconstruction and also visual information in the form of images. For many task-specific analyses both types of information are important.

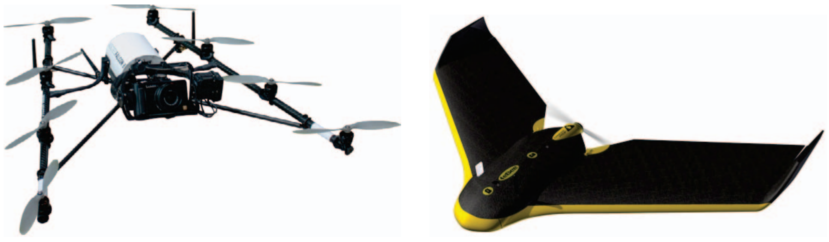
A further application domain is the mapping of constructions. Today often laser scanners are used to capture the 3D information of buildings. The resulting point clouds are then used to determine distance measures like the size of a window. Again, SfM is beneficial because it delivers 3D information as well as the visual information which is often important for subsequent processing.

For these applications typically images are acquired especially for the reconstruction purpose. Furthermore, the user has specific requirements on the resulting reconstruction like completeness and accuracy. In this context completeness means that the reconstruction contains all scene parts that are relevant for the specific task. The term accuracy is also task dependent. For some applications like the visualization of scenes it is sufficient that the reconstruction preserves basic scene properties like planarity or the correct ratio between different lengths measured in the reconstruction. For other tasks like the documentation of archaeological excavations, it is required that the reconstruction is metrically correct and that scene coordinates can be determined with an accuracy that is higher than a few millimeters.

The common ground of all applications is that they are large-scale and they require a high resolution reconstruction. Furthermore, many of the outlined applications are not feasible with ground-based images. Therefore, a new class of image acquisition devices, Micro Aerial Vehicles (MAVs) which are small flying devices equipped with a high-resolution camera, opens up the possibility to realize large-scale but close-range reconstructions using SfM.

### 1.1.2 New Devices

Micro Aerial Vehicles (MAVs) are small and lightweight fixed-wing planes or multi-roter copters equipped with a high-resolution still image camera. Two MAVs used within this thesis are shown in Figure 1.3. They either act autonomously or they are controlled by a user remotely. Since they are stabilized by sophisticated control algorithms, even non-expert users are able to operate these MAVs. The typical flight time varies between 20 and 40 minutes and the operation height is between 3 m and 150 m above the ground



(a) (b)

Figure 1.3: Two different Micro Aerial Vehicles. (a) Asctec Falcon 8 octo-copter equipped with a 16 Mpx still image camera. The flight time is around 20 minutes and the operation height varies between 3 m and 150 m above ground level. (b) The Ebee MAV manufactured by Sensefly is a fixed-wing MAV equipped with a 12 Mpx still image camera. After the definition of a flight plan, it operates up to 45 minutes autonomously. The camera is mounted such that nadir images are collected. Its operation height is between 70 m and 300 m.

level. Furthermore, multi-copter are very agile and often can approach each point in space. Since their safety border to objects is very low (2 m to 4 m), they can be operated even in densely populated areas and allow the collection of closeup images of vertical and horizontal structures. These capabilities bridge the gap between ground-based image acquisition and high-altitude flying planes. Hence, the combination of powerful SfM pipelines and images acquired from these small flying devices enable the realization of the previously mentioned applications. Figure 1.4 shows sample images acquired by a fixed-wing MAV and a multi-rotor copter and demonstrates the new challenges that especially arise with multi-copters. Their capability to approach each point in space increases the number of possible viewpoints as shown in Figure 1.4(a) and 1.4(b). This variability cannot be reached by using only ground-based images and therefore is a new challenge for SfM. Furthermore, to efficiently use the limited flight time, the user typically explores a very limited part of all possible view points and therefore the resulting image set is typically more sparse than ground-based datasets.

### 1.1.3 New Challenges

State-of-the-art SfM pipelines like [89] are designed for processing unordered but highly redundant image datasets. Furthermore, they assume that the image acquisition process is uncontrollable and therefore they have to stick to the provided data. Hence, quality

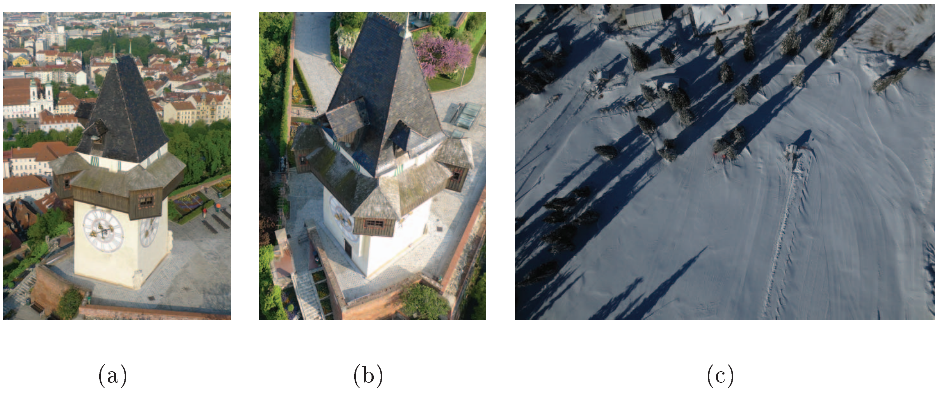


Figure 1.4: (a),(b) Sample images acquired by a octo-copter. (c) Image acquired by a fixed-wing MAV. Due to its flexibility, the octo-copter acquires images from very unusual view points. The sample also shows one problem that comes up with the flexibility: images can have very diverse viewing angles and therefore can be difficult to process by SfM.

parameters of the reconstruction like accuracy and completeness that largely depend on the image set cannot be influenced.

However, to realize such applications as outlined in Section 1.1.1, often quality parameters like a certain accuracy or completeness have to be guaranteed. Since these parameters largely depend on the input data, the image dataset must fulfill certain requirements so that the subsequent reconstruction process is able to obtain the required quality parameters. The consequence is that typically an arbitrary collection of images is not suited and the image acquisition process has to be carefully designed according to the requirements of the reconstruction. Therefore, the image acquisition becomes a working stage in the processing pipeline that often requires manual effort which is time- and cost-consuming. Hence, one possibility to be efficient is to acquire a small but well-suited set of images because this reduces the time on-site as well as the processing time. Therefore, these datasets often have quite different characteristics than datasets taken from the Internet. The most important difference is that the collected image sets are typically more sparsely distributed over the scene than the highly redundant datasets that are used to reconstruct sights from tourists photo.

But the reliability and accuracy of todays SfM pipelines is often reached by the highly redundant input data. For example, the triangulation accuracy of a 3D point is related to the number of images that observe a specific scene part. Furthermore, a reduced redundancy has a severe impact in reliability. Since the state-of-the-art in SfM relies on automatic correspondence estimation between image pairs which is prone to errors, high redundancy increases the probability that the full scene can be reconstructed even if some images cannot be used in the reconstruction process. Another issue that is related to redundancy but also to the spatial distribution of the input images over the scene is coverage. The coverage of a reconstruction is the property that all relevant parts are being reconstructed. Although this is an obligatory requirement for many applications, it is often not obvious how the cameras have to be spatially distributed to obtain full coverage. Figure 1.5(a) shows a typical failure case that is caused by the insufficient distribution of camera view points within the scene. Here, a wall of the building is completely missing because there are only a few input images that capture this part of the building. Another failure case that is caused by an insufficient distribution of the images over the scene is illustrated in Figure 1.5(b) and 1.5(c). Here, the SfM result is fragmented into two parts although the reconstructions are overlapping. The reason is that the viewing angles between some images are too large such that the pairwise camera orientation calculation partially fails.

An ad-hoc solution to these problems that comes into mind is to equally distribute the images over the input scene. A short numeric example shows why this is not feasible. Assuming we want to reconstruct a rather small volume of 50 m × 50 m × 50 m. If we

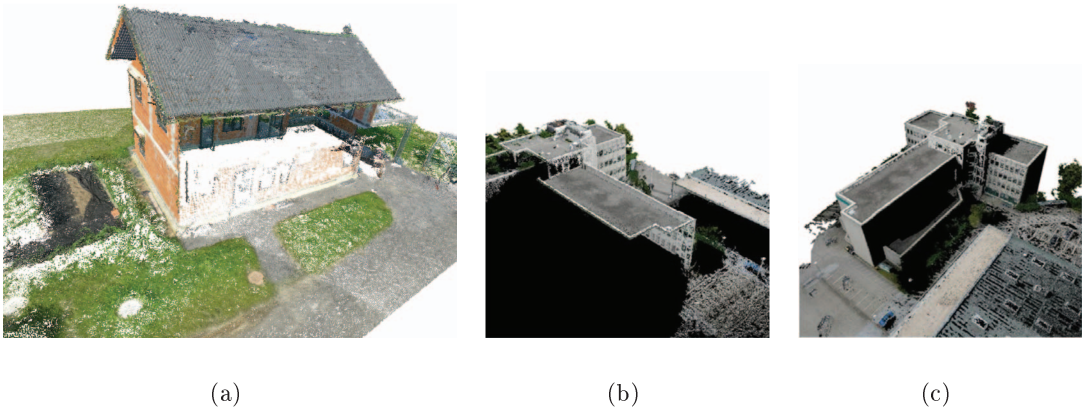

(a) (b) (c)

Figure 1.5: Typical failure cases of SfM. (a) Although the image dataset was acquired by an experienced SfM user, some parts of the facade could not be reconstructed because the spatial distribution of view points in the scene was not sufficient. (b),(c) The SfM result is fragmented into different parts although the reconstructions are overlapping.

want to place a camera each single meter this would result in 125,000 camera positions. When additionally considering the viewing angles and discretize the viewing direction into 20 degree steps, this results in about 40 Million camera positions. When assuming that a state-of-the-art consumer-grade camera that is typically mounted on an MAV is able to capture every second a high-resolution image, it is not feasible to acquire that amount of images. Furthermore, it is often also not required because most of the pictures are completely redundant or show irrelevant scene structures like the sky. Hence, these images only cause additional processing time without improving the overall reconstruction result.

Hence, a user typically acquires a set of images with much less redundancy whereupon the spatial distribution complies with the geometry of the object of interest. An expert in SfM often finds a valid view point configuration that leads to reasonable reconstruction results if the object of interest is not geometrically complex, e. g. a detached house or a single wall. But when thinking of geometrically complex objects like the nave of a baroque style church as shown in Figure 1.6, then even experts often fail. We observed that it is difficult for a user to remember after a few minutes the parts that have been already captured. Therefore, it often happens that parts are completely missing, images are not overlapping or that the viewing angle or the baseline between images is too large to allow feature matching.

Beside the geometric configuration of the view points, the texture of the scene also impacts the reconstruction result. If the scene is well textured as for example rocks or wood, many corresponding points can be matched across images and therefore the pairwise orientation of images is reliable. In contrast, if the scene consists of repetitive texture or large untextured regions, the pairwise orientation estimation might fail although the images are overlapping. For a user, it is often very difficult to estimate if a texture is well suited for feature matching. For example, the structure of plaster looks very repetitive for a human, but we found that it is perfectly suited for feature matching. On the other hand, images of vegetation are often very hard to match because of the self-similarity of the texture. Since the properties of the texture are a huge uncertainty factor in the reconstruction process, it is difficult to predict if the result meets the users expectations.

Therefore, acquiring an image dataset with reduced redundancy of a geometrically complex scene that meets all requirements is a challenging task. For a non-expert user this is even harder as the image set in Figure 1.7 shows. This image set has been acquired by a non-expert user who got a ten-minute instruction how images for SfM have to be collected. Although the complexity of the scene is rather low, the resulting set is not processable by state-of-the-art pipelines due too large view distortions between the images that is caused by the small amount of redundancy.

This experiment shows that the acquisition of an optimal image set with a reduced redundancy without any intermediate feedback is challenging for a non-expert user. But the property that the image acquisition can be controlled opens up the possibility to actively influence this important step. The potential to actively control the acquisition process has not been exploited extensively in SfM for getting more accurate and more reliable reconstructions.

Hence, in this thesis, we investigate how the fact that the image acquisition process can be controlled can be exploited to

- Speed-up the reconstruction process
- Increase reliability
- Achieve high accuracy

In particular, we propose two methods that exploit the fact that the image acquisition process is part of the reconstruction pipeline. Our first method is an interactive SfM pipeline that performs a reconstruction from high-resolution still images in an incremental manner. Beside the real-time estimation of camera poses and sparse 3D points, this

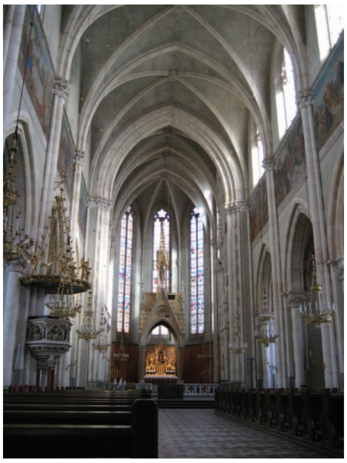


Figure 1.6: The nave of a baroque church is geometrically very complex. Acquiring a set of images that is sufficient for a complete reconstruction is a very challenging task.

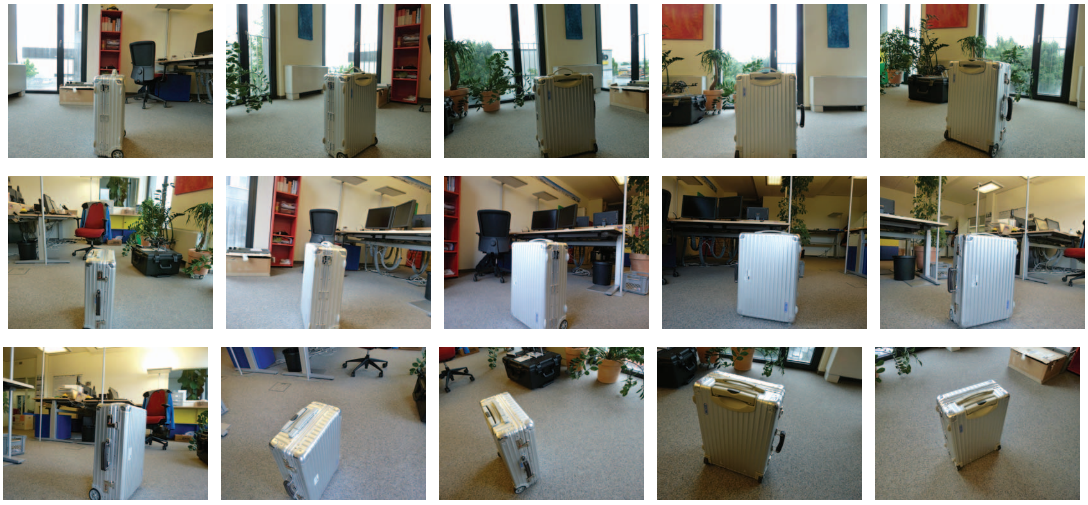


Figure 1.7: Typical dataset for the reconstruction of the suitcase acquired by a non-expert user. The floor is very repetitive and the texture of the suitcase creates only a few feature points. Furthermore, the angles between the images are very large. State-of-the-art SfM pipelines are not able to successfully reconstruct the scene using the input images.

method also extracts a triangular surface in a fully incremental manner. We demonstrate that the interactive SfM is an efficient tool to provide an instant feedback of the current reconstruction quality already during the image acquisition. Hence, this method supports the user in acquiring an image dataset that meets the requirements of a SfM pipeline. In our second method, we exploit the fact that the acquisition process can be controlled by proposing a view planning approach. Assuming that the geometry of the scene of interest is roughly given, we calculate a small set of view points that meets the requirements that are mandatory for a successful reconstruction. This approach is not limited to SfM but it is a general framework to calculate camera poses for a multi-view setup.

## 1.2 Contributions

In particular, we make the following contributions in this thesis.

**Real-time incremental SfM**. Standard SfM pipelines are batch-based methods that require up to hours to finish. This means that the delay between image acquisition and getting feedback about the reconstruction result is high. In order to provide the user a faster feedback, we propose a novel SfM method that incrementally reconstructs the scene from high-resolution images already during the collection of images. We therefore introduce a novel image-based localization that allows very fast processing. With this method the user instantly gets feedback if an image can be localized within the existing reconstruction. The method ensures that if the acquired images can be processed by the incremental SfM, they all can be integrated into a common reconstruction.

**Efficient and robust image-based localization**. Todays research on image-based localization focuses on the registration of high-resolution images within SfM point clouds. The major focus of recent research is to efficiently scale image-based localization to ultra-large image databases [4, 57]. In the setting of the aforementioned real-time SfM, the scene size is typically much smaller but efficiency is an even more important issue. Therefore, we propose a novel image-based localization method that is designed for being computational efficient. We extend an image-retrieval approach to work efficiently with a scale-space pyramid of images. Together with a new feature matching method and an improved RANSAC procedure, we achieve real-time localization rates even for high-resolution image datasets.

**Incremental surface extraction**. The extraction of a surface mesh from a noisy and irregularly sampled point cloud provided by a SfM pipeline is a hard task. Our solution estimates the surface from sparse triangulated image features for the purpose of visualization without a densification step. We formulate the problem as a binary labeling problem of a tetrahedralized point cloud and propose a new random field energy function. Furthermore, the energy function can be efficiently adapted to a growing point cloud. This property together with a dynamic graph cut that optimizes the adapted energy efficiently, enables incremental surface extraction from a growing point cloud while being largely independent from the overall scene size.

**Large-scale view planning for close-range SfM**. Finding a small number of view points around an object of interest whose images are suited for SfM is a non-trivial task. Although this problem is well-known in photogrammetry, the number of existing approaches for solving the close-range problem at a large scale is very low. We formulate the calculation of suited view points around an object of interest as a constrained multi-cover problem. Given the rough shape of the area of interest, we first generate a huge number of potential view points which are then reduced according to a novel quality criterion. Our formulation respects important SfM constraints like redundancy, resolution and triangulation angles. Since the quality function meets the demand of submodularity, we can apply efficient optimization methods while guaranteeing properties on the final solution. Thanks to the efficiency, our method is able to handle large-scale objects of interest. We demonstrate that the resulting view points can be approached by an MAV and that the resulting image set is suited for SfM.

## 1.3 Publications

This thesis is partially composed of publications that have been authored by myself. In particular, the following sections are based on publications:

- Section 4.2 is based on the paper:

  C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online feedback for structure-from-motion image acquisition. In *British Machine Vision Conference (BMVC)*, 2012

- Section 4.3 is based on the paper:

  C. Hoppe, M. Klopschitz, M. Donoser, and H. Bischof. Incremental surface extraction from sparse structure-from-motion point clouds. In *British Machine Vision Conference (BMVC)*, 2013

- Chapter 5 is based on the paper:

  C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, and S. Kluckner. Photogrammetric camera network design for micro aerial vehicles. In *Computer Vision Winter Workshop (CVWW)*, 2012

## 1.4 Outline of the Thesis

The remainder of this thesis consists of five chapters. In Chapter 2, we review the history and the state-of-the-art in SfM and in view planning. In the following Chapter 3, we provide background information about the techniques that have been used in this thesis. In particular, these are multi-view reconstruction from 2D images, image-based localization and surface extraction methods. We present our interactive SfM method in Chapter 4 which consists of a method for real-time SfM, an incremental surface meshing from sparse points and visualization techniques of reconstruction qualities. In this chapter, we also perform experiments on the individual contributions. Our second method that exploits the fact that images are acquired for the SfM process, is an off-line view planning approach which is presented in Chapter 5. Finally, we summarize our work in Chapter 6 and give an outlook on future work.

# Chapter 2

# Related Work

Building 3D maps of the environment or a special object is a complex task and involves a large number of aspects as shown in Figure 2.1. The large number of requirements concerning the map itself, the type of reconstruction process, the process of data acquisition

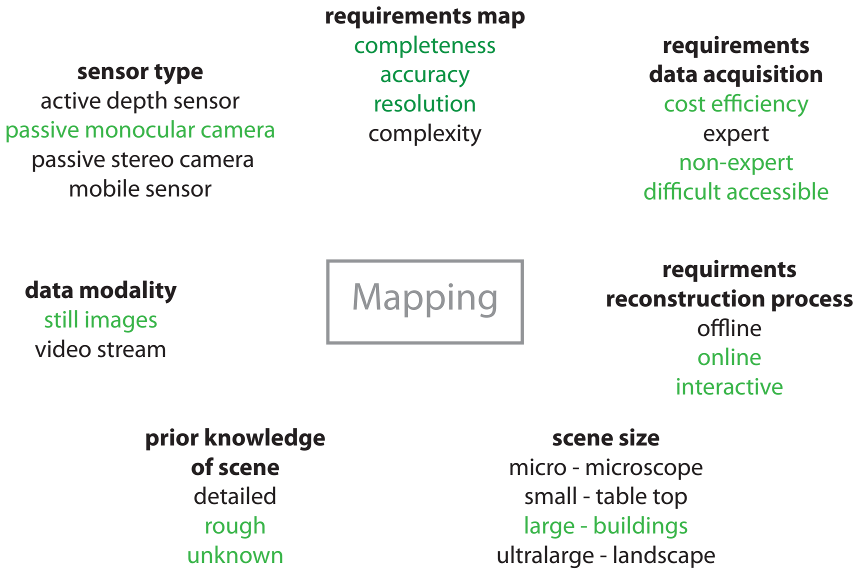


Figure 2.1: Map building has a large number of aspects. The requirements concerning the reconstruction process, the map, the data acquisition and the scene size influences the type of sensor and the data modality. Furthermore, the amount of prior knowledge also influence the choice of the applied reconstruction method. This list does not claim to be complete. Since the amount of factors is so large, we restrict ourselves to the green marked topics.

and foremost the size and the visual appearance of the scene have an impact on the applied sensor and reconstruction method. Due to the large number of impact factors, we concentrate in this thesis on a restricted number of applications that fulfill the green marked topics of Figure 2.1. Hence, we concentrate on applications that require a complete, accurate and a high resolution map of a large-scale but limited scene like an individual building. These assumptions suggests that a cost-efficient mapping can be performed by SfM using high-resolution monocular still images. Furthermore, we consider applications that require view points that are difficult to access like airborne positions. Finally, we assume that for certain proposed methods rough prior knowledge of the scene geometry exists.

Given the scope of this thesis, we will review the major research in two topics that are most relevant for this thesis. In particular, we first investigate the history and state-of-the art of image-based reconstruction and show the weaknesses that prevent SfM to be successful in our desired applications. In the second part, we discuss the related work of sensor planning which is the problem of defining a set of sensor locations such that a map with the desired parameters can be obtained. Since this is a topic that is well-known in photogrammetry and robotics, we will review also research from this disciplines.

## 2.1 History of Image-based Reconstruction

The history of image-based reconstruction started already in 1870 where a technique has been proposed to measure coordinates from 2D images. At this time the name *photogrammetry* has been suggested to describe the new technique [5]. In the early 1900, photogrammetry started as a new research discipline. Photogrammetry can be classified into two parts: close-range photogrammetry and aerial photogrammetry. In the 1920s with the development of airplanes, it was for the first time possible to create large-scale and high-resolution maps. Beside the 2D information there already existed methods to derive 3D information from a pair of images but he manual computation of the analytic solutions was time-consuming. For example, the spatial resectioning of a camera using logarithmic tables took up to three days. Therefore, opto-mechanical measurement instruments like the Stereoautograph have been developed to derive height information from a stereo image pair. In the 1970s with the development of computational hardware the mathematical problems became feasible: the time for re-sectioning decreased from days to milliseconds. The next boost for photogrammetry was in the 1990s where digital cameras became available. With decreasing costs, continuously growing resolutions and increasing computational power, photogrammetric methods became available also for new applications like in industrial metrology or nowadays in mobile phones.

One of the fundamental issues in large-scale photogrammetry is the problem of error propagation. If images are aligned only pairwise, small errors in the pose estimation may sum up and the result is geometrically inconsistent. For example, if images are taken in a loop around an object, a pairwise orientation typically results in an unclosed loop. To bypass this problem, bundle adjustment was developed. Bundle adjustment refines the reconstruction result by jointly optimizing camera poses and 3D structure with respect to some cost function [93]. Classically, bundle adjustment is formulated as a non-linear least-squares problem and therefore an accurate initialization of the problem is required, i.e. camera poses and 3D structure must be close to the final solution.

In 1992, Tomasi and Kanade [92] formulated the multi-view reconstruction problem as a global optimization problem. However, the limitation of this approach is that it is only applicable for affine cameras. Furthermore, it is not robust to outlier correspondences due to the squared error function and the corresponding points have to be visible in all images (each 3D point is visible in each camera). Since this cannot be guaranteed in large-scale reconstructions, this is not applicable in practice. Therefore, researchers tried to find mathematical formulations to solve SfM in a global optimal sense with missing correspondences. Kahl [45] showed that a global optimal solution under the $L_\infty$ norm can be found by quasi-convex optimization but only if the camera rotations are known in advance. However, the $L_\infty$ is very sensitive to corrupted data and for automatic reconstruction unsuited. Following the idea of estimating the camera rotations independently from their translation, several approaches [14, 64, 108] first estimate the rotation between images in a least squares sense and second optimize for the translation between the cameras. The advantages are faster processing and a better initialization for the final bundle adjustment [70]. However, Nister [75] has proven that finding the global optimum of a SfM problem is NP-hard if camera orientations as well as camera translations are involved if missing data exists.

The common ground of all SfM systems are corresponding points. In the beginning of photogrammetry, corresponding points have been determined manually by a human operator. Since this is a very tedious task, computer vision tries to automate the detection of corresponding points by finding visually significant image areas that can be re-identified in other images. Since the available feature matching algorithms in the decade between 1990 and 2000 have been far from being perfect (and they are still not perfect), geometric estimation methods that operate on the matches had to become robust against incorrect correspondences. This resulted in approaches that solve geometric problems with a minimum number of inlier correspondences within a Random Sampling Consensus (RANSAC) [20] loop. However, the limited computational power still prevented large-scale reconstructions. The research of this phase is summarized in the textbook of Hartley and Zisserman [31].

In the 2010s, with the increasing number of photographers that publish their private images on the Internet, SfM became popular to obtain 3D reconstructions from important sights by using Internet image collections. In this time, several SfM pipelines have been proposed that combine robust geometric estimation algorithms with bundle adjustment in an incremental manner. After initializing the reconstruction with a pair or a triplet of images and computing 3D points, further images are inserted with a robust 3-point pose algorithm [51]. To compensate the parameter drifting that causes error accumulation, the bundle adjustment problem is solved iteratively. Although bundle adjustment is computationally complex, it became feasible even for large reconstructions thanks to the increasing computational power. Hence, methods like [2, 89] register thousands of inhomogeneous and unordered images in a very short time. These approaches demonstrate that incremental methods are very robust and accurate while being computational efficient. Therefore, they are still state-of-the-art and we will explain them in more detail in Section 3.5.

The reliability of these approaches is mainly achieved by the ultra-high redundancy contained in publicly available photo community collections. But the enormous redundancy that is required is also a limitation of these methods. Figure 2.2 for example shows the reconstruction and the camera poses of *Notre Dame*. As it is shown, only the main part of the facade can be reconstructed because the image dataset contains here the most redundancy. Other parts which have not been captured that often, are completely missing. Although the overall redundancy is high, it often happens that the reconstruction becomes decomposed into several parts. Already a small part of the object that is captured with a low redundancy can cause this problem.

Another issue is the amount of data that cannot be integrated into the reconstruction. For example, to obtain the reconstruction of *Notre Dame*, 2,635 images have been processed, where only 598 (22%) images can be finally registered [89]. This very low registration rate basically shows two things. First, SfM is picky concerning the input images. A large number of images is rejected for example because their scale or view point change is too large to allow robust pairwise pose estimation. Second, a very large part of the computation time is wasted for images that finally cannot be registered and therefore a large amount of processing time is wasted. This analysis demonstrates that a reconstruction based on fairly randomly taken images requires a very large dataset to obtain reasonable results. Hence, to use SfM reliably and efficiently in applications, where images are acquired deliberately for the reconstruction process, the goal is to acquire an input dataset that contains enough redundancy to obtain an accurate and complete reconstruction without collecting a large number of unneeded images. Hence, a solution to reduce the number of unneeded images and to increases the reliability of SfM is to couple the reconstruction and the image acquisition process instead of handling both problems separately.

## 2.2 View Planning for Mapping

One fundamental problem of 3D reconstruction or map generation in general is that only scene parts are mapped that are captured by the sensor. This is in fact a big problem when mapping large-scale and cluttered scenes. Here it is often not obvious where to place the sensors such that all relevant parts can be mapped. Furthermore, the sensor locations largely influence the accuracy of the resulting map. Typically, the measurement uncertainty increases as the distance between object and sensor grows. Hence, placing sensors close to the surface increases accuracy but also more sensor locations are required to capture the overall scene. Hence, the goal for sensor location planning is to derive mathematical models that find a trade-off between accuracy and completeness with a low number of sensor locations. The two main disciplines that deal with these problems in the field of map generation are robotics and photogrammetry. Therefore, we give an overview of the related work in both research areas.

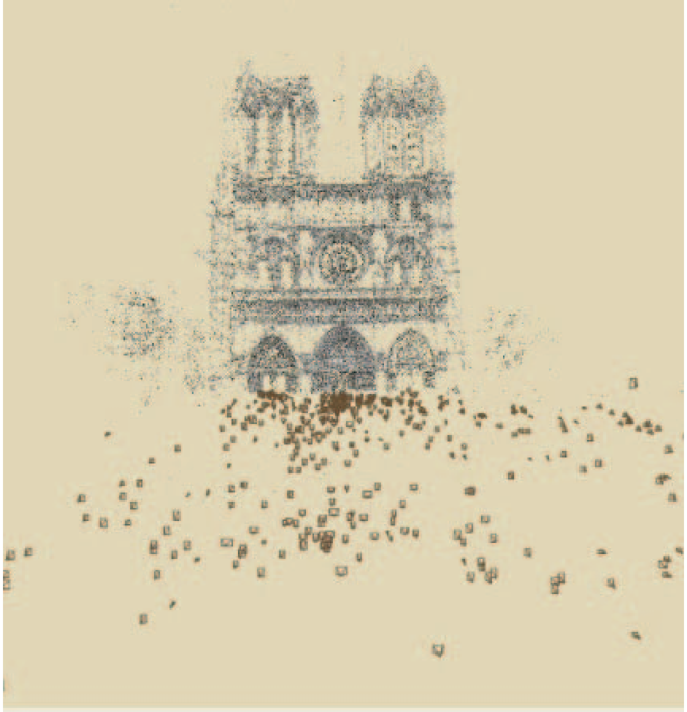
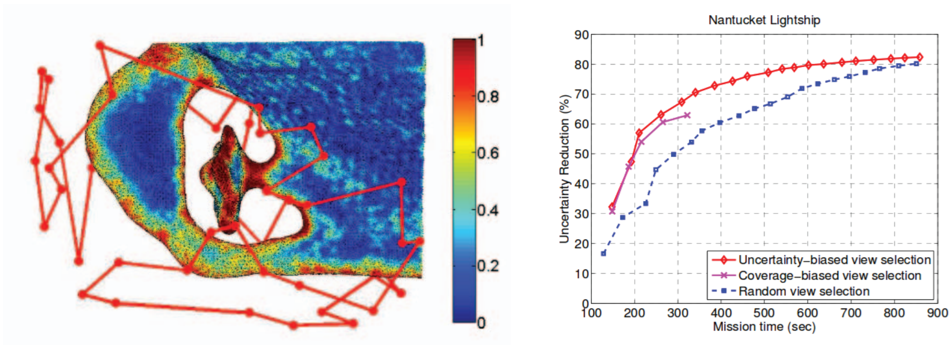


Figure 2.2: Reconstruction Notre Dame. According to [89], 2,635 images have been processed to obtain the final reconstruction that contains only 598 images. As we can see from the distribution of the camera positions, the obtained images are highly redundant.

### 2.2.1 View Planning in Robotics

The typical application scenario for autonomous mobile robots is to explore an unknown environment and simultaneously build a complete map of a limited environment in real-time. Typically, the robot builds an initial map around its starting position and then determines a next sensing position that enhances the map. This procedure is known as Next-Best-View (NBV) planning and the goal is to require as few as possible locations to build an overall map. In the past, research concentrated on wheeled or underwater robots where the weight of the sensor plays a secondary role which allows to carry sensors like laser scanners or sonar. These sensors deliver a 2D or 3D range image of the environment with a single scan. A good overview of existing approaches and their classification can be found in [87]. Recent research on NBV for autonomous robotics is concerned with the question where an object is graspable by a gripper [54, 94, 103]. Typically, some kind of range sensor is mounted on an artificial arm which moves around the object of interest and performs a reasonable 3D reconstruction to identify possible grasping positions. Since



(a) (b)

Figure 2.3: Sonar view planning of Hollinger et al. [33]. The goal is to minimize the uncertainty of the sensed ship surface. (a) The planned sensor locations and the remaining uncertainty of the surface. High-curvature parts have a higher uncertainty than planar regions. (b) The development uncertainty of the overall surface with respect to the plan execution time.

the objects are relatively small and the reconstruction accuracy is not a major concern, it is assumed that potential view points are located in a sphere around the object. The diameter of the sphere is often much larger than the object of interest. For large-scale, close-range reconstructions of arbitrary shapes these assumptions cannot be made.

NBV has to solve two complex problems simultaneously: building a map and defining the view points that can be safely approached to gather new information. Due to the complexity, those methods not only require powerful hardware to solve both problems in real-time but they are also prone to errors. Making an error in one of the tasks has severe impact on the result.

Another application area of robotics, where view planning is required, is inspection. Here, a robot equipped with some type of sensor senses its environment and delivers information about the current state of the object of interest. In those applications often geometric prior information of the object of interest is available which can be used to design a view plan off-line.

Such an off-line view planning approach for the inspection of a ship with an underwater robot equipped with sonar is presented by Hollinger et al. [33]. Given a 3D mesh of the underwater part of a ship, they first model the uncertainty of the surface estimate by a Gaussian process regression. The idea is to identify parts that are prone to measurement noise and therefore need to be inspected with higher precision. The final goal is to determine a set of sensor locations that minimize the uncertainty of the mesh. Hollinger et al. [33] formulate this problem as a sensor selection problem: given a large number of randomly sampled potential locations, a small subset that minimizes the uncertainty of the mesh is determined. Since this selection is a submodular problem in most cases, performance guarantees from submodular optimization theory can be applied directly. Figure 2.3 shows the resulting path of their approach and the reduction in uncertainty with respect to the execution time of the mission.

Since the goal of the mapping process is always to reduce the uncertainty of the environment, the approach of Hollinger et al. [33] is self-evident. However, we show later that in case of a SfM reconstruction, the reduction of uncertainty as the single objective does not deliver the best solution for a reliable reconstruction.

### 2.2.2 View Planning in Image-based Reconstruction

Active sensors like laser scanners or sonars that are frequently used in robotics deliver a range scan of their environment. In contrast, passive cameras only capture a 2D projection of the environment and therefore at least two images acquired from different positions are required to extract range information. Hence, accuracy as well as completeness of the resulting map does not depend on the view point of a single image alone, but it is a result of the overall camera network. Therefore, sensor planning for passive cameras is much more important but also more difficult.

Whereas in most autonomous robotics applications the completeness of the map is often of most interest, in aerial photogrammetry and even more in close-range photogrammetry, accuracy is a major issue and therefore is discussed in more detail. In aerial photogrammetry, images of the landscape are acquired by a high-flying aircraft which are used for reconstructing a 2.5D surface model. Since the aircrafts are flying in a cruising altitude of around 1000 m, which is often much higher than the buildings or vegetation, it is assumed that the observed landscape can be approximated by a 2D plane. Since the cameras are often mounted strictly nadir, the design of a camera network that guarantees certain triangulation angles and overlap is relatively easy. The rules of thumb propose a forward overlap between 60% and 80% and a side-lap of 50% to 60%. With such a design it is guaranteed that each point on the ground is visible in 10 to 15 images. This is motivated by the fact that the accuracy runs into saturation if a point is visible in more than 15 images as shown for example in [81].

The difference between aerial and close-range photogrammetry is that in the close-range case cameras are located much closer to the object such that the assumption of capturing a 2D plane does not hold anymore. Furthermore, in close-range photogrammetry the camera has often more degrees of freedom than just nadir view points. Since the major application of close-range photogrammetry is to measure the relative dimensions of objects, accuracy is a major issue and therefore existing methods focus on minimizing the uncertainty of the reconstructed points.

In order to measure the accuracy of a single reconstructed point, Wenhardt et al. [103] propose three different measures, called D(eterminant)-, E(igenvalues)-, and T(race)-Optimality. The measures are based on properties of the covariance matrix of the reconstructed point. To find a set of view points that maximize the accuracy, Wenhardt et al. propose the following Next-Best-View algorithm. Starting from an arbitrary position, the next position is determined that maximizes the accuracy. Since an efficient optimization is not possible, each time all possible camera positions are evaluated and the best is selected. Therefore, this brute-force approach is limited to a relatively small number of possible positions. The conclusion of this work is that the different measures result in similar reconstructions and the planning delivers more accurate results than just using random camera positions. Building on this work, Trummer et al. [94] propose an extended E-criterion. The extended E-criterion describes the roundness of the uncertainty ellipse of a reconstructed 3D point. The goal is to obtain an isotropic ellipsoid which induces that the uncertainty in all directions is distributed equally. For a single point, the globally optimal camera position to minimize the extended E-criterion can be found in closed-form under the assumption that the camera moves on a sphere around the object. However, it is not straight forward to generalize this to a set of points or for arbitrary camera motion and furthermore, the extended E-criterion does not consider the size of the uncertainty ellipsoid.

Haner et al. [28] propose a NBV system based on covariances specifically designed to work in SfM frameworks, which incrementally adds new images to an existing reconstruction. According to their findings the order in which the images are added plays an important role for the reconstruction quality. Instead of adding the image that has the largest overlap to the existing reconstruction as it is usually done, they calculate the expected covariance for each camera and add the image that has the smallest covariance. In order to determine the covariance of a new camera, they have to propagate the covariance through the whole camera network. Since they integrate cameras only by spatial resectioning without applying bundle adjustment, the impact of adding this procedure into a standard SfM pipeline stays unclear. Furthermore, this work only answers the question in which order images are inserted but the question where the images have to be taken, stays unaffected.

The aforementioned approaches concentrate on the problem of finding a set of view points that minimize the uncertainty of the reconstructed points. However, the important problem of the identification of corresponding points is either bypassed by using video streams that allow frame-to-frame feature tracking or is completely ignored. In aerial photogrammetry, where the view distortions between images are relatively low, automatic feature matching using sophisticated features is also relatively unproblematic. But in the application that we are interested in (close-range, wide-baseline), failures from missing correspondences are predominant. The most common failure is that images cannot be integrated into a single reconstruction and therefore it is decomposed into partial reconstructions. Therefore, a view planning algorithm for our applications has to take into account the wide-baseline feature matching problem.

Up to now, the number of approaches for view planning that take into account the wide-baseline feature matching problem is very low. Schmid et al. [86] propose a method that considers this problem in their view planning approach. They developed a method for off-line view planning for large-scale reconstructions using still images that are acquired by an MAV. They assume that a 2.5D digital surface model (DSM) of the object of interest is given. This DSM is smoothed and eroded to remove sharp edges. Then a large number of potential view points is created that observe the dilated DSM. Finally, a subset of the potential view points is selected. The selection process is based on heuristics that take into account the special requirements of wide-baseline feature matching which is in particular the viewing angle between neighboring views. The approach mainly concentrates on finding a set of view points that cover the overall object. But additional constraints like the accuracy or the redundancy of the reconstructed surface are not considered.

On the one hand, the view distortion, which depends on the geometry of the scene

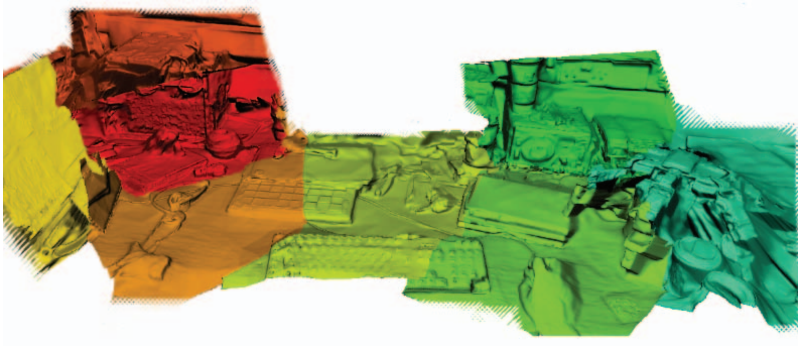


Figure 2.4: Live dense reconstruction of with a single moving camera. The individually depth maps are stitched into a common representation. It is not possible to modify the reconstruction after the depth maps have been stitched. Visualization taken from [71]

and the relative camera orientations, plays an important role for feature matching but also the texture has significant impact on the feature matching problem. Since it is difficult to predict if enough features can be matched to determine relative poses, the most reliable way is to perform the relative pose estimation and therefore the whole reconstruction in real-time already during the image acquisition. This has the advantage that the user can instantly check if the relative pose of a new image to another existing image can be established and therefore a single reconstruction can be obtained. However, such a real-time reconstruction method has not been realized for high-resolution still images but only for video sequences. The realization for video sequences is much easier because they circumvent wide-baseline feature matching problem but typically the resolution of videos is much lower than using still images.

A well-known real-time reconstruction using a video stream has been proposed by Newcombe et al. [71]. They perform at first a sparse reconstruction in a SLAM-like manner and furthermore generate depth maps for selected frames. These depth maps are then stitched together to obtain a dense 3D model (see Figure 2.4). The key contribution of this work is the generation of depth maps using multiple input images. Since dense matching is computationally complex, they first create a rough depth map using the sparse 3D points obtained from the SLAM process. This map serves as a prior do calculate a disparity map using optical flow. The whole procedure works in real-time but once the depth maps are created, they cannot be updated and therefore a refinement is not possible, even if new images are acquired later. In [73] Newcombe et al. extended their work by a new method for integrating the depth maps. Instead of stitching the depth maps, they integrate the depth information into a voxel volume using a signed distance function which allows an easy update if new information is available. The limitation of this approach is that the volume and the resolution of the volume have to be known in advance. The same approach can be directly used for integrating depth maps generated by a RGB-D sensor like the Kinect [72].

To summarize, the existing view planning approaches for image-based reconstruction mainly focused on optimizing the reconstruction accuracy. The wide-baseline feature matching problem is often not considered. Either the approaches are based on a video stream where correspondences are found by frame-to-frame feature matching or the view distortion and texture of the scene is of good nature that feature matching is a minor issue. Hence, there is a lack in research that considers accuracy as well as the wide-baseline feature matching problem in the view planning.

## 2.3 Conclusion

The analysis of the related work shows that research in SfM concentrated on improving the scalability and robustness of image-based reconstruction and nowadays, state-of-the-art methods are able to process thousands of unordered real-world images. Since the success of these methods is based on the enormous redundancy of the input data, they often fail if the redundancy in the dataset significantly drops. Hence, to realize applications with these pipelines reliably, the constitution of the input dataset is of major importance. Another disadvantage of the current state-of-the-art is that the processing follows a strictly feed-forward pattern and therefore, the success and the quality of a reconstruction can be only assessed at the very end of the long-lasting processing. Hence, from our perspective, the uncertainty between the image acquisition and the final result about the reconstruction success is the major obstacle for the success of SfM. This triggered our idea to propose an interactive SfM method, similar to the real-time dense reconstruction from video sequences, that allows an instant assessment of the reconstruction quality.

Calculating a set of view points that allows the reconstruction with a certain accuracy is well-known in photogrammetry and robotics. In robotics, many solutions are specially designed for active range sensors. Therefore, they are often not applicable to SfM reconstruction using monocular, passive cameras. In aerial photogrammetry, the problem is very restricted because it is assumed that cameras are mounted strictly nadir and the landscape is assumed to be a 2D plane. This assumption only holds if the object-camera distance is much larger than the height of the objects located on the surface. Overall, the view planning for image-based reconstruction mainly concentrates on the calculation of a view plan that delivers a maximum degree of accuracy. Since many of these methods rely on video data, the problem of wide-baseline feature matching is not considered during view planning. Since for our problem, the reconstruction of high-resolution 3D data from wide-baseline still images, feature matching causes the majority of errors, a view planning approach has to consider this special property. To our best knowledge, the number of approaches that consider accuracy, completeness and the feature matching problem is extremely low.

# Chapter 3

# Image-based 3D Scene Reconstruction

Reconstructing 3D scene information using multiple input images is one of the fundamental problems in computer vision which is denoted as Structure-from-Motion (SfM). Given a set of 2D images, the goal is to jointly recover their six-degrees-of-freedom (6DoF) pose and the scene structure by using only 2D image information. Thanks to fundamental research, nowadays the basic geometric relationships between 2D images and the 3D environment are well understood. Research in the last decade focused on automating the reconstruction process. Today, we are able to recover camera poses and 3D scene information fully automatic from scenes captured by thousands of images. This basic scene representation by triangulated sparse feature points often serves as input for further processing, e. g. the calculation of a dense point cloud or the extraction of a surface mesh.

In this chapter, we will explain the fundamental geometric concepts between the 3D environment and 2D images and their application to robustly recover 3D information in a fully automatic fashion. In Section 3.1, we start with the mathematical model of a pinhole camera model followed by the mathematical relation between two images in Section 3.2. In Section 3.3, we explain the triangulation of a 3D point from multiple 2D observations and how corresponding points can be identified using image features in Section 3.4. Section 3.5 summarizes the previous sections by describing a state-of-the-art SfM pipeline.

Furthermore, we present two methods that are based on the SfM result, i.e. the extraction of a surface mesh and the localization of new images with respect to a SfM result. In Section 3.6, we review a standard image-based localization approach and in Section 3.7, we have a closer look on existing surface reconstruction algorithms that operate on sparse and dense scene information.

## 3.1 Camera Model

In this section, we describe the geometric relation between 3D points and their projection onto a 2D image. We limit ourselves to the pinhole camera model since this model is appropriate for most consumer cameras. First, we describe the idealized linear camera model and then the non-linear distortions that occur in real cameras.

The principle of the pinhole camera model has been discovered already 1021 AD by the persian scientist Alhazan [99]: A light ray reflected by an object travels through a pinhole and impacts on a planar surface. Mathematically, the pinhole camera is defined by the center of projection $C$ which is the pinhole, the image plane $\pi$ and the focal length $f$ which is the distance between $\pi$ and $C$. The projection of $\mathbf{X} = [X, Y, Z]^T \in \mathcal{R}^3$ to the point $\mathbf{x} = [u, v]^T \in \mathcal{R}^2$ on $\pi$ is given by the intersection of the ray that starts in $\mathbf{X}$ travels through $C$ and intersects $\pi$:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{pmatrix} \tag{3.1}$$

By introducing homogeneous coordinates, we can rewrite Equation 3.1 to a system of linear equations

$$\mathbf{x} = \begin{pmatrix} u' \\ v' \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} [I_{3\times3}|0_{3\times1}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \tag{3.2}$$

where $\mathbf{x}$ is given in homogeneous coordinates and $[I_{3\times3}|0_{3\times1}]$ is a $3 \times 3$ identity matrix concatenated with a $3 \times 1$ zero vector. The image coordinates $(u, v)^T$ can be obtained from $\mathbf{x}$ by normalizing $\mathbf{x}$ with its third coordinate: $u = u'/w$ and $v = v'/w$.

So far, we assumed that the origin of the image plane coincides with the projection of the camera center $C$ to $\pi$. In practice, when dealing with a digital camera, the origin of the image plane is often translated by an offset $p = (p_x, p_y)$, called *principal point*, for example to the upper left corner of the image. Considering the principal point, Equation 3.2 turns

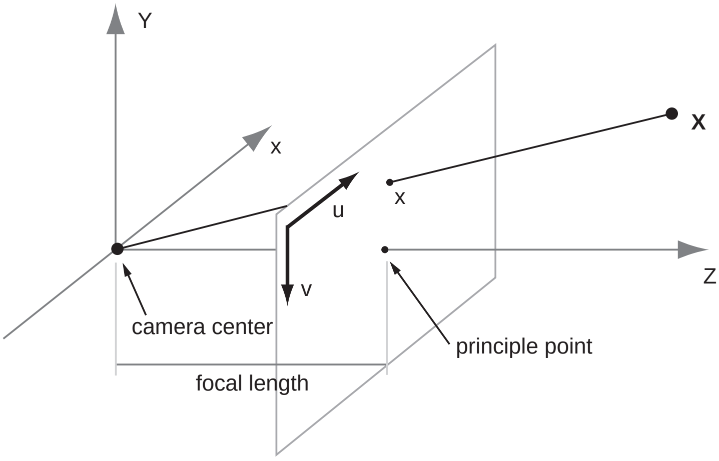


Figure 3.1: The pinhole camera model. The scene point **X** is projected to the 2D point $x$ on the image plan. The intersection of the ray that starts in **X** and ends in $C$ with the image plane is the projected point. The principal point $p$ is the projection of the camera center $C$ to the image plane. The focal length is the distance between $C$ and the image plane.

into

$$\begin{pmatrix} u' + p_x \\ v' + p_y \\ w \end{pmatrix} = \underbrace{\underbrace{\begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}}_{K} [I_{3\times 3}|0_{3\times 1}]}_{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P\,\mathbf{X}. \tag{3.3}$$

The first $3 \times 3$ matrix $K$ describes the *intrinsic* parameters of the camera and is therefore called *camera calibration* matrix. $K$ multiplied with the $3 \times 4$ matrix results in the homogeneous *projection matrix* $P$ which describes the mapping of a homogeneous 3D point **X** to a homogeneous 2D point **x** on the image plane. The intrinsic matrix $K$ defined in Equation 3.3 implicitly assumes that the shape of a pixel on the image plane is a square which is the general case in modern digital cameras. In the rare case that the pixel are sheared an additional parameter $s$ is introduced and $K$ becomes

$$K_s = \begin{pmatrix} f & s & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.4}$$

So far, we assumed that the coordinates of point $\mathbf{X}$ are defined in the (Euclidean) coordinate system that is spanned by the camera. The more general case is that the 3D point is defined with respect to a *world coordinate frame*. To describe the transformation between the camera coordinate frame and the external world coordinate frame, we integrate an Euclidean transformation into the projection matrix $P$

$$P = K \begin{pmatrix} \mathbf{R}_{3\times 3} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} = K[\mathbf{R}|\mathbf{t}]. \tag{3.5}$$

The matrix $[\mathbf{R}|\mathbf{t}]$ rotates and translates the 3D world point into the coordinate system of the camera before the point is projected on the image plane. Since $[\mathbf{R}|\mathbf{t}]$ describes the orientation of the camera to the world coordinate frame the matrix is denoted as *external* parameters. The projection center $C_w$ in world coordinates can be calculated by the inverse transformation of the cameras origin $C = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T$

$$C_w = \begin{pmatrix} \mathbf{R}_{3\times 3} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}^{-1} C = -\mathbf{R}^{\mathbf{T}}\mathbf{t} \tag{3.6}$$

The mathematical model of the pinhole camera (Equation 3.3) is a linear mapping between a homogeneous 3D point and a homogeneous 2D point on the image plane. In real camera systems, this linear model does not always hold due to the use of imperfect lenses which causes non-linear distortions namely *tangential* and *radial* distortion. The tangential distortion is produced by an imperfect alignment of the lens and the image plane. The radial distortion shifts points projected by the linear model inwards or outwards which leads to the effect that straight lines are bended. In practice, the radial distortion is more important and the effect increases as the focal length decreases. Mathematically, the radial distortion is defined by a function $L(r)$ that depends on the Euclidean distance $r$ of the distorted point $x_d = (u_d, v_d)$ to the center of distortion which is typically assumed to be the principal point $(p_x, p_y)$. The undistorted point $x_l = (u_l, v_l)$ that corresponds to the point that would have been obtained with a perfect linear camera is obtained by

$$\begin{pmatrix} u_l \\ v_l \end{pmatrix} = \begin{pmatrix} p_x + L(r)\, u_d \\ p_y + L(r)\, v_d \end{pmatrix}. \tag{3.7}$$

The distortion function $L(r)$ is often approximated by a Taylor function:

$$L(r) = r + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \dots, \tag{3.8}$$

where $\kappa_n$ are called *distortion coefficients*.

Many computer vision algorithms like the estimation of the cameras pose assume a linear projection model. Therefore, the compensation of the non-linear radial distortion is very often a pre-processing step and therefore the estimation of the parameters $\kappa_n$ of $L(r)$ is part of a calibration step. It is often formulated as a minimization problem of an objective function that penalizes the deviation from the linear projection. For more details on camera calibration we refer the reader to [95, 109].

## 3.2 Two-View Geometry

In this section, we derive the geometric relation of two projection matrices $P$ and $P'$ that project the same point $\mathbf{X}$ to the points $x$ and $x'$, respectively. The relation between $x$ and $x'$ is well known as the *epipolar geometry*.

### 3.2.1 Epipolar Geometry

Given a 3D point $\mathbf{X}$ projected by a camera $P$ to point $x$, the epipolar geometry restricts the point $x'$ to be located on a line $l'$ in the image of a second camera $P'$. Figure 3.2 shows that the scene point $\mathbf{X}$ and the camera centers $C$ and $C'$ form a plane $H_\pi$ which is called *epipolar plane*. The points $e$ and $e'$, the *epipoles*, are the projected camera centers $C'$ in $P$ and $C$ in $P'$. The distance between the $C$ and $C'$ is called *baseline*. The intersection of the plane $H_\pi$ with the image planes defines then the *epipolar lines* $l$ and $l'$. Since $l'$ connects $e'$ and $x'$ it can be written as $e' \times x' = [e']_\times x'$ where $[e']_\times$ is the skew symmetric matrix of $e'$. Because $x'$ can also be expressed by $x' = H_\pi x$, we can write

$$l' = [e']_\times x' = [e']_\times H_\pi x = Fx, \tag{3.9}$$

where $F = [e']_{\times} H_{\pi}$ is the *fundamental matrix*. If $x'$ and $x$ correspond, which means that they are the projections of the same point $\mathbf{X}$, then $x'$ is located on $l'$ and

$$x'^T l' = 0 \tag{3.10}$$

holds. From this the *correspondence condition*

$$x' F x = 0 \tag{3.11}$$

is derived. This condition is necessary if $x$ and $x'$ are corresponding points but it is not sufficient, because all points on $l'$ satisfy Equation 3.11.

The $3 \times 3$ fundamental matrix is a homogeneous matrix $F$ and is uniquely defined up to a scaling factor and therefore has 8 independent ratios. Furthermore, the determinant of $F$ is zero and therefore $F$ has 7 degrees-of-freedom.

If the relative orientation and translation between both cameras are known, the fundamental matrix can be easily calculated by the essential matrix $E$ that encodes the relative orientation:

$$E = R[R^T \mathbf{t}]_{\times}, \tag{3.12}$$

where $R$ is the $3 \times 3$ relative rotation matrix and $\mathbf{t}$ the relative translation. $F$ then can be

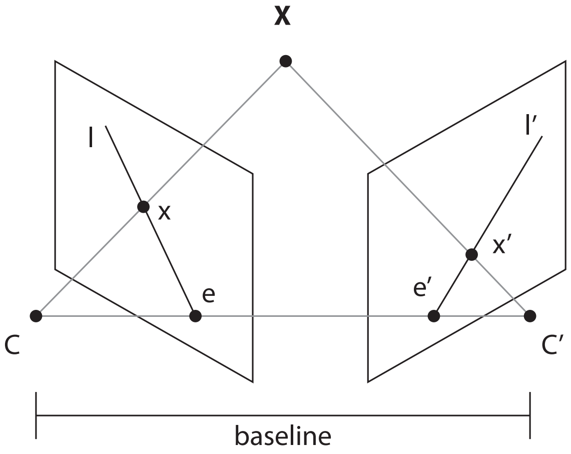


Figure 3.2: Epipolar geometry. The scene point $\mathbf{X}$ is projected by two cameras to the points $x$ and $x'$. The points where the baseline, the line that connects $C$ and $C'$, intersects the image plane are the epipoles $e$ and $e'$. $C$, $C'$ and $\mathbf{X}$ span the plane $H_{\pi}$.

computed by

$$F = K'^{-1} E K^{-1}. \tag{3.13}$$

In the SfM process images are given without knowledge of their exact position. Therefore, one of the main tasks is to recover the relative orientations between image pairs. In the next section, we describe methods to calculate the fundamental matrix $F$ and the essential matrix $E$ by using point correspondences.

### 3.2.2 Estimation of Epipolar Geometry

To estimate the fundamental matrix $F$ from a set of sufficient many corresponding points, we can use the correspondence condition of Equation 3.11 to setup a linear system of equations where each correspondence pair $x_i \leftrightarrow x'_i$ generates one linear equation. Given the points $x_i = (x, y, 1)$ and $x'_i = (x', y', 1)$ in homogeneous coordinates, the resulting linear equation is

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0, \tag{3.14}$$

where $f_{ij}$ are the entries of the fundamental matrix $F$. Re-writing Equation 3.14 as a vector inner product leads to

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\mathbf{f} = 0, \tag{3.15}$$

where $\mathbf{f}$ is the 9-vector containing the coefficients of $F$. By stacking all equations obtained by $n$ (at least eight, therefore the algorithm is called *eight point algorithm*) corresponding points together to a $n \times 9$ matrix $A$, we obtain

$$A\mathbf{f} = 0. \tag{3.16}$$

If more than eight correspondences are given, $\mathbf{f}$ can be found by solving Equation 3.16 in a least-square sense while taking care that $F$ has rank 2. For more details on the eight-point algorithm, we refer the reader to [31].

In many practical problems the intrinsic camera parameters $K$ and $K'$ are known in advance and therefore the estimation of the fundamental matrix reduces to the calculation of the essential matrix $E$. Although $E$ is also a $3 \times 3$ matrix, it describes 3 parameters for the rotation and 3 for translation. Since $E$ is a homogeneous matrix, it is only defined up to scale and therefore $E$ has only 5 degrees-of-freedom which leads to additional constraints on $E$. Since $E$ is a special version of the fundamental matrix, the constraint

$$\det(E) = 0 \tag{3.17}$$

must be fulfilled. Furthermore, Huang and Faugeras [38] has proven that two singular values of $E$ are equal and the third one is zero, which leads to an additional, cubic algebraic constraint

$$EE^TE - \frac{1}{2}trace(EE^T)E = 0. \tag{3.18}$$

These constraints lead to the *Five-Point algorithm* proposed by Nister [74] that calculates the essential matrix from 5 correspondences. Similar to Equation 3.16, a system of linear equations is set up

$$A\tilde{E} = 0, \tag{3.19}$$

where $A$ is the $5 \times 9$ matrix build in the same way as described in Equation 3.15. Then four vectors $\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{W}$ that span the right nullspace are computed by a QR-factorization. By re-writing the vectors $\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{W}$ to $3 \times 3$ matrices $X, Y, Z, W$ the essential matrix can be written as a linear combination

$$E = s_1X + s_2Y + s_3Z + s_4W, \tag{3.20}$$

where $s_1 \dots s_4$ are scalar. Since $E$ is defined only up to scale, $s_4$ is assumed to be one. The factorized essential matrix is then put into the ten cubic constraints obtained by the constraint on the trace and the determinant. After performing Gauss-Jordan elimination with partial pivoting a system of linear equations is obtained. This system is then re-written to a new matrix $B$ that has the constraint that $\det(B) = 0$. The determinant of $B$ is a tenth degree polynomial. To find the roots of the polynomial which are then the solution, Nister proposes to use Sturm-sequences [24] due to their computational efficiency. The full algorithm and more details on the accuracy and the stability are available in [74].

## 3.3 Triangulation

In Section 3.1 we described the relation between a 3D point $\mathbf{X}$ and its projection to $\mathbf{x}$ under the projection matrix $P$: $\mathbf{x} = P\mathbf{X}$. *Triangulation* denotes the inverse problem where we determine $\mathbf{X}$ given two or more corresponding 2D points $\mathbf{x_i}$ and their projection matrices $P_i$.

The projection of $\mathbf{X}$ to $\mathbf{x} = (u, v, w)$ can be re-written into three individual equations

$$\begin{aligned} wu &= P_1^T\mathbf{X} \\ wv &= P_2^T\mathbf{X} \\ w &= P_3^T\mathbf{X}, \end{aligned} \tag{3.21}$$

where $P_n^T$ denotes the $n^{th}$ row of the projection matrix $P$. To eliminate the scaling factor $w$ the projection can be written as a cross product [30] which then results in two independent linear equations

$$\begin{aligned} 0 &= P_1^T\mathbf{X} - P_3^T\mathbf{X}u \\ 0 &= P_2^T\mathbf{X} - P_3^T\mathbf{X}v. \end{aligned} \tag{3.22}$$

Since we are dealing with homogeneous coordinates, our scene point $\mathbf{X}$ has 4 unknowns. Therefore, we require at least two image correspondences $x \leftrightarrow \hat{x}$ and their projection matrices $P$ and $\hat{P}$ to solve the following equation system

$$\begin{pmatrix} P_1^T - P_3^T u \\ P_2^T - P_3^T v \\ \hat{P}_1^T - \hat{P}_3^T \hat{u} \\ \hat{P}_2^T - \hat{P}_3^T \hat{v} \end{pmatrix} \mathbf{X} = A\mathbf{X} = 0. \tag{3.23}$$

This equation system can be easily solved in a least-square sense by performing a singular value decomposition (SVD) on $A$. The Eigenvector corresponding to the smallest Eigenvalue is then the 3D triangulated point. If more than $n > 2$ correspondences are available, each additional correspondence adds two more equations to $A$ resulting in a $2n \times 3$ matrix. The solution can also be obtained by the SVD as before.

In practice, the position of the image correspondences are perturbed by noise and therefore the correspondence condition of Equation 3.11 is only met approximately $xF\hat{x} \simeq 0$. Geometrically, this has the effect that the rays spanned by the corresponding points do not intersect in a single point $\mathbf{X}$, but they form a glancing intersection. Due to the uncertainty in the localization of the corresponding points, the triangulation result also has an uncertainty. The relation between the 2D uncertainty originated by the corresponding point estimation and the resulting 3D uncertainty is shown in [9]. They demonstrate that the covariance matrix of $\mathbf{X}$ can be calculated by the covariance matrices of the individual corresponding points.

## 3.4 Visual Features

In the previous section we discussed how the relative orientation between two cameras can be automatically recovered if corresponding points $x_i \leftrightarrow x'_i$ are available. Hence, the key component for automatic image-based scene reconstruction is the ability to find correspondences between image pairs reliably. Because this is one of the fundamental problems in computer vision, an enormous number of approaches to establish correspondences automatically has been developed. A detailed survey of state-of-the-art methods is available in [96].

The basic concept of most methods is to extract points, called *keypoints*, that are discriminative according to their surrounding and then to find a description of its neighborhood (called *feature*) that is discriminative and stable under transformations like lighting, viewpoint, scale and rotation. Most descriptors are represented by an $n$-dimensional vector whose entries stay nearly constant even under the aforementioned distortions. To establish correspondences between two images, for each feature of one image its counterpart is determined by finding the feature of the second image that is most similar. Hence, the matching problem can be casted to a nearest neighbor problem.

The selection of a certain keypoint/descriptor combination always depends on the requirements of the underlying application concerning computational complexity and variance of the input images. For example, when tracking 2D points frame-wise in a video stream, the descriptor does not have to be extremely robust against lighting and viewpoint changes. Therefore, lightweight descriptors which are computational efficient can be used. On the other hand, the reconstruction of wide-baseline images by SfM requires features that are highly robust against a large number of distortions: viewpoint changes, lighting, scale and rotation. The robustness of the used feature is directly related to the variety of images that can be reconstructed by the SfM process.

For SfM using wide-baseline images the Scale Invariant Feature Transform (SIFT) [62] proposed by Lowe is an approved keypoint/descriptor combination. SIFT combines a keypoint detector that identifies blob-like structures in a scale-space pyramid using the difference-of-Gaussian (DoG) function and a descriptor that represents the distribution of gradient orientations around the keypoint in a histogram with 128 entries. The information on which scale the maximum response of the DoG function occurs is used to define the patch size that is taken into account for building the descriptor and therefore makes the descriptor invariant against scale changes. The descriptor itself is inspired by the biological observation that gradient information is much more stable under 3D viewpoint variations than the gray-scale information. Loewe has shown in [62] that the matching quality is stable up to a viewpoint change of about 30 degrees.

Since SIFT achieves very good results for wide-baseline matching, several modifications have been proposed to either reduce the computational effort like the Speeded Up Robust Features (SURF) [8] or for being more robust against affine viewpoint changes like the Affine-SIFT (ASIFT) [68]. Nevertheless, the original SIFT method is a trade-off between quality and computational complexity and thanks to the availability of efficient implementations on the GPU [105], the computation is feasible even for high resolution images.

A new class of descriptors that have been developed in the last years are binary descriptors that directly operate on the gray-scale pixel values instead of the images first-order-derivatives. These descriptors have in common that they are built from a pairwise intensity comparisons. The result of each comparison is thresholded and then stored as a single bit in the feature. Hence, a feature consists of a bitstring instead of a floating-valued vector. As a similarity measure typically the Hamming distance is used because this can be very efficiently calculated on today's computational hardware. Typically, the distance calculation of a feature pair requires less than 0.3 micro-second which is two orders of magnitude faster than the distance calculation between two SIFT features (33 micro-seconds) [91]. However, most of these binary features are not as robust against view distortions as for example SIFT, and therefore they are rarely used for wide-baseline feature matching. They are often applied for feature matching in video sequences. An exhaustive comparison of binary features can be found in [32]

### 3.4.1 Feature Matching

Feature matching is the task to identify corresponding features given two sets of features $F$ and $F'$. We will explain the procedure using the example of SIFT but the principle can be applied to most real-valued features.

Given an $n$-dimensional feature $f \in F$ we want to find the corresponding feature $f' \in F'$ that shows the same scene point. To make things worse, it is not known if $F'$ even contains the feature that shows the same scene point. The matching procedure proposed by Lowe [62] is based on the features $f'_n$ and $f'_m \in F'$ that are the nearest and second nearest neighbor of $f$ under some metric $\|\cdot\|$

$$f'_n = \min_{f'_i \in F'} \|f'_i - f\|. \tag{3.24}$$

Typically, the Euclidean distance is used as metric. The second nearest neighbor $f'_m$ is used to decide if $f$ and $f'_n$ are correspondences. If

$$\frac{\|f - f'_n\|}{\|f - f'_m\|} < \tau \tag{3.25}$$

the match is established otherwise it is discarded. Depending on the application $\tau$ is chosen between 0.5 and 0.8. The intuition behind this definition is to measure the discriminativety of the match. The smaller the ratio of Equation 3.25 is, the more reliable the match is. Lowe has shown that this measure establishes more correct matches than just using a threshold on $\|f - f'_n\|$.

Since several thousand features can be extracted from high resolution images, the brute-force nearest neighbor search is computationally expensive. Therefore, various approaches have been presented to speed up the search [88]. However, the current solutions for exact nearest neighbor search are exponential in the feature's dimension in either time or space complexity [3]. Fortunately, the brute-force method to calculate all possible pairwise feature distances in case of normalized feature vectors can be efficiently parallelized and therefore computed on the GPU. Given two normalized feature vectors $q$ and $d$, their

Euclidean distance can be expressed by

$$\|q - d\| = 2 - 2\sum_i q_i d_i \tag{3.26}$$

as a vector multiplication [76]. Hence, the pairwise feature distances can be calculated by a dense matrix multiplication. Query features are stacked row-wise to a matrix $Q$ of size $n \times k$, where $n$ is the number of features $k$ the feature dimension. In the same way, matrix $D$ of size $m \times k$ is build using the other set of features. The distance matrix $G$ is then obtained by the matrix multiplication

$$G = QD^T, \tag{3.27}$$

where $G$ is of size $n \times m$. Since the matrix multiplication can be efficiently parallelized, the computation on the GPU is very fast. Also the calculation of the row-wise or column-wise minimum to find the nearest neighbor can be easily parallelized.

## 3.5 Multi-view Reconstruction

In multi-view reconstruction we are interested in recovering a previously unknown 3D scene structure from a set of redundant 2D images by just using image information. This is one of the fundamental computer vision problems and therefore is active research over the last three decades. As described in Section 2.1, the State-of-the-art for SfM from unordered images are incremental reconstruction pipelines that iteratively perform global optimization to obtain an accurate result. In the following, we describe the different steps of such a pipeline.

The reconstruction pipeline typically consists of several steps that are conducted subsequently:

- Visual feature extraction
- Pairwise feature matching
- Geometric verification
- Geometry estimation
- Geometry optimization

In the first two steps, visual features are extracted and corresponding points are established by matching features pairwise. In the third step, the relative camera orientations, more formally the essential matrix $E$ between all image pairs are computed. This results in the *epipolar graph* that encodes which camera pairs share common scene parts. In the subsequent step, the relative orientations are integrated into a common coordinate system and scene points are triangulated. To prevent error propagation during the integration, the structure as well as camera positions are globally optimized by *bundle adjustment*.

### Visual feature extraction

The first step of the pipeline is to extract distinct visual features and their corresponding descriptors. Since SfM is often performed on still images that are taken from quite different positions, a descriptor is required that is robust against a large number of distortions. Because SIFT [62] has a good trade-off between speed and robustness, it is used in many standard SfM pipelines. However, the feature itself is only important for the subsequent feature matching to establish corresponding points. For all other reconstruction steps the feature itself is irrelevant since only corresponding points are of interest.

### Pairwise feature matching

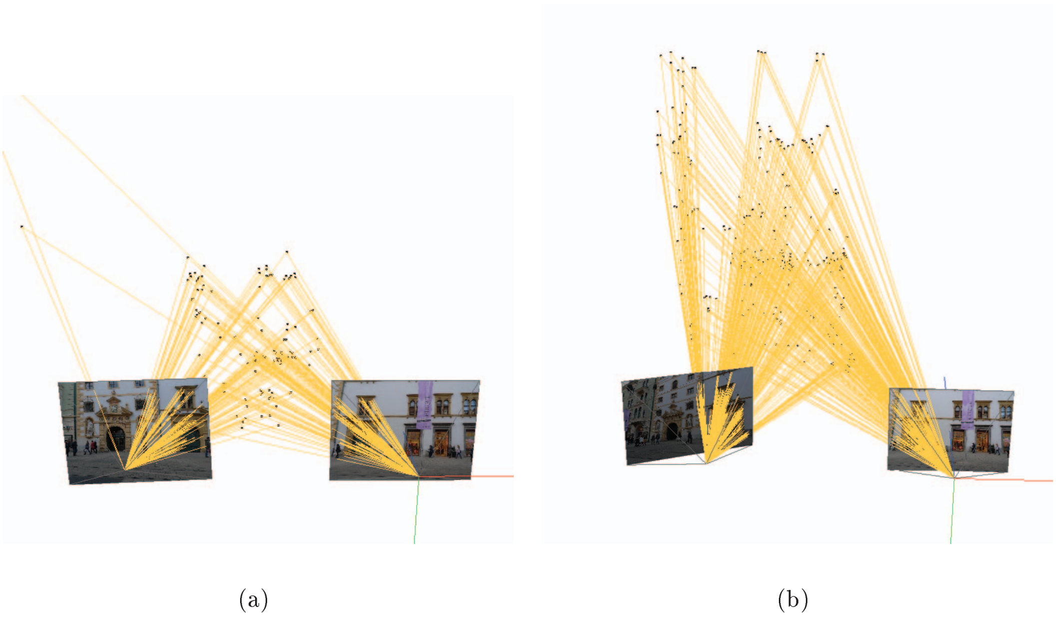
Feature matching between image pairs is performed to find corresponding image points that are later used for calculating the epipolar geometry as described in Section 3.2.2. Because images are unordered and it is not known which images show a common scene part, it is required to match all possible image pairs. The computational effort therefore is $n^2$ in the number of images. On large datasets containing hundreds or thousands of images matching is often the most time consuming part. Klopschitz et al. [48] propose to split the matching in a coarse and a detailed matching step to reduce the computational complexity. The goal of the coarse matching is to restrict the matching to images that potentially show the same scene part. The authors propose to use an image retrieval approach [74] to rank images according to their visual similarity. Feature matching is then performed only on the $k$ top-ranked images. Therefore, the computational complexity is $n\,k$ and therefore is linear in the number of images. This allows to speed up the feature matching and even large image sets can be processed.

### Geometric Verification

In this step, we estimate the relative orientation between the matched image pairs by calculating their fundamental matrix $F$. In the case where the intrinsic camera parameters of both images are known, this reduces to an estimation of the essential matrix $E$ using the Five-point [74] algorithm. The geometric verification is performed for two reasons. First, false feature matches can be identified and eliminated. Second, the relative orientations provide information about the special distribution of the images and their redundancy.

To calculate the fundamental or essential matrix, the corresponding points obtained from the feature matching are used. Since they are potentially highly contaminated by false matches, the estimation has to be performed in a robust manner. The standard technique for fitting models to point sets that contain a large number of outlier is the random sampling and consensus (RANSAC) [20]. RANSAC estimates the optimal solution by randomly selecting the minimum number of correspondences that is required to calculate a hypothesis $H$. Then the portion of corresponding points is determined that supports $H$. To decide if a correspondence supports $H$, an error metric and threshold have to be defined. In case of the epipolar estimation, the support is measured by the number of correspondences whose Sampson error [31] is below a certain threshold. The procedure of hypothesizing and testing is repeated a fixed number of times to find the hypothesis $H_{max}$ that has the highest number of inlier. If $H_{max}$ contains less than $k$ inlier, we assume that no epipolar geometry between the image pair can be determined using corresponding points. Instead of setting $k$ to a fixed pre-defined value, Irschara et al. [40] propose a method to calculate $k$ by taking the spatial distribution of inliers into account. In practice, this method reduces the chance of getting wrong relative poses especially if a very large number of correspondences are given or the scene consists of repetitive structures.

The result of the geometric verification is the so-called *epipolar graph* $E_{\mathcal{G}}$. $E_{\mathcal{G}}$ is a directed graph whose nodes are the images and the edges are the relative orientations between the image pairs. It is typically used to initialize the next processing step that lifts the pairwise relative orientations into a common coordinate system. Figure 3.4 shows two epipolar graphs that have different characteristics.



(a)

(b)

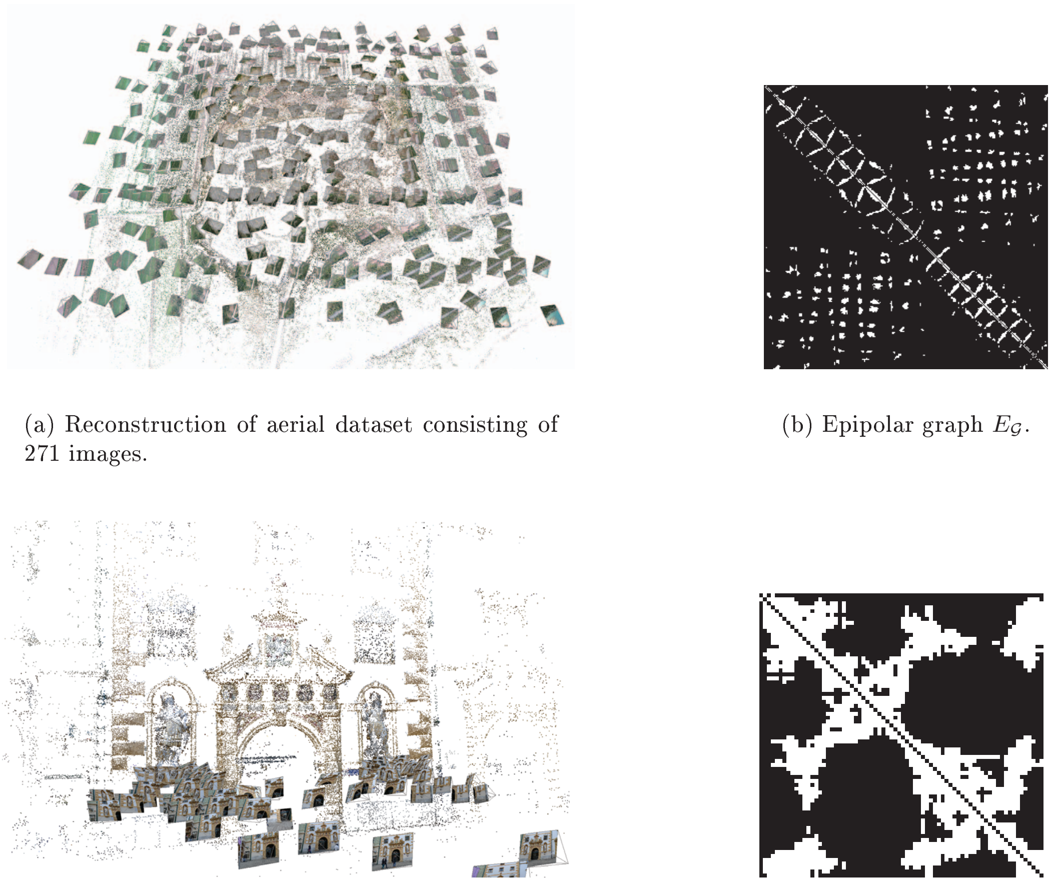
Figure 3.3: Two examples of a verified image pair. The right camera is in the origin and the second camera is set to the pose estimated by the five-point algorithm. The yellow lines indicate the viewing rays to the triangulated image features.

### Geometry estimation

In this step, the individual relative orientations determined in the geometric verification are lifted into a common coordinate system following the greedy approach of [89].

Given the relative orientation from an initial image pair, we triangulate the corresponding points that are inliers of the geometric verification. Next, an additional image is selected that is registered with respect to the existing reconstruction using an absolute pose algorithm [51]. In case of known camera intrinsics, at least three 2D–3D correspondences have to be known to compute the pose of the new image. To be robust against misleadingly identified corresponding points, the pose is estimated in a RANSAC loop. If the new image can be localized, the 3D structure is expanded by triangulating new correspondences. This procedure is repeated until all images are integrated or no image can be registered anymore.

Although the algorithm is basically very simple,the selection of the initial image pair is very crucial. It influences the reconstructions accuracy as well as the number of images that can be localized within a common reconstruction. Snavely et al. [89] propose to choose an



(a) Reconstruction of aerial dataset consisting of 271 images.

(b) Epipolar graph $E_{\mathcal{G}}$.

(c) Reconstruction of a facade using 72 images.

(d) Epipolar graph.

Figure 3.4: Two reconstructions and their corresponding epipolar graphs. The reconstruction in (a) is obtained by 271 images acquired by a fixed wing Micro Aerial Vehicle in two flights with crossing stripes. The epipolar graph represented as a 2D matrix in (b) shows if a relative pose between an image pair has been determined by the five-point algorithm. The second scene (c) is reconstructed by 72 images. (d) The epipolar graph here is much more dense than at the aerial reconstruction. From this we can conclude that the images are more redundant than in the aerial dataset.

initial pair that has a large baseline as well as a large number of correspondences to get a well-constraint initial reconstruction. Klopschitz et al. [48] in contrast propose to first find the image in the epipolar graph $E_{\mathcal{G}}$ that has the highest degree, i.e. has the most verified epipolar geometries and therefore is a central element of the reconstruction. The second image is then chosen among all neighbors with a large baseline and a large number correspondences.

Since the reconstruction is the result of an incremental approach the problem of error propagation occurs, i.e. small errors in the camera pose estimation may accumulate to larger errors as more and more images are getting integrated. Periodically, the estimated geometry is globally optimized to avoid error accumulation.

### Geometry Optimization

The global optimization of the estimated geometric configuration of the previous step is known as *Bundle Adjustment* [31, 60, 93] which is defined as follows: given a set of projection matrices $P_i$, a set of 3D points $X_j$ and the observation $x_{ij}$ which is the measured 2D position of $X_j$ in image $P_i$. The task is to jointly optimize $P_i$ and $X_j$ such that a cost function is minimized. Typically, the *reprojection error*

$$\epsilon_{ij}(P_i, X_j) = d(P_i X_j, x_{ij}) \tag{3.28}$$

is used to define the cost function $\mathcal{C}(P_i, X_j)$. The reprojection error measures the Euclidean distance $d(\cdot)$ between the re-projected position $X_j$ in camera $P_i$ and the observed position $x_{ij}$. The overall cost function to be minimized is then defined as the sum over all residuals $\epsilon_{ij}$

$$\mathcal{C}(P_i, X_j) = \sum_i \sum_j \nu_{ij} \rho(\epsilon_{ij}), \tag{3.29}$$

where $\rho(\cdot)$ is a function on the residuals and $\nu_{ij}$ is a binary variable that indicates if the point $X_j$ is visible in camera $P_i$.

Under the assumption that the observations $x_{ij}$ are perturbed by Gaussian noise the maximum likelihood estimate is obtained by setting $\rho(\cdot) = \frac{1}{2}(\cdot)^2$. Hence, the minimization of Equation 3.29 turns into a least-squares optimization problem.

In practice, the noise of the observations is not Gaussian because the observations $x_{ij}$ are obtained by feature matching and therefore can be contaminated by severe outlier. In case of the least-squares solution, a single outlier can distort the whole reconstruction. Therefore, it is desired to find a cost function $\rho(\cdot)$ that is robust against those outlier. In 1964 Huber et al. [39] proposed the class of M-estimator functions (“M” stands for “maximum likelihood-type”) that gives individual samples different importance according to their residual $\epsilon_{ij}$. Functions often used in robust bundle adjustment are for example the

Tukey and the Huber [39] M-estimator. The Tukey M-estimator

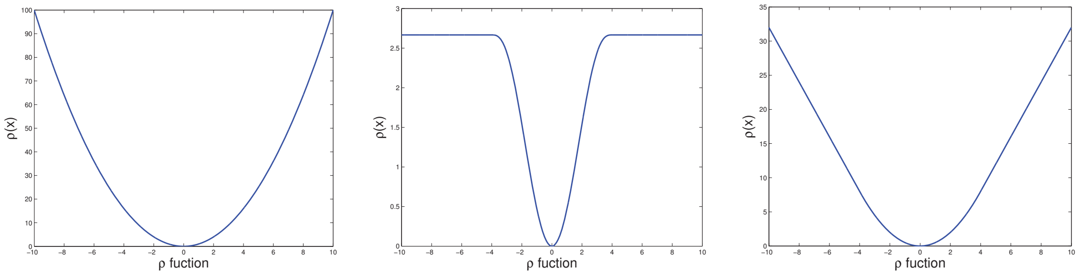
$$\rho_{Tukey}(\epsilon_{ij}) = \begin{cases} \frac{c^2}{2}(1 - [1 - (\epsilon_{ij}/c)^2])^3, & \text{if } |\epsilon_{ij}| \leq c \\ \frac{c^2}{6}, & \text{if} |\epsilon_{ij}| > c \end{cases} \tag{3.30}$$

assigns outlier above a threshold $c$ a constant value whereas residuals below $c$ have a squared impact on the optimization result. Figure 3.5(b) shows the shape of the Tukey M-estimator. In contrast, the Huber M-estimator (Figure 3.5(c))

$$\rho_{Huber}(\epsilon_{ij}) = \begin{cases} \frac{x^2}{2}, & \text{if } |\epsilon_{ij}| \leq c \\ c(|e_{ij}| - \frac{c}{2}), & \text{if} |\epsilon_{ij}| > c \end{cases} \tag{3.31}$$

combines a squared part for points with a reprojection error below $c$ and points above $c$ have a linear influence.

The cost function 3.29 is non-convex and therefore finding the global optimum efficiently is not possible [75]. To determine a local optimum, typically iterative non-linear least-square solvers like the Levenberg-Marquardt [93] algorithm are used. Due to the high non-convexity a good initial guess of the parameters is essential.



(a) Squared error function (b) Tukey (c) Huber

Figure 3.5: Common cost functions for bundle adjustment. (a) Squared error function. (b) Tukey M-estimator. (c) Huber M-estimator.

## 3.6 Image-based Localization based on SfM Results

Recovering the 6DoF camera pose for localizing a query image with respect to a 3D point cloud recently has been a popular field of research, especially due to the recent advances in SfM which now allow the reconstruction of large-scale 3D models of urban scenes within a few hours. It has a wide range of applications starting from the localization of user community photos [40, 57, 83, 85] up to the localization of video sequences as used in Augmented Reality (AR) [4] or autonomous robotic applications [58, 102].

Most image-based localization methods can be divided into three consecutive steps: extracting features, establishing 2D–3D correspondences by feature matching and analyzing the obtained correspondences in a robust hypothesize-and-verify algorithm like RANSAC to estimate the camera pose. Existing methods can be classified into two categories depending on how they establish the 2D–3D correspondences, either image-retrieval based or by brute-force 2D–3D matching. Methods based on image-retrieval reduce the number of 3D point candidates for matching by constraining the search to the visually most similar images in the database using well known image retrieval techniques like the vocabulary tree [76]. The main advantage of such approaches is that they are able to handle even very large 3D reconstructions, since image-retrieval scales well with increasing database sizes. In such a way the quality of the ranking is very important for the localization performance. To increase the retrieval quality, e. g. Irschara et al. [40] proposed a new ranking method (probabilistic scoring) and introduced *synthetic views* that simulate additional camera poses within the reconstruction.

Sattler et al. [83] have shown that in many cases brute-force 2D–3D matching between the query and the whole point cloud increases the number of successfully localized frames. To reduce the matching effort, features of the point cloud are stored within a vocabulary tree. For matching a query feature, only 3D points are considered that are assigned to the same leaf node as the query feature. For this method, they report timings between 160 ms and 740 ms per image. The disadvantage of direct matching is that all features have to be stored in memory which is critical for very large reconstructions. In [84], the same authors combine the image-retrieval approach with direct matching. A ranking of the input images based on a limited matching is used to improve the image retrieval quality. To reduce the memory demands, they introduce quantized SIFT features that can be efficiently matched by calculating their Hamming distance.

## 3.7 Surface Extraction from SfM results

The result of a Structure-from-Motion pipeline are the camera poses as well as a set of triangulated, sparse image features. For many applications in robotics, Augmented Reality (AR) or for visualization, a sparse point representation of the environment is not sufficient because the inhomogeneous sampling of the surface makes a geometric interpretation of the scene very difficult. Tasks like obstacle avoidance or physic simulations often require a denser representation or even better, an implicit or explicit representation of the surface. Most state-of-the-art methods for image-based surface reconstruction operate on dense depth information which means that the scene is represented by significantly more 3D points as obtained by the point cloud generated by the SfM pipeline. Due to the enormous amount of data to be processed, most of these algorithms are computational very demanding. For applications where computational power is limited, it is an alternative to work on the sparse data directly. However, the sparseness and the limited redundancy of the scene representation makes the extraction from such data even more challenging.

Most approaches formulate the surface extraction problem as a classification problem of a discretized volume into free- or occupied space. Therefore, we first introduce two different volume discretization methods, namely a regular and an irregular discretization schema. Second, we give an overview of existing methods that require dense and regularly sampled 3D points of the scene to extract a surface. Finally, we concentrate on state-of-the-art methods for surface reconstruction methods from sparsely and irregularly sampled point clouds which are obtained from SfM.

### 3.7.1 Volume Discretization

Most existing surface reconstruction methods are based on a volumetric discretization of the space. The discretization can be classified by the shape of volume elements which can be *regular* or *irregular*. The best-known regular volume element is the voxel which is the generalization of a 2D pixel in 3D. A well-known irregularly shaped volume element is the tetrahedron that is often used for the surface extraction from sparse data.

In the following, we describe the discretization methods and analyze their advantages and disadvantages.

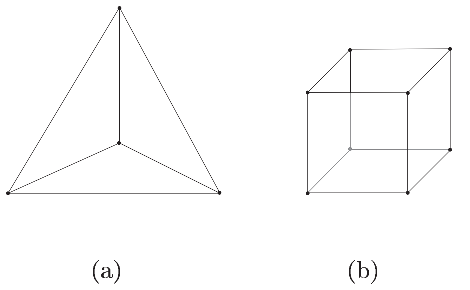


Figure 3.6: Two volume elements. (b)Voxel. (a) Tetrahedron.

### Regular Discretization

The basic element of a regular discretization is a voxel which is an equilateral cube and can be seen as an extruded pixel in 3D as shown in Figure 3.6(b). The advantage of the voxel-based discretization is that it can be easily obtained if the dimension of the volume and the discretization resolution are known. Assuming an axis-aligned Euclidean volume, the voxel coordinates for a certain 3D point can calculated by a division and a rounding operation. Due to the regular structure of the volume discretized into voxel, it is very well suited for being processed on modern GPU hardware. However, the regular discretization requires that the resolution as well as the dimension of the volume has to be chosen at the programs start. A belated modification of both parameters is often not possible. In case of a surface reconstruction, a disadvantage is that the volume is uniformly discretized. Since a large number of reconstructed scene points are typically close to the surface, most voxel are empty whereas in parts of the surface several points are discretized into a single voxel. An alternative to bypass this problem are octrees [82] that are hierarchical 8-ary tree structures with varying resolution. However, those cannot be handled efficiently on the GPU anymore.

### Irregular Discretization

Most existing surface reconstruction algorithms for sparse point clouds perform an irregular discretization into tetrahedra. In contrast to the regular discretization that ignores the density of the 3D points, the irregular discretization is based directly on the 3D points and therefore adapts to the density of the point cloud. A tetrahedron is a volume element that is spanned by 4 3D points where each of the 4 points is connected to all other points. This results in a volume element that is bordered by 4 triangles and looks like a pyramid with a triangle footprint (Figure 3.6(a)). Given an arbitrary set of points, there is a large number of alternatives to connect 4 points such that they form a tetrahedral structure. However, for a detailed surface reconstruction we are interested in a discretization that results in small and equally shaped triangles. The Delaunay Triangulation (DT) [16] method is one approach that meets these requirements.

Formally, the DT is defined as follows: Given a triangulation $DT(P)$ of a point set $P \in R^n$, $DT(P)$ is a DT if no point of $P$ is inside the circum-hypersphere of any simplex of $DT(P)$. In the case of a 2D point set ($n = 2$), the circum-hypersphere is a circle and

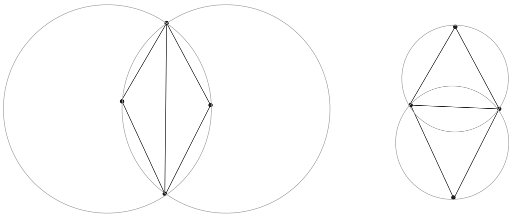

(a) (b)

Figure 3.7: Triangulations in 2D. (a) is a triangulation in 2D that does not fulfill the *empty sphere* property of a valid DT. In contrast the triangulation of the same point set in (b) is a valid DT.

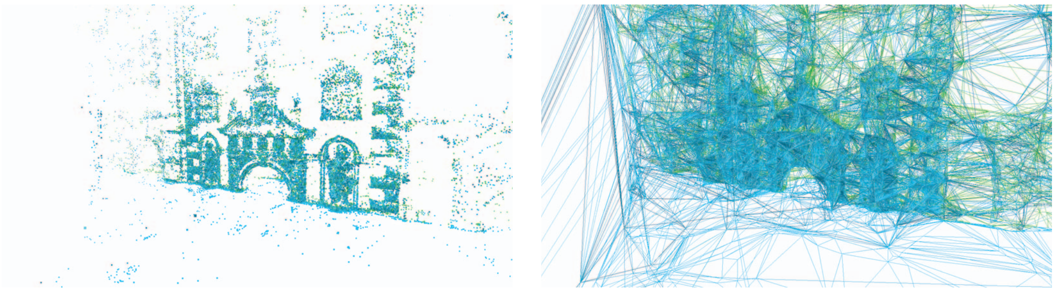

(a) (b)

Figure 3.8: Real-world 3D Delaunay Triangulation example. (a) Input point cloud. (b) DT illustrated in a wire-frame presentation. Outlier points in front of the facade cause large volume elements whereas the dense points on the surface yield a fine-detailed discretization.

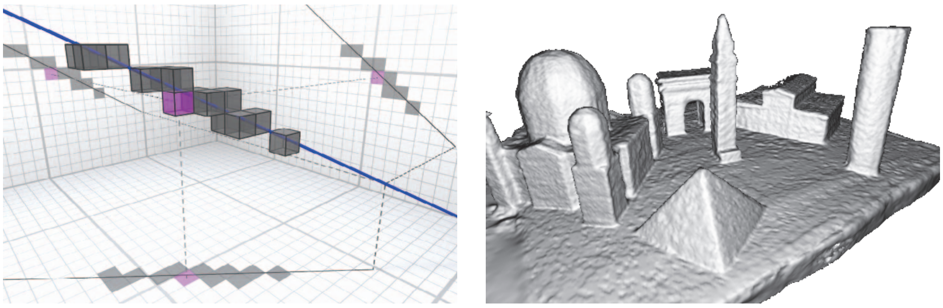
the simplex is a triangle, whereas in 3D ($n = 3$), the simplex is called tetrahedron and the circum-hypersphere is the circum-sphere. One of the most important properties is that this triangulation maximizes the minimum angle of all simplexes in the triangulation and therefore avoids the generation of skinny simplexes. Figure 3.7 shows two possible triangulations of 4 points in 2D. The triangulation shown in Figure 3.7(a) does not meet the empty circle criterion in contrast to the triangulation illustrated in Figure 3.7(b). The DT has several properties. The tetrahedral structure is regular, i.e. each tetrahedron (except tetrahedra on the convex hull) is connected to four neighboring tetrahedra by sharing one face which in 3D is a triangle. In 2D the number of simplexes in the triangulation is $\mathcal{O}(n)$ whereas in 3D the number can vary between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$. Attali et al. [6] have shown that the complexity of the construction is bounded by $\mathcal{O}(n\,log\,n)$ under a mild uniform sampling condition which often holds in the case of surface reconstructions. Figure 3.8 shows a real-world example of a triangulated sparse point cloud.

For the generation of a valid DT, numerous methods exist like divide-and-conquer approaches, local improvement or incremental insertion methods [65]. For our desired purposes, the incremental insertion approach is the most suitable one. The basic concept of the incremental DT algorithms is to start with the minimum set of points that define the simplex and to add points in an iterative manner. Each time a new point is added, the simplex that contains the new point is partitioned. Then the circum-hypersphere criterion is tested for all simplexes adjacent to the new ones. If the criterion does not hold, a *local transformation* is performed which is the flip of a face between adjacent tetrahedra. Local transformations are performed recursively until the empty-sphere criterion is again true for all simplexes in the triangulation. More details on the incremental DT and local transformations can be found in [44]. In the worst case, all existing tetrahedra are destroyed and replaced by completely new set of volume elements. In practice, the tetrahedra that are destroyed are bounded locally such that only a limited number of new tetrahedra are created.

### 3.7.2 Surface Reconstruction from Dense Data

State-of-the-art methods for surface extraction from dense depth data can be classified into two groups:

- Implicit surface representation in a discretized volumetric space.



(a) (b)

Figure 3.9: Voxel in front and behind the estimated depth of a viewing ray. Pixel in front get positive and voxel behind get negative values. The surface is then the zero crossing. (Illustration courtesy of [26]).

- Robust meshing of a densified point cloud.

The above mentioned methods have in common that they assume a “sufficiently dense” and equally distributed 3D sampling of the underlying surface.

A very popular method for solving the surface estimation problem given dense depthmaps is based on the embedding of an implicit function into a regularly discretized volume [72, 80, 104]. The surface itself is then found by extracting an appropriate iso-surface of this function. In particular, many successful methods are based on a truncated signed distance function [107]. Given a 2.5-dimensional depthmap and the corresponding camera pose, a truncated signed distance field along the viewing ray is accumulated in the voxel space. Since the density of the depth information is typically higher than the resolution of the discretized space, the surface can be robustly obtained by finding the iso-surface within the volume. The enormous amount of redundancy in the depthmap also causes that the method is very robust against noisy depth information. Inherently, this approach can work in an incremental manner and thanks to powerful GPU hardware, it can be applied by real-time applications [26, 72]. The disadvantage is that for efficient calculation the whole volume has to be stored on the GPU or sophisticated methods for memory handling between CPU and GPU are required [80, 104]. Furthermore, since the computational effort is high, it is not well suited for applications like robotics or AR where computational power is often limited.

The second class of methods explicitly densifies the sparse point cloud directly in 3D space by triangulating additional 3D points and then interpolate a surface into these points. For the densification process a large number of approaches exists [7, 10, 22]. They differ in their computational effort and in their quality of the resulting point cloud. The PMVS approach of [22] for example requires several hours to generate a point cloud from a medium-sized dataset. However, the result is very accurate and is nearly free of outlier as shown in Figure 3.10(a). In contrast, planesweep methods like [7] can be efficiently implemented on the GPU but are also prone to outlier. The presence of outlier as well as the spatial distribution of the points is crucial for the subsequent surface extraction algorithm.

For regularly sampled point clouds that contain only a few outlier like that obtained from [22], the Poisson surface reconstruction [46] is well suited. This algorithm extracts the surface from a point cloud by computing an indicator function and then extracts an appropriate iso-surface. Since this method only rely on the scene points and their normals, this approach is not limited to the reconstruction of point clouds obtained by image-based methods but can be used for any kind of point clouds. However, since the indicator function is represented by an octree the memory consumption can be very high, if a detailed reconstruction is required. Furthermore, if the surface is not sampled homogeneously, the algorithm tends to produce bubble-like artifacts as shown in Figure 3.10(b).

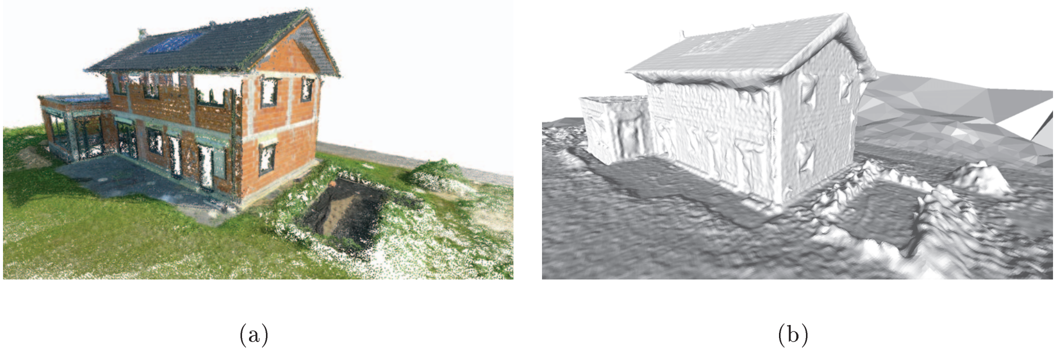


Figure 3.10: (a) PMVS point cloud [22] and (b) the corresponding surface extracted with the Poisson surface reconstruction method [46]. The Poisson reconstruction tends to create bubble-like structures if 3D points are missing on the surface.

The Poisson surface as well as the implicit surface extraction from a truncated signed distance function rely on a discretization into a regular voxel grid (the octree that is used by the Poisson reconstruction can be seen as a voxel discretization with varying resolution). In contrast the approach of Labatut et al. [56] relies on a discretization into tetrahedra by performing a DT on the triangulated points. Given the discretized volume, Labatut et al. formulate the surface extraction as a binary labeling problem into free- and occupied tetrahedra. The final surface is then the interface between differently labeled neighbors. To find the optimal labeling, they formulate the labeling problem as a conditional random field:

$$E(\mathcal{S}) = E_{vis}(\mathcal{S}) + \alpha_{photo} E_{photo}(\mathcal{S}) + \alpha_{area} E_{area}(\mathcal{S}), \tag{3.32}$$

where $\mathcal{S}$ is the actual binary label configuration for all tetrahedra. The visibility term $E_{vis}$ is related to the number of violated visibility constraints. Visibility constraints are given by the information which 3D point projects into which camera. $E_{photo}$ defines a photo consistency measure and $E_{area}$ favors compact surfaces. $\alpha_{photo}$ and $\alpha_{area}$ are factors to weight between the different terms. Since the overall energy is a binary submodular function, the globally optimal labeling can be efficiently found for example by graph cuts [53]. The authors showed that this approach works for quasi-dense point clouds that contain a significant amount of outlier.

The approach has several useful properties. Based on the adaptive and irregular discretization, this method is suited to extract surfaces from largely varying point densities as shown in Figure 3.11(b). On the left side of the tower, the point density is very high whereas on the right side only a few 3D points are triangulated on parts that are well textured. In contrast to the Poisson reconstruction that creates bubble-like artifacts, missing parts are linearly interpolated. For man-made scenes like buildings this is often more appropriate than the Poisson interpolation. Since the DT can be easily updated if new 3D points are available, it is suited to perform the discretization in an incremental manner. Furthermore, the memory consumption does not depend on the volume covered by the point cloud but only on the number of 3D points.

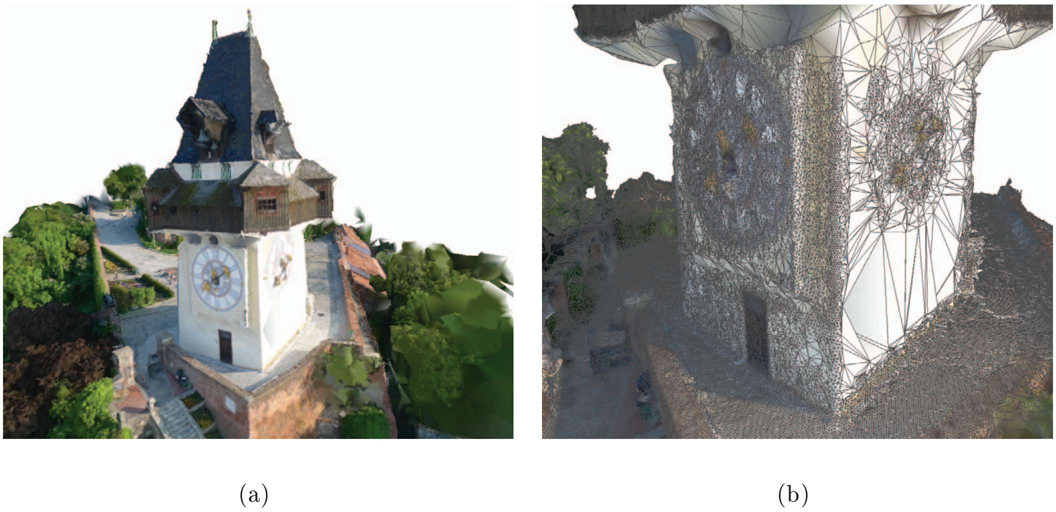


Figure 3.11: Example of graph cut based surface extraction proposed by [56]. The closeup in (b) shows that different densities are handled inherently. Missing data is approximated by planes which is appropriate especially in man-made environment.

### 3.7.3 Surface Reconstruction from Sparse Data

Surface extraction from sparse point clouds that are the result of a standard SfM pipeline has not gained much interest in the past. One reason is that due to the sparseness of the 3D points, the extracted surface has limited resolution and therefore it is thought that the additional knowledge is not very useful for many applications. But we will show later that many applications can benefit from even such a low-resolution surface.

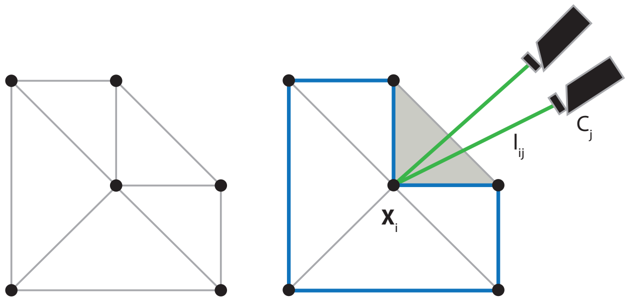
From an algorithmic point of view, the sparseness is also challenging. In case of dense data, robustness is achieved due to the enormous redundancy which is not available in the sparse point cloud. In this section, we give an overview over the related work and explain important methods for surface reconstruction from sparse data in more detail.

Basically all existing methods follow a two-step approach: first, the space is irregularly discretized into tetrahedra using the DT and second, all tetrahedra are classified into free- or occupied space. The interface is then the resulting surface. The existing methods differ from each other in their way to classify free- and occupied space. However, all methods use the visibility information associated with each triangulated 3D point for the classification.

In this context, the visibility constraint is defined as the set $\mathcal{L}$ of line segment $l_{ij}$ that connects 3D points $\mathbf{X}_i$ with camera centers $C_j$ they are visible in. Since it is known which triangulated scene point is originated by which image features, the visibility constraint $\mathcal{L}$ can be directly obtained from the sparse reconstruction. Figure 3.12(b) illustrates the visibility constraints in a toy example.

The simplest method for the classification is to use the visibility constraints of a scene point directly. Given a line segment $l_{ij} \in \mathcal{L}$, all volume elements intersected by $l_{ij}$ have to be free-space and all remaining tetrahedra are occupied space. This simple algorithm is also known as free-space carving [61]. The resulting surface is then constructed from the triangles that are at the interface between free- and occupied tetrahedra. The disadvantage is that this is not robust against outlier 3D points. Only a single outlier point that is located inside of an object causes a hole in the object's surface. Furthermore, the extracted surface is not a watertight manifold which is important for many applications like physical simulations.

Therefore, this basic approach has been improved over time. Pan et al. [77] focused on being robust against noisy point estimates that are close to the real surface. Instead of classifying a tetrahedron as free space if it is intersected by a single visibility constraint $l_{ij}$, they define for each triangle of the DT a probability score for being an element of the surface. The probability depends on the number of intersections as well as on the distance



(a) (b)

Figure 3.12: Principle of free-space carving. (a) Given the Delaunay triangulation of 6 points, we can extract the surface using visibility constraints. For each simplex of the DT it is tested if it violates a visibility constraint. The visibility constraints are given by the line segments that connect a 3D point with the camera centers it is triangulated of (green lines). If a simplex violates this constraint (gray triangle), it is classified as free-space otherwise it is occupied space (white triangles). Finally, the surface is defined as the set of faces between free- and occupied simplexes (blue lines).

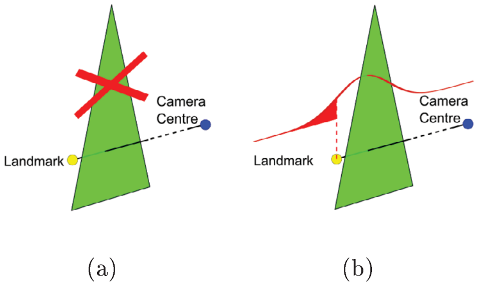


Figure 3.13: Probabilistic free-space carving of Pan et al. [77]. (a) Instead of removing a triangle if it is intersected by a visibility constraint, they also take the distance between the intersection point and the scene point into account. If the intersection is close to the scene point, the probability is high that the triangle belongs to the surface otherwise the probability is low. The function is modeled such that it compensates Gaussian noise of scene points. (Figure courtesy of [77])

between the intersected triangle and $\mathbf{X}_i$. If the intersected triangle is close to $\mathbf{X}_i$, the probability is high that the triangle is on the surface. Obviously, the larger the distance between an intersected triangle and the scene point $\mathbf{X}_i$ is, the lower is the probability that the triangle is on the surface. This probability is modeled by a Gaussian function to handle scene points that are perturbed by Gaussian noise. However, this method is not able to handle severe outlier that are located far from the real surface.

The aforementioned approaches for surface extraction are designed to work in a batch-based manner, i.e. it is assumed that all input data are given at the programs start. However, when thinking of applications like Simultaneous Localization and Mapping (SLAM) where new 3D points are created constantly, these methods are not suited to integrate the new obtained scene information. Hence, for applications where new scene information comes available, surface extraction algorithms have to be designed such that they integrate the new data in an incremental manner.

Lovi et al. [61] adopted the free-space carving approach to work in an incremental manner in a SLAM framework. In the context of tetrahedral space carving two steps have to be adopted for an incremental method: the DT and the classification. As mentioned in Section 3.7.1, there exist DT methods that inherently work in an incremental manner. Every time a new 3D point is integrated into the DT, new tetrahedra are created which subsequently have to be classified. In case of the free-space carving this is the crucial point, because all new created tetrahedra have to be tested if they violate any visibility constraint

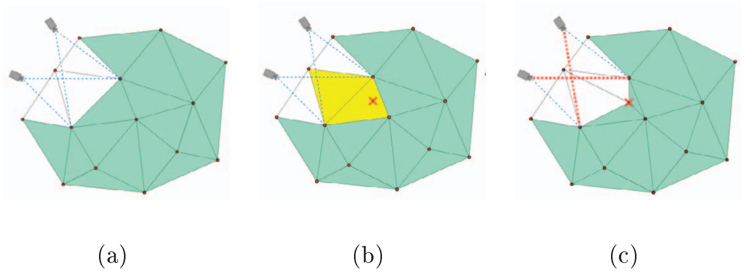


Figure 3.14: Incremental free-space carving of Lovi et al. [61]. (a) Initial state of the triangulation. (b) Integration of a new point (red cross) invalidates the DT and the yellow marked triangles are destroyed. (c) The new created tetrahedra are again tested against the visibility constraints (red dashed lines). Visualization of [61].

$l_{ij} \in \mathcal{L}$. Hence, in a naive implementation this algorithm grows at least linear in the number of visibility constraints and is computationally demanding if $|\mathcal{L}|$ is large. To overcome this problem, Lovi et al. propose to store in each tetrahedron the visibility constraints that intersect this volume element. If this volume element is destroyed due to the insertion of a new scene point, the new created tetrahedra only have to be tested against line segments that passed the destroyed tetrahedron. A visualization of the procedure is shown in Figure 3.14. This approach reduces the number of intersection checks but increases the memory consumption drastically. Furthermore, the computational complexity for inserting a new point may vary drastically. Due to the adaptive triangulation, sometimes very large volume elements are generated that are passed by a very large number or even all line segments. If this tetrahedron is destroyed, new created tetrahedra have to be tested against all line segments. To bypass this problem, Lovi et al. propose a heuristic to store only a selected number of visibility constraints per tetrahedron. However, the basic problem that free-space carving is not robust against outlier still exists.

Yu et al. [106] propose a different approach for incremental surface reconstruction. Instead of carving tetrahedra that are classified as free-space, they aggregate them. The classification into free- and occupied space is performed by checking if a tetrahedron violates the visibility constraint. The border of the set of free-space tetrahedra is then the surface. Since this results in an arbitrary surface shape which is often not a 2-manifold, they implement methods to guarantee that the extracted surface maintains this property. Their method is incremental in that way that they are able to add new 3D points in scene parts where no surface has been extracted before. Hence, this algorithm is not able to refine existing parts of the mesh which restricts the method to strictly forward camera motion. Litvinov et al. [59] extended this approach to also allow the refinement of existing surface parts. The method itself is computational as well as from an implementation point-of-view quite complex and it is not clear how robust this approach against outlier is.

## 3.8 Conclusion

In this chapter, we provided the background of image-based scene reconstruction. We first described the geometric relations of single-view, two-view and multi-view reconstructions and gave an exemplary description of a state-of-the-art Structure-from-Motion reconstruction pipeline. In the second part, we reviewed the related work in the field of surface extraction from SfM results. We described the advantages and disadvantages of different volume discretization methods and presented state-of-the-art algorithms for surface extraction from sparse as well as from dense visual information.

To sum up, the research on automatic orientation of a large number of unordered images has reached maturity and therefore, several methods like image-based localization and surface reconstruction methods that are based on SfM results have been developed. Nevertheless, the number of applications that are realized by SfM is rather low.

# Chapter 4

# Real-time and Interactive Structure-from-Motion

The properties of an image dataset to obtain an accurate, fully connected and complete reconstruction are versatile. The main influencing parameters are the degree of redundancy, the spatial distribution of view points over the area of interest, the relative orientation between view points and the texture of the scene. The acquisition of an input dataset that fulfills all parameters such that the expected reconstruction quality is obtained, is quite difficult for several reasons. The first issue is that the photographer has to be aware of these parameters. A non-expert user in 3D reconstruction is often not able to acquire a suitable image set because he has no knowledge about the important parameters. Even if the user gets an intensive briefing, we observed that the resulting image sets are not well-suited for the reconstruction because (a) the user underestimates the required redundancy and (b) the importance of the relative orientation between the images to allow automatic feature matching. However, even if the user is an expert in SfM, it is not guaranteed that he acquires a well-suited image set because even for an expert it is difficult to ensure redundancy and completeness in large-scale and geometrically complex scenes. Beside the well-known theoretical requirements like overlap and triangulation angles, many practical problems arise from implementation details of the SfM pipeline used e.g. the applied feature descriptor, the way how images are integrated into the reconstruction and so on. Because the number of factors that influence the reconstruction result is so large, it is very difficult to define abstract rules for a photographer how to acquire a suited image dataset.

Our idea to tackle this problem is to integrate the acquisition process directly into the SfM pipeline. Instead of first acquiring a set of images and then processing the whole dataset in batch-based manner, we propose a closed-loop interactive SfM processing that performs an incremental reconstruction in real-time. Our closed-loop approach first determines the camera pose of a new acquired image, updates the sparse point cloud and then extracts a surface mesh from the triangulated image features. This allows us to compute quality parameters like the Ground Sampling Distance (GSD) and the image overlap which are then visualized directly on the surface mesh.

The advantages of the interactive SfM are versatile. First, the user instantly gets feedback if an image can be integrated into the reconstruction. This decreases the risk that in a final batch-processing the reconstruction fragemnts into different parts. Second, the representation of the reconstruction as a surface mesh eases the interpretation of the SfM result also for non-expert users. And finally, the visualization of quality parameters supports users to ensure that all relevant parts of the environment have been captured with enough overlap and with the desired resolution.

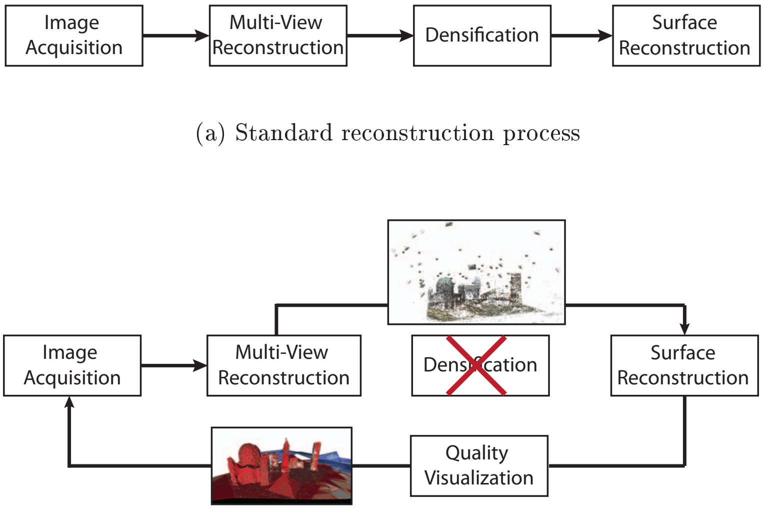
In order to realize the closed-loop schema as it is visualized in Figure 4.1(b), all components have to work (a) in a fully incremental manner and (b) in real-time. Therefore, we developed a novel incremental sparse reconstruction method and a fully incremental surface meshing method which are then integrated into a full real-time system. In the next section, we outline our methods in detail.

## 4.1 Method Overview

As shown in Figure 4.1(a), a standard SfM pipeline typically consists of four steps: image acquisition, multi-view sparse reconstruction, densification and finally a surface meshing. In order to give an instant feedback about the contribution of a new image, we propose a modified processing pipeline as shown in Figure 4.1(b). Given a freshly acquired image, we first perform SfM incrementally, i. e. we calculate the pose of the new image and expand the sparse scene reconstruction. However, assessing the reconstruction's quality by using the sparse reconstruction only, can be difficult even for an experienced user. A representation that is easier to understand is a surface mesh. Since most existing methods for a surface extraction rely on a dense 3D point cloud the is computationally expensive, our goal is to skip the densification and perform a meshing directly on the sparse points. Therefore, we propose a new surface extraction method that is able to cope with sparse and inhomogeneous point data in a fully incremental manner. Finally, we visualize parameters of the image dataset like redundancy and resolution on the extracted surface. The accuracy of the resulting sparse orientation is sufficient that a standard densification method like PMVS [22] can be performed afterwards. In particular, our contributions are:

1. Real-time SfM with efficient image-based localization.
2. Incremental surface reconstruction using sparse point clouds.
3. Visualization of quality parameters of the current reconstruction.
4. Integration of all methods into a common application.

The remainder of this chapter is structured as follows. In Section 4.2, we propose our incremental SfM framework and present our novel efficient image-based localization



(a) Standard reconstruction process

(b) Our modified SfM pipeline for optimized image acquisition

Figure 4.1: Classical and interactive reconstruction pipelines. (a) The classical processing is a pure feedforward method. Image acquisition is decoupled from the subsequent processing. (b) In our proposed interactive reconstruction method, we combine an on-line multi-view approach with a surface reconstruction from sparse point clouds and provide feedback to the user by calculating important scene parameters. The whole processing requires only a second to integrate a new image into the reconstruction.

method. Section 4.3 describes the incremental surface extraction algorithm. In Section 4.4, we demonstrate how the surface mesh can be used to visualize quality information like Ground Sampling Distance and image overlap. Since the integration of all parts into a real-time system is not straight forward, implementation details are outlined in Section 4.5. Finally, the evaluation of all individual contributions and the overall method is given in Section 4.6.

## 4.2 Real-time SfM with Efficient Image-based Localization

Many batch-based SfM approaches assume spatially unordered images as input and therefore require up to hours to determine the spatial ordering by constructing an epipolar graph [89]. In order to integrate a new image in real-time, we have to solve the SfM problem in an incremental fashion. This is closely related to SLAM systems [47] but in contrast, we do not work on a continuous video stream but on high resolution still images.

Since the construction of the epipolar graph comprises the calculation of relative orientations between all image pairs, this is the most time-consuming task in batch SfM pipelines. In our addressed problem, we can assume that a user does not acquire images in a totally random order. If we assume that a new input image $I$ has an overlap to an already reconstructed scene part, we can skip the epipolar graph construction and the SfM problem can be split into two tasks that are easier to solve: a localization and a structure expansion part. More formally, given a freshly acquired input image $I$ and a reconstructed scene $M$, we determine the position of $I$ within $M$ and finally, we expand the map $M$. For bootstrapping the scene $M$, we rely on initialization schemes that are also used in batch-based SfM methods. In the following sections, we describe the bootstrapping as well as the expansion of the reconstruction if the pose of the image $I$ is known. Finally, we present our new image-based localization approach to determine the pose of $I$ in a computationally efficient manner.

### 4.2.1 Structure Initialization and Expansion

For bootstrapping the initial map $M$, we require two images taken from different viewpoints and perform brute-force feature matching. The Five-Point pose estimation algorithm of Nister [74] in a RANSAC [20] loop is used to find the relative orientation between both cameras. Using the inlier correspondences returned by the RANSAC, we triangulate the initial set of 3D points. Since the initialization is crucial for the whole subsequent process, we have to ensure that the initial model is consistent. Therefore, we require that the triangulation angle of the triangulated features exceed a minimum threshold and that the initial map consists of a minimum number of features. If this cannot be achieved, the user is asked to take new images until an image pair is found that fulfills the requirements.

For the expansion of the structure, we assume that the pose of a new $I$ image with respect to the map $M$ is calculated by an image-based localization approach as described in the subsequent Section 4.2.2. Since image correspondences between $I$ and other reconstructed images are already determined within the localization step and the pose of $I$ is known, we can easily triangulate image features to create new 3D points. To discard erroneous correspondences before triangulation, we perform an epipolar consistency check, i.e. we calculate the distance of a corresponding feature pair $x \leftrightarrow x'$ to their epipolar line in the corresponding image.

If the image cannot be localized instantly, the user can manually trigger the localization of this image at a later point in time. Alternatively, this can be performed by a background process.

To prevent scene drift caused by incremental map building, we use a global optimization scheme to obtain a consistent map. Hence, we perform iterative bundle adjustment [93] in a parallel thread.

### 4.2.2 Efficient Image-based Localization

The problem of determining the pose of $I$ within $M$ is an image-based localization problem. Because our aim is to provide feedback to the user if the image can be integrated as fast as possible, the computational complexity is an issue. Most research on image-based localization focuses on the localization of an image within a reconstruction containing tens of thousands of images [42, 57, 83, 85]. The goal of these approaches is to be scalable to ultra-large databases while being computational efficient. Our application has different requirements. We typically deal with a much lower number of images but computational complexity and response times are a major issue. The same requirements are also important in AR applications where video streams are localized [4] or in autonomous robotics applications [58, 102]. Therefore, we propose a new image-based localization method that is much faster than existing methods.

The reduction in complexity is motivated by the insight of state-of-the-art methods like [42] and [83] that showed that only a few correct 2D–3D correspondences are sufficient to recover a valid pose. Based on this observation, the question arises if matching has to be performed for all $n$ features of the query image or if it is possible to a-priori find a subset of features that lead to a valid pose with high probability. We will show that considering only a subset of query features according to their keypoint scale information can reduce the computational costs significantly. Starting with features connected to keypoints at a large scale, we iteratively consider a larger number of query features by adding more and more features extracted on increasingly finer details. Consequently, this leads to a localization approach based on a hierarchical principle. Since testing all scales takes twice as much time than using only the largest scale, our localization method has to recover the valid pose on coarse scale features with high probability.

We performed an initial experiment (see Section 4.6.1) to investigate which of the the two state-of-the-art approaches (brute-force or image-retrieval) is able to handle low-resolution images more reliable. Based on the result that the image-retrieval method is better suited in such a low-resolution scenario, we propose three contributions to increase the localization quality. First, we improve the image-retrieval ranking in case that the resolution of query and database images is very different. Second, we reduce the 2D–3D matching effort by an order of magnitude when localizing low-resolution images by taking the keypoint size into account. Finally, we propose a RANSAC pre-verification test that discards wrong hypotheses in an early stage.

#### 4.2.2.1 Improved Image-Ranking

Basically, the ranking process of image retrieval (IR) depends on a (robust) comparison between bag-of-words histograms of database and query images [76]. To get reasonable results, scoring methods like term frequency–inverse document frequency (TF-IDF) implicitly require that the number of features between a database image $D_i$ and $Q$ is in the same range. When considering situations where $D_i$ and $Q$ have very different resolutions and therefore the number of features varies drastically, the histogram shapes are also much different which has a large impact on the ranking quality as we show in the experiments.

To increase the ranking quality, Irschara et al. [42] proposed a scoring method called *probabilistic scoring* that models the voting process as a binomial stochastic process. They, as well as [84], have shown that for a localization scenario, probabilistic scoring achieves better results than TF-IDF scoring. Therefore, we shortly describe the idea of probabilistic scoring to outline the emerging problem when comparing images of different resolutions.

Let $f_i^Q$ be a feature of the query image $Q$ and $f_j^{D_i}$ a feature of a database image $D_i$. The probability $p_1$ that $f_i^Q$ and $f_j^{D_i}$ are equal and fall into the same leaf node is assumed to be relatively high and to have a constant value. In contrast, the probability that $f_i^Q$ and $f_j^{D_i}$ are not the same landmark but fall in the same leaf depends on the number of leaf nodes that are occupied by $D_i$. This can be calculated by

$$\bar{p}(D_i) := \frac{\#D_i}{\#leaves}\,, \tag{4.1}$$

where $\#D_i$ is the number of leaf nodes where features of $D_i$ are located in and $\#leaves$ is the number of overall leaves in the vocabulary tree. With these probabilities, we can calculate the chance of getting $k$ votes for a document $D_i$ if $Q \equiv D_i$ as a binomial probability function

$$k \sim B(|Q|, p_1) \text{ if } Q \equiv D_i\,, \tag{4.2}$$

where $|Q|$ is the number of features in the query image. The probability for getting $k$ votes if $Q \not\equiv D_i$ is modeled as

$$k \sim B(|Q|, \bar{p}(D_i)) \text{ if } Q \not\equiv D_i\,. \tag{4.3}$$

To obtain a score for each document given the raw votes, the score is derived from the posterior probability by Bayes' rule:

$$\frac{P(\#votes = k|Q \equiv D_i)}{P(\#votes = k|Q \not\equiv D_i)}\,. \tag{4.4}$$

In practice, the log-value of the posterior probability is used as the scoring value.

We found that the ranking quality degrades if the number of query features is small. This is caused by the assumption of Equation 4.3 that features vote for unrelated documents uniformly. The larger the number of query features, the more likely it is that this assumption holds. In contrast, when the number of query features is small this assumption is often violated. We also recognized that an unbalanced number of features between query and database intensifies this problem. To reduce the imbalance, one could either down-sample the database images to the size of the query image or adapt the sensitivity of the keypoint detector such that query and database images are represented by the same number of features. However, this requires that the resolution of the query image has to be known in advance and in a multi-scale approach, multiple vocabulary trees for each scale are required, which increases memory consumption.

A general solution to get better rankings is to reduce the number of votes for false documents, for example by an additional feature matching step to all features within the leaf node, as proposed by Sattler et al. [84]. However, we follow the idea of Jegou et al. [43] to consider the keypoint scale information for reducing the number of false votes. The scale reports the size of a landmark in image space. A high-resolution image typically contains a large number of fine-detailed structures that are not visible in its low-resolution counterpart. For the voting process, these fine details are problematic, because they increase the number of votes for a particular document although such a fine detailed feature could not have been extracted on the low-resolution image. Hence, our idea is to vote only for documents whose features also could have been extracted in the corresponding low-resolution query image. For that purpose we make use of the scale information provided by the keypoint detector.

A keypoint detector like the Difference of Gaussian (DoG) as used in SIFT reports the layer of the scale-space pyramid where the maximum response is detected. The layer depends on the size of the landmark and the focal length of the camera. To make feature scales between images acquired by different cameras and at different resolutions comparable, we derive following normalization.

Given the 2D scale $s$ reported by the keypoint detector and the focal length of the camera $c_1$ we can reconstruct the feature size $L$ of the landmark by

$$L = \frac{d\, s_1(f)}{c_1}\,, \tag{4.5}$$

where $d$ is the distance between the image plane and the landmark $L$. If this landmark is pictured by a second camera with focal length $c_2$ having the same distance $d$ to the landmark as $c_1$, the feature size $s_2$ can be computed by

$$s_2(f) = \frac{L\, c_2}{d}\,. \tag{4.6}$$

Hence

$$\frac{s_1(f)}{c_1} = \frac{s_2(f)}{c_2} \tag{4.7}$$

holds for cameras with different focal lengths $c_1$ and $c_2$. Therefore, we propose to normalize the feature scale $\bar{s}(f)$ of a feature $f$ by

$$\bar{s}(f) = \frac{s(f)}{c} \quad , \tag{4.8}$$

where $s(f)$ is the scale of the feature reported by the keypoint detector and $c$ the focal length of the camera. In many cases, the focal length can be read out from the EXIF data stored along with the image or in case of robotics or augmented reality applications often calibrated cameras are used. Figure 4.2 illustrates the relation between pictured scale and focal length.

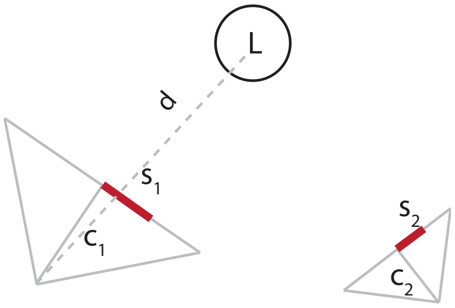


Figure 4.2: Feature scale normalization. Two different cameras with focal length $c_1$ and $c_2$ having the same distance $d$ to the landmark $L$. The pictured scales $s_1$ and $s_2$ if $L$ are related to each other according to Equation 4.7

Under the assumption that $Q$ and $D_i$ are taken at the same position, corresponding features have the same normalized scale. With this condition, we could reduce the number of false votes during scoring by excluding all database features whose scale differs from the query feature. But this assumption is too strong, because in many scenarios the position of $Q$ and $D_i$ is not identical. Therefore, we weaken this assumption and assume that the position of $Q$ is located between the scene structure and a certain distance behind $D_i$. This leads to the fact that the scale of all corresponding features of $Q$ and $D_i$ are larger or equal in $Q$. Consequently, we can ignore all features of $D_i$ that are smaller than the smallest feature of $Q$. Therefore, we can exclude the fine details of $D_i$ that could not have been extracted in $Q$ which removes a large number of spurious features if the resolution of

$Q$ is much higher than $D_i$.

More formally, we define the smallest features $min_Q = min_{\bar{s}(f)}(Q)$ of $Q$ and $min_{D_i} = min_{\bar{s}(f)}(D_i)$ of $D_i$ and take the maximum of both by

$$m(Q, D_i) = \alpha * max(min_Q, min_{D_i})\,, \tag{4.9}$$

where $\alpha < 1$ is a weighting parameter. This parameter indirectly defines the maximum distance that $Q$ is allowed to be located behind $D_i$ with respect to the scene structure. We fixed $\alpha = 0.5$ for all datasets used in the experimental evaluation. To ignore all features that are smaller than $m(Q, D_i)$ in the scoring process, the sets $Q$ and $D_i$ of Equations 4.1-4.4 are replaced by new sets $\bar{Q}$ and $\bar{D}_i$ that contain all features larger $m(Q, D_i)$ defined as

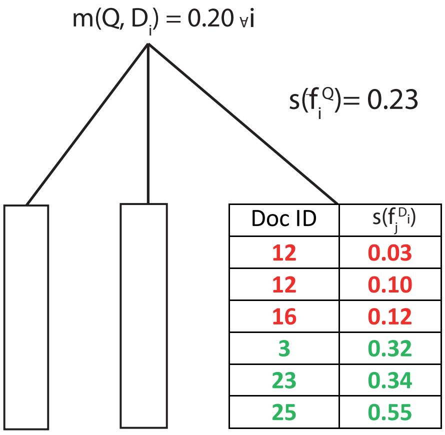
$$\bar{Q} = \bigcup_i f_i^Q \;\text{ where }\; \bar{s}(f_i^Q) > m(Q, D_i) \;\text{ and} \tag{4.10}$$

$$\bar{D}_i = \bigcup_j f_j^{D_i} \;\text{ where }\; \bar{s}(f_j^{D_i}) > m(Q, D_i)\,. \tag{4.11}$$

If $\alpha$ is set to zero, $\bar{Q}$ and $\bar{D}_i$ are identical to $Q$ and $D_i$ and the voting turns into the original *probabilistic scoring*.

For implementing this scoring approach, we do not have to build $\bar{Q}$ and $\bar{D}_i$ explicitly. Equation 4.2 and 4.3 require only $|\bar{Q}|$ which is achieved by simply counting all features that are larger than $m(Q, D_i)$. This can be efficiently computed using sorted lists. Equation 4.1 counts the occupied leaf nodes which depends on $m(Q, D_i)$. This can be pre-computed for different discretized steps of $m(Q, D_i)$. For scoring, we use only features of $D_i$ whose scale is larger than $m(Q, D_i)$. This requires that for each database feature its scale $\bar{s}(f_j^{D_i})$ is stored within the leaf node. The additional memory overhead is small, since $\bar{s}$ is a single floating number. The computational overhead for the scoring is negligible since we only have to compare if a feature scale $\bar{s}(f_j^{D_i})$ in a certain leaf node is larger than $m(Q, D_i)$. This can also be optimized by storing the features in each leaf node in a sorted list. Figure 4.3 illustrates the scoring process.

We also experimented with other criteria for voting for a document for example by applying a tophat function to the ratio between $s(f_i^Q)$ and $s(f_j^{D_i})$. We found that the results were similar to the proposed method but are more restrictive since this limits the query to lie in a circle around the database image.



| Doc ID | $s(f_j^{D_i})$ |
|---|---|
| 12 | 0.03 |
| 12 | 0.10 |
| 16 | 0.12 |
| 3 | 0.32 |
| 23 | 0.34 |
| 25 | 0.55 |

Figure 4.3: Illustration of scale dependent feature selection. Given a query feature with normalized scale $\bar{s}(f_i^q)$ that falls into the third leaf node votes for documents 3, 23 and 25. 12 and 16 are ignored since their scale is smaller than $m(Q, D_i) = 0.20 \forall i$.

Our scoring formulation is not restricted to tasks where the query is smaller than the database image. Furthermore, the relevant features for the scoring are determined for each image pair $(Q, D_i)$ individually and can be efficiently computed during the query. Therefore, our ranking supports databases of inhomogeneous resolutions and requires no additional information when creating the vocabulary tree.

Finally, after obtaining the shortlist $S$ of visually similar images by applying the described image retrieval approach, we aim at establishing 2D-3D correspondences by feature matching as it is described in the next section.

#### 4.2.2.2 Reduced 2D–3D Correspondence Matching

For finding the absolute pose of $Q$, we match its features against $S_i$ where $S_i$ is a single image of $S$. Since we want to establish 2D-3D correspondences, we are only interested in features of $S_i$ that are connected to a 3D point. A match is established if the SIFT distance ratio test passes 0.7. This measure enforces that only features are matched that are discriminative within the feature set of $S_i$. Typically, the more features contained in $S_i$, the lower is the number of matches. If $S_i$ is a high-resolution image, it also contains fine-detailed features that could not have been extracted in $Q$ due to scale difference. With the same argument as in the image-retrieval step, we can remove the fine-detailed features. This has two positive effects: First, the number of features for matching between $(Q, S_i)$ is lower and therefore it is faster. Second, the chance of missing correct matches is lower due to the smaller number of features. We take only features into account whose normalized scale $\bar{s}(f)$ is larger than $m(Q, S_i)$. We demonstrate in the experiment that this reduces the matching effort significantly while increasing the number of localized low-resolution frames.

#### 4.2.2.3 RANSAC Pre-Verification Test

To estimate the pose given the 2D–3D correspondences typically the absolute pose problem is solved within a robust hypothesize-and-verify algorithm like RANSAC. All these methods have in common that they randomly sample the minimum number of correspondences that is required to estimate the 6 degrees-of-freedom pose and then verify this pose using all other correspondences. To reduce the effort for hypothesis verification, we propose, related to the idea of [12], a pre-verification step which again is based on feature scale information. Note that our pre-verification step can be integrated into any hypothesize-and-verify algorithm.

Since each 3D point $P$ of the point cloud is connected to at least two features $f$ and the distance between the camera of $f$ and $P$ is known, we can reconstruct the size $L$ of the landmark according to Equation 4.5. By establishing a match of query feature $f_q$ with a point $P$, we estimate the distance $d_e$ of the query camera to $P$ using the feature scale $s(f_q)$. Next, we select three correspondences according to the sampling schema of the applied RANSAC variant and calculate a hypothesis $H$. If $H$ is a valid hypothesis, we expect that our estimated distance $d_e$ and the distance of $H$ to $P$, denoted $d_H$, are similar. Empirically we found that the expected distance $d_e$ and the distance $H$ to $P$ typically differs by less than 50% if $H$ is a valid hypothesis. Hence, if at least one of the three correspondences that were used to generate $H$ differ by more than 50%, we reject this hypothesis without verifying all correspondences. The difference between $d_e$ and $d_H$ is mainly caused by the inaccuracy of the keypoint detector. In the experiments in Section 4.6.1, we show that this simple pre-verification procedure reduces the number of required hypothesis verification steps by more than 80%.

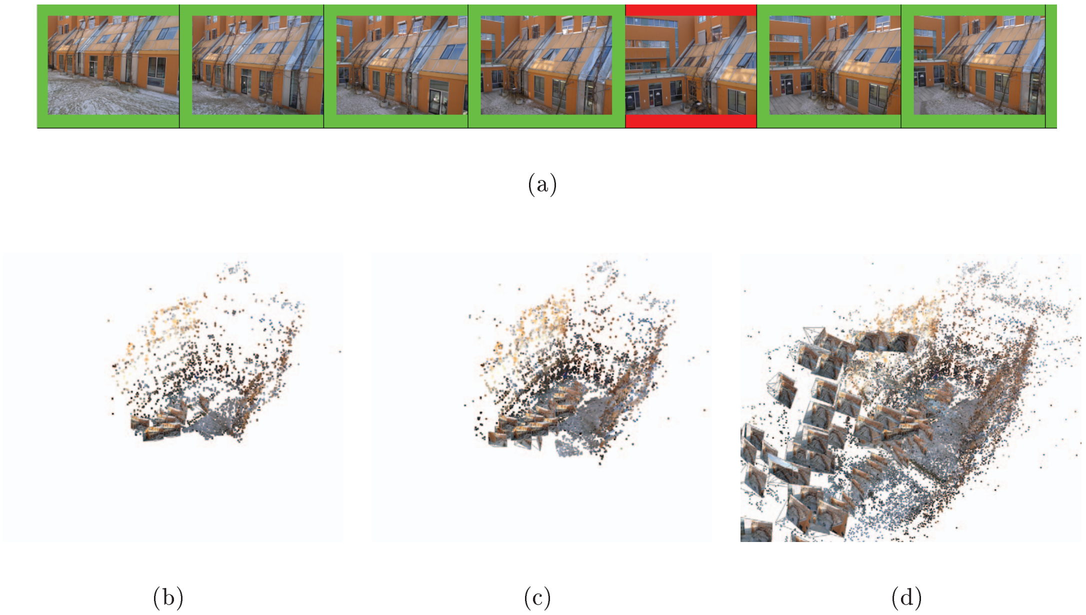


Figure 4.4: (a) Feedback for the user if the acquired images can be registered within the map $M$. Red indicates that the image was not registered. A green border marks a successful localization. The algorithm is not able to register the red marked image although it is visually similar to its neighboring images. This shows who difficult it is for a user to predict if an image is usable for the reconstruction. (b)-(d) Sparse reconstruction at different points in time obtained by on-line SfM.

### 4.2.3 Discussion of Real-time SfM

The incremental real-time SfM makes the image acquisition for batch-based SfM more reliable in two ways. First, the user gets a feedback within a second if the new acquired image can be registered within the previously reconstructed map $M$. Hence, the user instantly discovers if the new image fulfills fundamental parameters like overlap and relative camera position that allows feature matching. For an unexperienced user the notification if the image could have been localized or not is very helpful to intuitively learn about the spatial relation between successfully localized images.

Furthermore, the real-time SfM can be almost completely realized with methods from standard batch-based methods. Hence, important parts like feature extraction and matching and relative pose estimation whose results often depend on the actual implementation can be shared. This is not only an advantage in the implementation but this also guarantees that the batch-based reconstruction will also be successful.

The second advantage of the real-time SfM is that the sparse reconstruction can be visualized directly to the user. Therefore, an experienced user can evaluate the completeness of the reconstruction on site and missing parts can be identified. Hence, these parts can be easily captured by additional images. Figure 4.4 illustrates the feedback information that is given to the user.

In the next section, we present our novel surface extraction method that operates on the sparse point clouds directly in a fully incremental manner.

## 4.3 Incremental Surface Reconstruction from Sparse Points

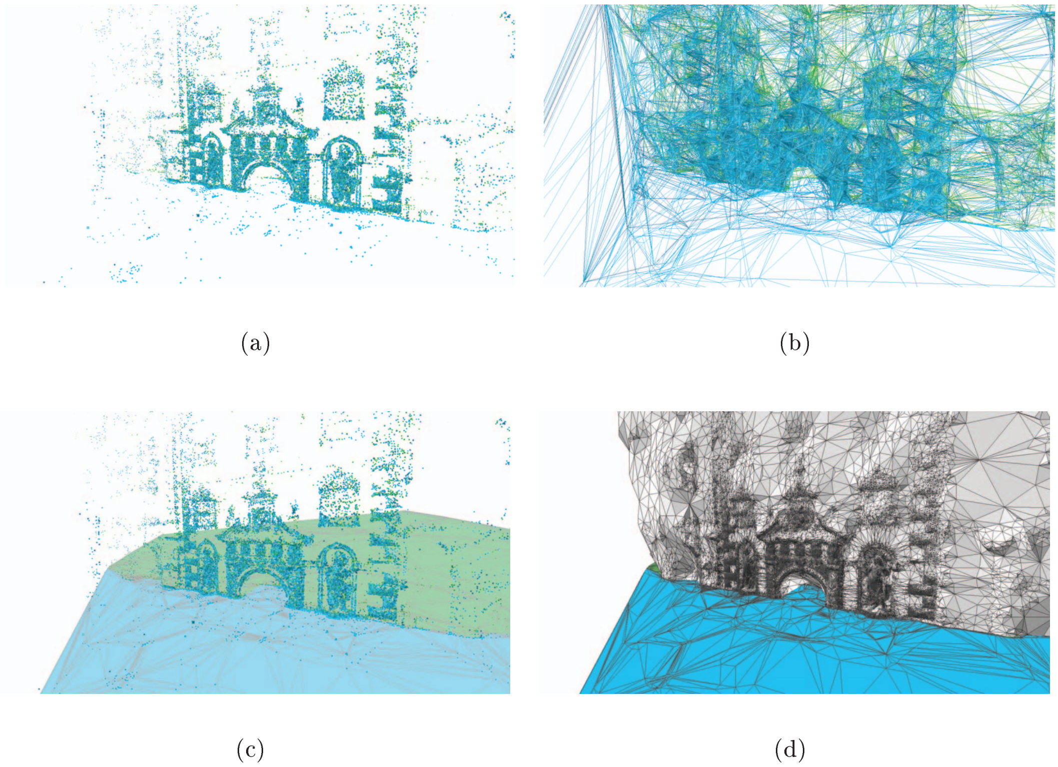


Figure 4.5: Workflow of the surface reconstruction method. (a) Starting with a sparse point cloud, a 3D Delaunay triangulation is performed which is shown in (b). The labeling of the tetrahedra results in a classification of free- or occupied space. Figure (c) shows a slice through the labeled tetrahedra where green is occupied space and blue is free space. The final surface (d) is than extracted as the set of triangles that are at the interface of free- and occupied tetrahedra.

The extraction of a surface from a sparse point cloud is not only important for an intuitive visualization of the current reconstruction quality but it is also beneficial for other application areas like occlusion handling in AR or navigation tasks in robotics.

Extracting surfaces from the 3D point cloud that is obtained from image-based reconstruction is a complex problem because the density of the points is highly irregular and perturbed by outliers. Existing solutions often either assume a densely, regularly sampled surface [46] or make use of additional knowledge like visibility information [56]. The information content of meshes that are extracted by the latter mentioned approaches is higher than using 3D points only, because the visibility information is also exploited.

The problem gets even harder if the surface has to be extracted from a continuously growing point cloud as generated by our real-time SfM. The same problem also occurs in Simultaneous Localization and Mapping (SLAM) methods [15, 18], where additional scene information is continuously provided. Such methods have to handle an increasing amount of data in real-time, which means that several hundred points have to be integrated into the surface per second. The state-of-the-art methods for incremental surface reconstruction such as [26, 72] make heavy use of powerful GPGPU units which are often not available in application areas like robotics or AR. Furthermore, these approaches represent the scene in an equally discretized voxel space and consequently, they are restricted to a limited scene size.

Therefore, we propose a new method to incrementally extract a surface from a continuously growing SfM point cloud in real-time. Our method is based on the Delaunay Triangulation of the 3D points. The core idea is to robustly label the tetrahedra into free- and occupied space using a random field formulation and to extract the surface as the interface between differently labeled tetrahedra. Therefore, we propose a new energy function that achieves the same accuracy as state-of-the-art methods but reduces the computational effort significantly. Furthermore, our new formulation allows us to extract the surface in an incremental manner, i. e. whenever the point cloud is updated, we adapt our energy function. Instead of minimizing the updated energy with a standard graph cut, we employ the dynamic graph cut of Kohli et al. [52] which allows an efficient minimization of a series of *similar* random fields by re-using the previous solution. The combination of the dynamic graph cut with our new formulation allows us to extract the surface from a continuously growing point cloud nearly independent of the overall scene size.

### 4.3.1 Energy Function for Surface Extraction

Our method for extracting a surface from a sparse SfM point cloud is motivated by the truncated signed distance function (TSDF), which is known from voxel-based surface reconstructions like [26, 107]. The TSDF models for all voxels along the ray connecting the camera and a 3D point $X$ their probability of being free space or occupied. Typically, this information is aggregated for a large number of 3D points obtained by several dense depth maps, where the resulting surface is then extracted as the zero crossing within the volume exploiting inherent redundancy.

By contrast, when using sparse points as in our intended application field, redundancy is limited and an extraction of the surface by finding the zero crossing is not possible. Therefore, our main idea is that given the tetrahedralized point cloud, we formulate surface extraction as a binary labeling problem, with the goal of assigning each tetrahedron either a *free*- or *occupied* label. For this reason, we model the probabilities that a tetrahedron is free- or occupied space analyzing the entire available visibility information $\mathcal{R}$, which consists of the set of rays that connect all 3D points to image features. Following the idea of the TSDF, a tetrahedron in front of a point $X$ has a high probability to be free space, whereas the tetrahedron behind $X$ is presumably occupied space. We further assume that it is very unlikely that neighboring tetrahedra obtain different labels, except for tetrahedra close to a point $X$. Such a labeling problem can be elegantly formulated as a pairwise random field.

Formally, given a set of tetrahedra $V$ obtained by the Delaunay Triangulation (DT) of the point cloud, we define a random field where the random variables are the tetrahedra of $V$. Our goal is to identify the binary labels $\mathcal{L}$ that give the maximum a posteriori (MAP) solution for our random field, analyzing the provided visibility information $\mathcal{R}$. The binary labels specify if a certain tetrahedron $V_i \in V$ is free- or occupied space. To identify the optimal labels $\mathcal{L}$, we define a standard pairwise energy function

$$E(\mathcal{L}) = \quad \textstyle\sum_i (E_u(V_i, \mathcal{R}_i) + \sum_{j \in \mathcal{N}_i} E_b(V_i, V_j, \mathcal{R}_i)) \, , \tag{4.12}$$

where $\mathcal{N}_i$ is the set of the four neighboring tetrahedra of the tetrahedron $V_i$ and $\mathcal{R}_i$ is a subset of $\mathcal{R}$, consisting of all rays connected to the vertices that span $V_i$.

For defining the unary costs $E_u(V_i, \mathcal{R}_i)$, we follow the idea of the TSDF that the

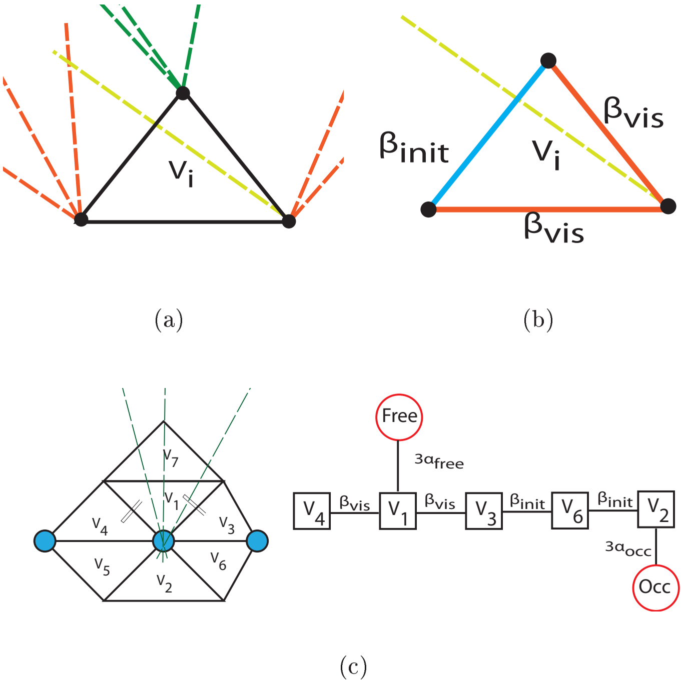


Figure 4.6: (a) For defining the unary term for a specific tetrahedron $V_i$ we only analyze rays (dashed lines) connected to vertices that span $V_i$. (b) For the pairwise term we only consider rays that pass through the tetrahedron and that are connected to the vertices of the tetrahedron. (c) Graph representation of the energy function. The pairwise weights that are not shown are set to $\beta_{init}$.

probability that a certain tetrahedron $V_i$ is free space is high, if many rays of $\mathcal{R}_i$ pass through $V_i$. Therefore, we set costs for labeling $V_i$ as occupied space to $n_f \alpha_{free}$, where $n_f$ is the number of rays of $\mathcal{R}_i$ that pass through $V_i$. In contrast if $V_i$ is located in extension of many rays of $\mathcal{R}_i$ the probability is high that $V_i$ is occupied space. For this reason, the costs for labeling $V_i$ as free space are set to $n_o \alpha_{occ}$, where $n_o$ is the number of rays in front of $V_i$. Figure 4.6(a) illustrates the unary costs for a small example. Here, $n_f$ is 1 since only the light green ray passes $V_i$ and $n_o$ is 3 because $V_i$ is in extension of the three green rays. The red rays do not contribute to the unary costs.

For the pairwise terms we assume that it is very unlikely that neighboring tetrahedra obtain different labels, except for pairs $(V_i, V_j)$ that have a ray through the triangle connecting both. Let $R_k$ be a ray of $\mathcal{R}_i$ that passes $V_i$. If $R_k$ intersects the triangle $(V_i, V_j)$, $E_b(V_i, V_j, \mathcal{R}_i)$ is set to $\beta_{vis}$. Triangles $(V_i, V_j)$ that are not intersected by any ray of $\mathcal{R}_i$ are set to $\beta_{init}$. Figure 4.6(b) shows the pairwise costs in an example and Figure 4.6(c) visualizes the graphical model of the energy function for a small example.

Since $V_1$ is passed by three rays, the costs for labeling $V_1$ *free* is set to $3\alpha_{free}$. In contrast, $V_2$ is in extension of three rays and therefore $V_2$ is connected to the *occupied* node with the weight $3\alpha_{occ}$. The edge weights between all neighboring tetrahedra are set to $\beta_{init}$ except the edges $(V_1, V_4)$ and $(V_1, V_3)$ which are set to $\beta_{vis}$.

Having defined all terms for our random field formulation, we are then able to derive a globally optimal labeling solution for our surface extraction problem using standard graph cuts since our energy function is submodular.

At a first glance, our energy seems to be similar to the visibility part of the energy defined by Labatut et al. [56]. The major difference is the definition of the pairwise costs which has a large impact on the computational complexity when adapting the energy to a new DT structure. Labatut et al. initialize the pairwise costs with a low value and increase the costs if an arbitrary ray $R_n \in R$ intersects $V_i$ as well as $V_j$, i. e. if $R_n$ intersects the triangle between $V_i$ and $V_j$. Therefore, the pairwise costs are not restricted to local visibility around $V_i$ but may depend on the global distribution of the rays. This might drastically increase the computational complexity for updating the energy to a new DT structure, although only a small part of the DT has changed as we demonstrate in the experiments in Section 4.6.2.

### 4.3.2 Incremental Surface Extraction

To enable an efficient incremental surface reconstruction, our method has to consecutively integrate new scene information (3D points as well as visibility information) in the energy function and to repeatedly find the optimal labeling. In this section, we first show how the energy terms are updated and second, how the modified optimization problem can be efficiently solved in an incremental manner using the dynamic graph cut.

#### 4.3.2.1 Energy Update

The energy function $E(\mathcal{L})$ depends on the structure of the DT and the visibility information $\mathcal{R}$ and therefore has to be updated if either the DT structure changes or new visibility information becomes available. First, we describe the energy update from $E_n(\mathcal{L})$ at time $n$ to the new energy $E_{n+1}(\mathcal{L})$ if new visibility information is available followed by the description how the energy is adapted to a modified DT structure.

**Visibility update**. The integration of new visibility, i. e. a new ray $R_k$ is added, affects only the tetrahedra next to the 3D point the ray is connected to. To update the unary costs, we determine the tetrahedra $V_f$ and $V_b$ that are located in front and behind the 3D point with respect to the ray direction respectively and add the terms $\alpha_{free}$ and $\alpha_{occ}$ to our energy. Since the destination of $R_k$ is a point of the DT, $V_f$ and $V_b$ can be efficiently found as follows. We slightly shift the destination according to the ray direction and test in which tetrahedron the shifted point is located. For the pairwise term, we additionally determine the faces of $V_f$ that are not intersected by $R_k$ and set their costs to $\beta_{vis}$.

Since the integration of new visibility does not affect the structure of the DT, the number of terms in the energy stays constant and only a few terms are changed: two terms of the unary costs and three terms of the pairwise costs which are the faces of $V_f$ that are not intersected by $R_k$. Hence, in contrast to space carving algorithms the integration of new visibility information is independent of the number of tetrahedra intersected by the ray.

**DT update**. The energy function has to be adapted if the DT structure changes, i. e. whenever a new 3D point is added, removed or shifted. Typically, the modification of a single point (usually) only effects a local area $\mathcal{A}$, i. e. some tetrahedra are deleted and new tetrahedra are created. Technically, all tetrahedra within $\mathcal{A}$ are destroyed and the DT is re-triangulated for the points in $\mathcal{A}$ (see Figure 4.7). Consequently, we remove all terms from $E_n(\mathcal{L})$ that are related to deleted tetrahedra and add costs for new tetrahedra. The costs for the new terms are updated as explained before for the visibility update.

The case that $\mathcal{A}$ comprises all tetrahedra and therefore the whole energy function has to be recomputed, can theoretically occur but this case has never been observed in any of our experiments.

The complexity for adapting the energy $E_n(\mathcal{L})$ to a new DT structure depends on the number of rays connected to 3D points located in $\mathcal{A}$. Assuming $N$ 3D points are located in $\mathcal{A}$ and each is connected to $M$ rays on average, the complexity is $N \times M$.

#### 4.3.2.2 Incremental Energy Optimization

In order to extract the surface after updating $E_n(\mathcal{L})$ to $E_{n+m}(\mathcal{L})$, we have to solve the minimization problem again. Static graph cuts like [11] are designed to solve a random field only once. For this reason, if we want to directly use [11], we would have to re-build

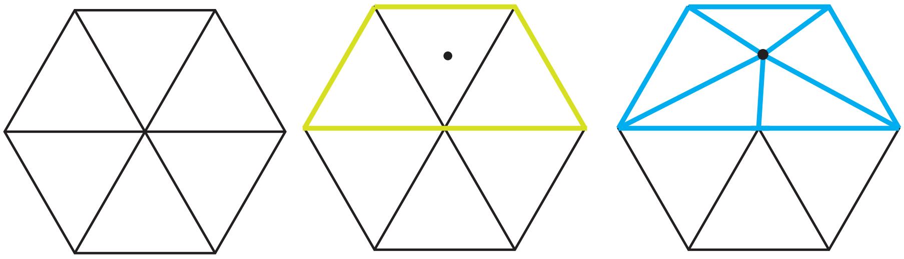


Figure 4.7: Insertion of a new point in the upper triangle changes the DT. Triangles within the change boundary $\mathcal{A}$ (green lines) are destroyed and the blue triangles are created.

the graph for each energy $E_n(\mathcal{L})$ and repeatedly solve the minimization problem from scratch, where the runtime for finding an optimal solution grows in practice linearly with the number of terms as we show in the experiments in Section 4.6.2. Although the overall problem size grows over time, the energies $E_n(\mathcal{L})$ and $E_{n+m}(\mathcal{L})$ typically differ only by a few terms. Kohli et al. [52] proposed a dynamic graph cut for such problems where a sequence of energy minimization problems has to be solved and the corresponding energy functions only differ by a few terms. The complexity for updating the weights in the graph is linear in the number of changed weights. In our case, also the time for optimization depends on the number of changed terms and therefore on average is independent of the overall scene size. This property combined with our fast adaption of the energy function to new 3D points and visibility information as described in Section 4.3.2.1 allows us a surface extraction in real-time independent of the overall scene size.

We start with the set of initial tetrahedra $V_{init}$ obtained from the DT of the point cloud $P_{init}$. We setup the energy $E_0(\mathcal{L})$ according to Section 4.3.2.1 and minimize $E_0(\mathcal{L})$ with the graph cut algorithm of [11]. We then extract the triangular surface mesh by finding all pairs of tetrahedra $(V_i, V_j)$ where $V_i$ and $V_j$ are labeled differently. Finally, we smooth the resulting mesh using a Laplacian kernel [29]. For each new 3D point, we first update the DT and the energy function and then integrate the new visibility information. Finally, we solve the labeling problem for the new function $E_{n+m}(\mathcal{L})$ by the dynamic graph cut [52]. Typically, we integrate several new points with their visibility information into the energy before solving the minimization, i. e. $m$ is between 500 and 2000, dependent on the user requirements. The evolution of a mesh over time is shown in Figure 4.8.

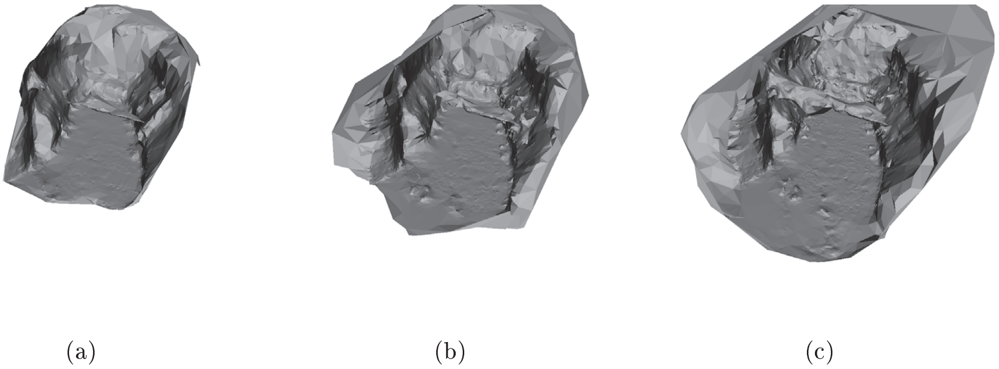


Figure 4.8: Evolution of the mesh over time. (a) Even a mesh extracted from 6 000 sparse 3D points gives information about the basic shape of the building. We can see how the mesh quality increases over time if (b) 14.000 or (c) 34.000 scene points are integrated.

## 4.4 Quality Visualization

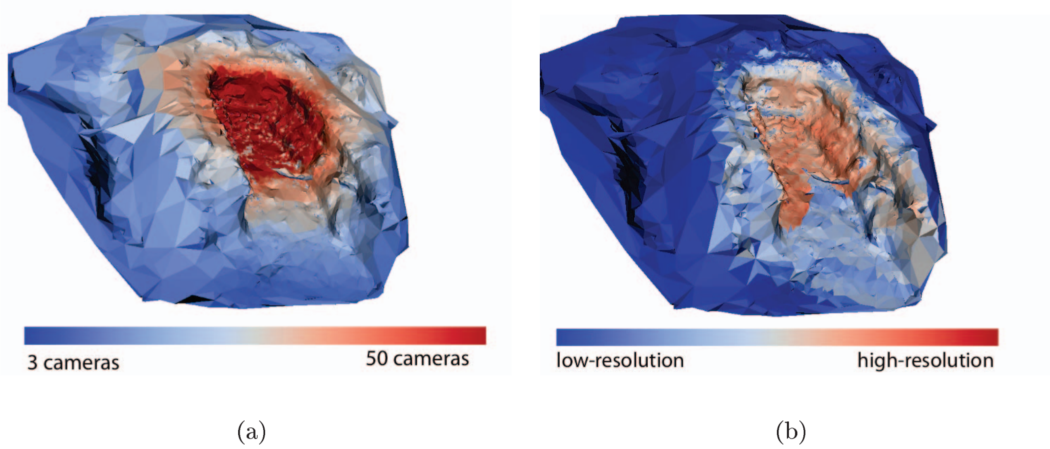
The combination of the real-time SfM with the incremental surface reconstruction enables us to obtain camera poses, the sparse point cloud and the surface mesh in real-time. This information can be instantly provided to the user already during the image acquisition. In order to support the user to judge on the reconstruction quality, we visualize additional quality measures of the reconstruction. In particular, the Ground Sampling Distance (GSD) and the degree of redundancy are overlaid on the current surface mesh. To evaluate the GSD, we calculate the maximum resolution a mesh triangle is mapped to in image space. We re-project each triangle $T_i$ of the mesh $S$ to each aligned camera $I_t$. We then calculate the maximum resolution which corresponds to the minimum value of the GSD

$$R(T_i) = \min_{I_t} \sqrt{\frac{A(T_i)}{P(T_i, I_t)}} \tag{4.13}$$

where $P(\cdot,\cdot)$ is the number of pixels that triangle $T_i$ covers in camera $I_t$ and $A(\cdot)$ is the area of the triangle in 3D space. To handle self-occlusions of the mesh correctly and for efficient calculation we employ the GPU that is optimized for visibility estimation of meshes. For calculating $R(T_i)$, we assign a unique color to each triangle. We then render $S$ using OpenGL from viewpoint $I_t$ and read out the image buffer. We calculate $P(T_i, I_t)$ by counting the pixels that have the color assigned to $T_i$.

The degree of redundancy can be computed at the same time by counting the number of of cameras $T_i$ is visible in. We define that $T_i$ is visible in $I_t$ if less than 50% of $T_i$'s area is occluded. This prevents triangles from being counted as visible that are largely occluded. Since the mesh is rendered on graphics hardware and only counting is performed on the CPU, the coverage computation takes around 50 ms for a single viewpoint.

To visualize both measures, we overlay the mesh by a color map according to the measure's value. The user interactively selects which information he requires for the decision on his next step. Since the scale of the SfM result is arbitrary, $A(\cdot)$ is typically not in metric scale and we determine the range of the color map by $\alpha$-trimming all values of $R(T_i)$ where $\alpha = 10\%$. If the scale of the reconstruction can be determined, for example by aligning the reconstruction to GPS data, we can choose the color map according to predefined resolutions. Figure 4.9 demonstrates this visualization for the reconstructed of an atrium.



(a)

(b)

Figure 4.9: Visualization of redundancy (a) and (b) resolution

## 4.5 Integration into Real-Time Feedback Application

We combine the real-time SfM, the incremental surface reconstruction and the quality visualization into a common application that integrates a new high-resolution still image within less than two seconds. To achieve real-time performance, we make use of todays multi-core CPUs and distribute the individual parts onto different threads. The localization of a new image as well as the structure expansion part are running in a single thread. SIFT features that are used for localization and point triangulation are extracted by the GPU implementation of Wu [105]. Because the camera positions and the triangulation of 3D points is performed incrementally, the problem of error accumulation may occur. We reduce this effect by performing bundle adjustment in a parallel thread. Since bundle adjustment for a large set of cameras is computationally complex and grows with the number of involved cameras and points, we follow the idea of Klein et al. [47] and perform bundle adjustment in a sliding window, i. e. we limit the number of cameras that are optimized to the last $n = 10$ added cameras. Hence, the optimization complexity stays constant over time.

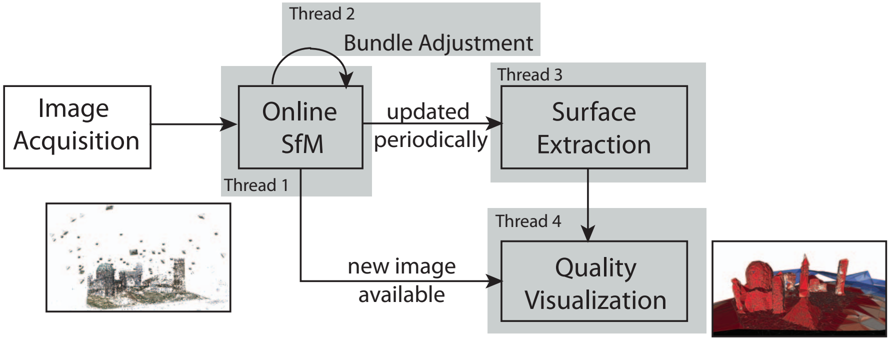


Figure 4.10: Overview over implementation details. We split the individual steps into separate threads for parallel execution. Bundle adjustment also runs in a parallel thread to prevent error accumulation. The GPU is used for feature extraction, feature matching and the visualization of the reconstruction's quality.

The continuous bundle adjustment causes that the integration of the incremental surface extraction method is not straight forward. Our method presented in Section 4.3 can add new triangulated 3D points but once integrated, points cannot be changed anymore. However, bundle adjustment may modify also the position of the points that are already integrated into the surface. We circumvent this problem by a so-called *late fusion*. Instead of using all sparse points for meshing, we only select those points that have been optimized by bundle adjustment several times. Once a point is integrated into the DT its position is not updated within the DT anymore. Over a long time or if a loop closure occurs this may cause a deviation between the extracted surface mesh and the real-time SfM result. However, the problem that SfM and the extracted surface mesh differ is diminished by the fact that only the last $n$ cameras are considered during bundle adjustment and therefore most of the scene points remain constant.

In the fourth thread we perform the quality visualization. In our implementation we update the quality measures if either a new frame is localized within the on-line SfM or the mesh is updated. In case of a new frame, the mesh is only projected into the according camera. In case of a mesh update, we recalculate the mesh quality using all registered cameras. In an improved implementation one could first determine the triangles that has been changed during the mesh update and identify the cameras they are visible in. This would further reduce the computational complexity.

Figure 4.11 shows the user interface. On the top part of the window the user sees the acquired images. If an image is not surrounded by a colored border, this implies that the image is not yet processed. Green marked images have been registered successfully whereas red have not been localized. Hence, the user instantly recognizes if a new collected image has been successfully integrated into the reconstruction. On the left window, the sparse point cloud and the camera positions are presented. On the right side, the surface mesh is shown and color coded according the quality measure. The user can move freely in the scene and can select between the visualization of the redundancy and the resolution.

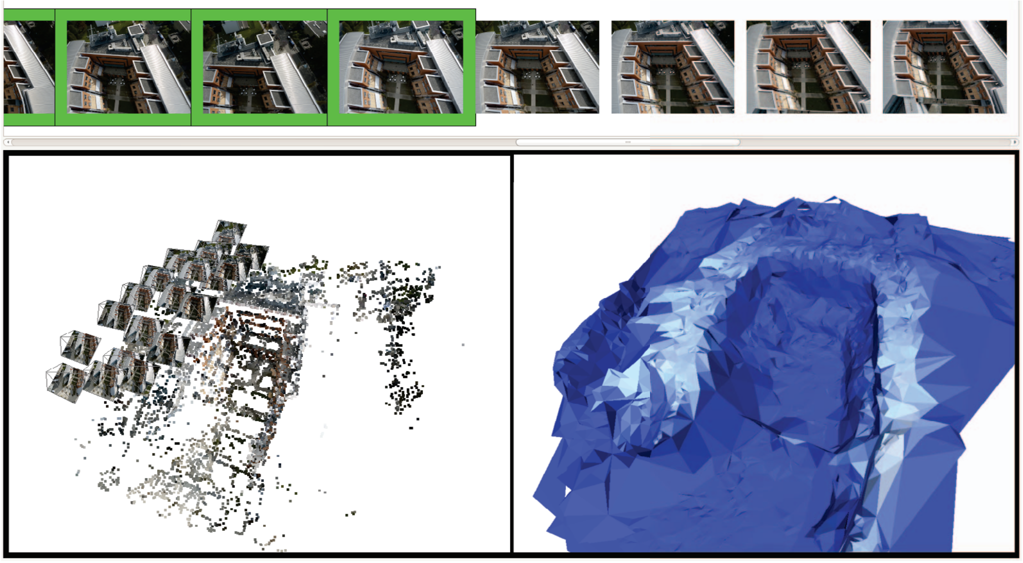


Figure 4.11: User Interface. In the top row the users sees the acquired images and if the image has been integrated in the spare reconstruction. On the left side, the reconstruction of the SfM is shown. The extracted and color coded mesh is shown then on the right. The user can move freely within the scene.

## 4.6 Experiments

The presented method for interactive SfM consists of several subsystem where each of them is considered as an individual contribution. Therefore, we first evaluate each subsystem and finally perform an experiment that takes into account the combination of all methods.

In Section 4.6.1, we analyze the properties of our new image-based localization method. In the following Section 4.6.2, we demonstrate that a rough surface mesh that is extracted from sparse points is useful for various applications. Furthermore, we provide evidence that the proposed method is suited to create a surface mesh in real-time while the computational complexity is largely independent from the overall scene size. Finally in Section 4.6.3, we perform experiments that demonstrate that the interactive feedback method is a useful tool to reliably generate image sets that are suited for SfM.

### 4.6.1 Evaluation of Efficient Image-based Localization

In this section, we evaluate the efficiency of our optimized image-based localization method. In an initial experiment, we investigate which of the state-of-the-art methods (brute-force matching or image-retrieval based approach) are better suited for a coarse-to-fine localization approach. The result that image-retrieval methods are more successful in case of small images led to our new localization approach. In the second experiment we provide evidence that each of our improvements increases the localization rate of small images while reducing the computational complexity at the same time. In the final experiment, we show that in a coarse-to-fine setting our method can reduce the computation time to state-of-the-art methods by a factor of three.

#### Brute-Force vs. Image-Retrieval

We first investigate which of the state-of-the-art approaches, brute-force (BF) or image-retrieval based localization (IR), is better suited in a scenario where the resolution difference between reconstruction and localization is large. Therefore, we setup the following experiment. We acquired 354 still images of an outdoor scene with a resolution of $1280 \times 720$ and reconstructed the scene by a SfM pipeline. In a second run, we acquired a 4 minute video of the same scene also with the same resolution. We cropped one frame per second (238 frames in total) from this video and tried to estimate the camera pose with respect to the point cloud using both, the BF and the IR approach. For the reconstruction as well as for the localization we use the SIFT implementation of Wu et al. [105]. For both sequences the intrinsic camera parameters are known.

For the BF approach, we follow the setup defined in [83]. We select for each reconstructed 3D point a single SIFT representative that has the minimum distance to all others belonging to the same 3D point. To establish 2D-3D correspondences, we apply an approximated nearest neighbor search to establish feature matches. A correspondence between a 2D feature and a 3D point is established if the SIFT distance ratio test is smaller than 0.7.

For the IR based approach, we follow the main pipeline proposed in [42], which efficiently searches visually similar images using a vocabulary tree [76]. Given a query image $Q$ the task is to find a shortlist $S$ of $k = 10$ images that contains the most similar images within the database $D$, that was used to reconstruct the 3D point model. To determine $S$ we use all the extracted features within the database images $D_i$. The shortlist $S$ is then used to establish 2D-3D correspondences between $Q$ and the point cloud. As ranking scheme we used the probabilistic scoring proposed in [42]. The vocabulary tree was trained on arbitrary images downloaded from the internet. We choose a branch factor of 50 and a depth of 3 which results in 125 000 leaf nodes. In all experiments, we set the parameter $p_1$ of the probabilistic scoring to 0.2. We then establish 2D-3D correspondences between each pair $(Q, S_i)$ individually using the SIFT ratio test where only features of $S_i$ are considered that are connected to a 3D point.

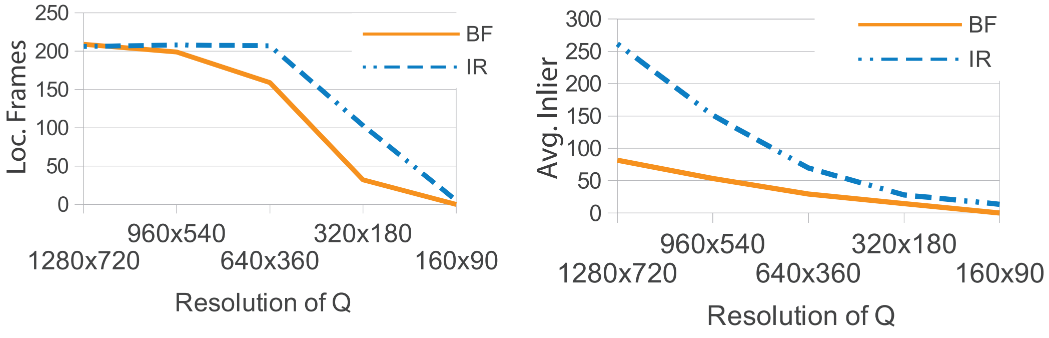
In both cases we solve the absolute pose problem with a RANSAC variant [13] where correspondences are ordered according to the SIFT matching quality. Following [42, 83], a pose is classified as valid if 12 or more 2D-3D inlier correspondences are found.

Figure 4.12(a) shows the number of localized frames for the BF and the IR approach when reducing the size of the query image stepwise from $1280 \times 720$ to $160 \times 90$ pixel. As expected, the number of extracted features varies between 8 000 features on the full resolution and 100 on the smallest scale. For full resolution images, both methods are nearly identical but the performance of BF drops drastically if the image resolution (and as a consequence the number of features) gets smaller. In contrast, the performance of IR stays constant even if the resolution of $Q$ is half the size. When using only a quarter of the resolution the advantage is even more explicit. Where BF localizes only 32 frames, IR is able to localize more than 103 frames correctly. And in case of only 1/8 of the resolution

(160x90 pixel) still 5 images can be registered using IR whereas BF completely fails.

The explanation for the weak performance of BF is as follows. The features of $Q$ are matched to all features of the point cloud. In such a way only very discriminative features are selected. If the query image contains a large number of features such an approach is successful because enough 2D-3D correspondences can be established. If the number of features is small, discarding features that are not discriminative with respect to the point cloud is crucial because often not enough 2D-3D correspondences for estimating a valid pose are found. In contrast, IR establishes 2D-3D correspondences between features of $Q$ and $S$. Hence, a feature of $Q$ has to be discriminative only with respect to $S_i$. This may cause more spurious correspondences but also reduces the number of missed matches. Spurious correspondences effect the runtime of RANSAC, whereas missed matches may prevent a successful localization. This explanation is also supported by the average number of inlier, which is consistently lower for BF than for IR (Figure 4.12(b)).

To summarize, this experiment has shown that in a low-resolution scenario IR achieves much better results than BF. Therefore we decided to improve the IR approach to reduce the overall localization complexity.



(a)

(b)

Figure 4.12: Resolution-dependent performance: (a) While localization performance stays constant for IR even if bisecting the image resolution, for BF the performance significantly drops. (b) IR has more than twice the number of average inlier compared to BF.

| | Outdoor | City-of-Sights |
|---|---|---|
| Image size | 1280x720 | 640x480 |
| Images | 354 | 383 |
| 3D Points | 164,424 | 106,506 |
| Features | 938,861 | 1,005,675 |

Table 4.1: Key information of the used reconstructions.

### Comparison to State-of-the-Art

In this section, we repeat the experiment of the previous Section 4.6.1 with our new method of Section 4.2.2 and investigate the performance difference to the baseline method (IR).

We use the dataset of Section 4.6.1 (*Outdoor*) and the publicly available *City-of-Sights* [27] dataset. This dataset (referred as *CoS*) consists of 10 different videos (called FARO 3 to FARO 12) at a resolution of 640x480 that are acquired by a camera mounted on a robot arm and moved within the scene. For the reconstruction, we have chosen every second frame of the sequence FARO 6. Key information on both datasets is shown in Table 4.1. For the localization, we consider the sequence FARO 4 which consists of 932 frames. Figure 4.13 shows the sparse reconstructions of both datasets.

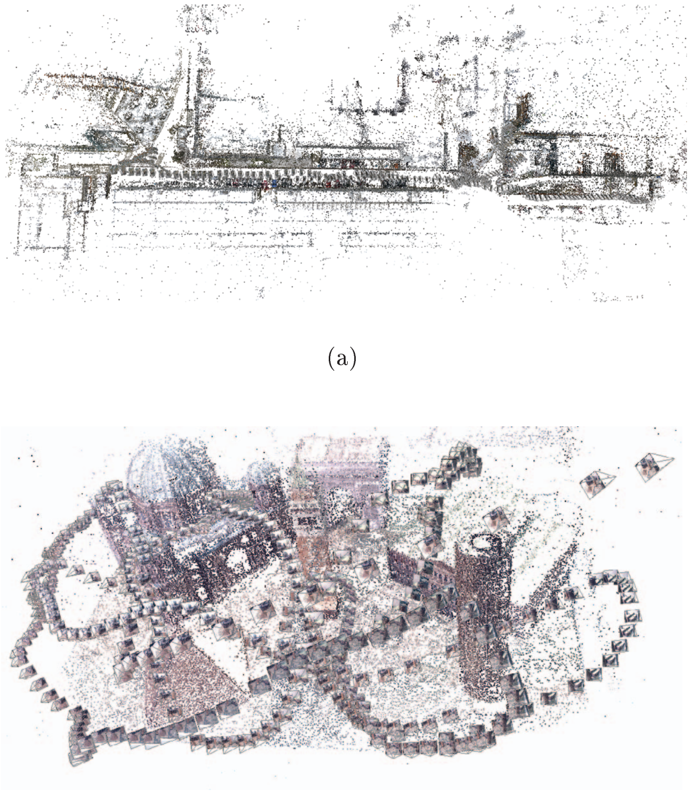
Figure 4.14 shows that our method consistently outperforms the baseline at all resolutions on both datasets. On the *Outdoor* scene, the largest gain of our method is obtained at a resolution of $320 \times 160$, where the baseline method localizes 103 images and our proposed method registers 163 frames. Compared to BF we even increase the number of registered frames by a factor of five. When using the full resolution of $Q$ which is the same as the resolution of the images used for reconstruction, our extensions do not adversely influence the result.

### Increased Robustness

In this experiment, we show that the modified image ranking as well as the removal of fine details during the 2D-3D matching improves the overall localization rate of low-resolution images.

When performing feature selection as outlined in Section 4.2.2.2, localization performance is slightly increased (blue fine-dotted curve) over the baseline. The reason is that by filtering the fine confusing features the probability that the SIFT ratio test is passed increases. By applying our improved ranking as introduced in Section 4.2.2.1, the overall number of localized frames increases by more than 30% on the *Outdoor* scene and 15% on the *CoS* dataset (green solid curve). Combining both of our proposals yields the highest number of localized frames (green dashed curve). The increased ranking quality is also visually noticeable. The baseline method delivers a shortlist where 8 out of 10 images are showing a completely different part of the scene, whereas our method delivers consistent results, as it is illustrated in Figure 4.16.

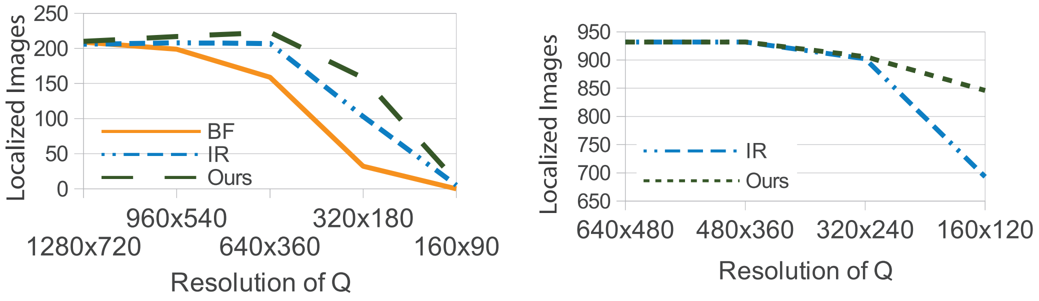
Since we remove features from the matching process, we potentially loose correspondences. Figure 4.17 disproves this concern for both sequences. The number of inlier per localized image is nearly identical in comparison to the baseline method for both datasets.



(a)

(b)

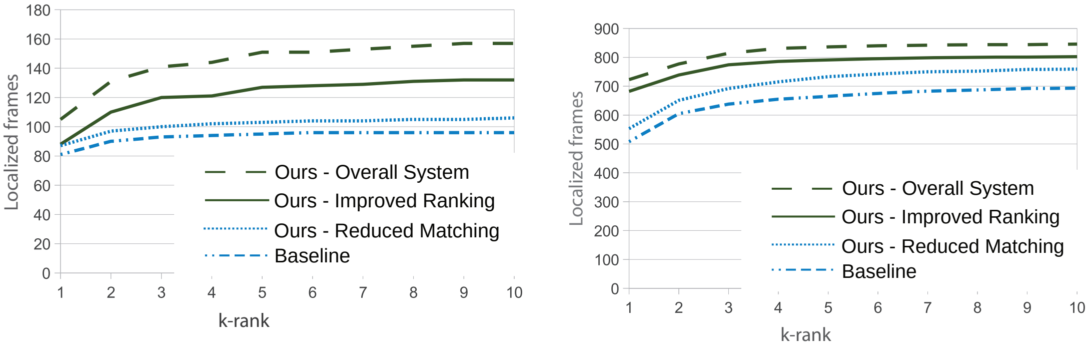
Figure 4.13: Underlying reconstructions that are used for the evaluation. (a) Reconstruction obtained from 354 images acquired by an MAV. (b) Reconstruction of the City-of-Sights dataset.



(a) Outdoor

(b) City-of-Sights

Figure 4.14: Localization performance when integrating our method of Section 4.2.2. Our approach increases the number of localized frames by more than 50% on images with a resolution of 320 × 180 on the *Outdoor* dataset (a). On the *CoS* sequence (b) the number of localized low-resolution images increases by 153.



(a) Outdoor

(b) City-of-Sights

Figure 4.15: Detailed analysis of the ranking and the feature selection in the localization process for (a) *Outdoor* with query resolution of 320 × 160 and FARO 4 (b) with query resolution of 160 × 120.

On small resolutions it is even slightly increased, i. e. on the *CoS* dataset our approach obtains 28 inlier on average compared to 24 inlier for the baseline method.

### Reduced Computation Time

Beside an increase of robustness, our method reduces the computational complexity. Figure 4.18 highlights the required computation time for registering an image at different resolutions. The timings include feature extraction on the GPU, image ranking based on

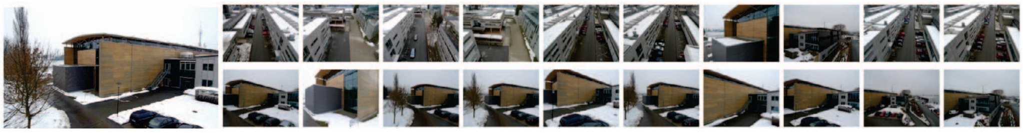
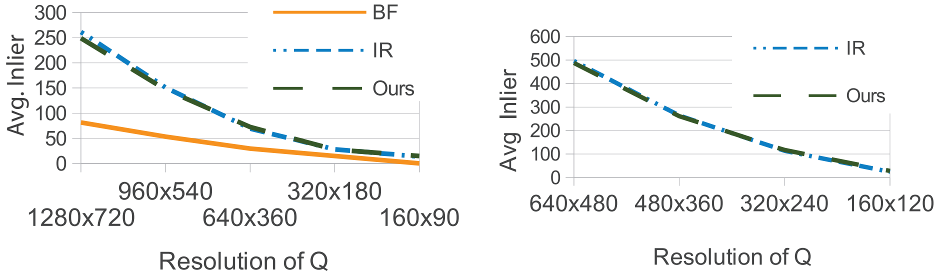


Figure 4.16: Comparison of image ranking using the probabilistic scoring (first row) and our modified version of Section 4.2.2.1 (second row). The query image (first column) has a resolution of 320x180 whereas the images in the database have 1280x720 pixel. The baseline method top-ranks 8 out of 10 images that do not show the same scene. Our method ranks the images in a more meaningful order.

retrieval, matching on the GPU and the RANSAC as explained in Section 4.2.2.3. Timings are measured on a Intel Core i7 960 and a NVIDA GTX 560 GPGPU. We can observe that our proposed method decreases the computation time in comparison to the baseline method. Caused by the filtering of features during 2D-3D matching, the complexity reduction gets larger, the more the resolutions of $Q$ and $D_i$ differ. For example, on the *Oudoor* sequences 105 features per $320 \times 180$ query image are extracted on average. For the baseline method these are matched against 2 609 features per shortlisted image $S_i$. After filtering the fine-detailed features, this reduces to a matching problem of 105 query features against 170 point cloud features per image. Overall, the number of feature comparisons decreases from 270 000 to 18 000. The same is true for the smallest scale images of the *CoS* sequence. Here, we reduce the matching problem down from 71 vs. 3 281 to 71 vs. 190 features.

The proposed modification of the scoring process requires computational overhead. First, $m(Q, D_i)$ has to be determined, the number of query features and database features



(a) Outdoor

(b) City-of-Sights

Figure 4.17: Inlier per localized frame dependent on the query resolution. For both sequences, the number of inlier is not affected by our proposed feature selection.

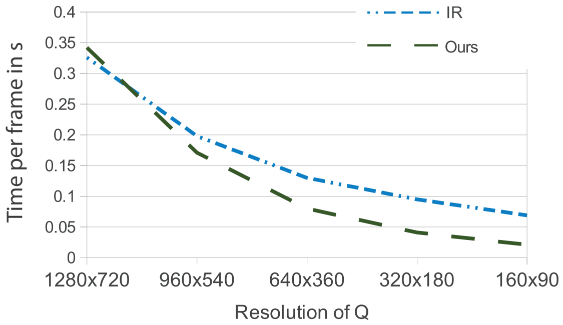


Figure 4.18: Required time for registration in dependence of the query image resolution. The filtering of unmatchable features decreases computational complexity. For larger differences between the query and database image resolution, this can save more than 50% of the time.

that are larger than $m(Q, D_i)$ (Equation 4.2 and 4.3) have to be counted and finally, only votes for documents whose features are larger than $m(Q, D_i)$ are counted. In absolute values, the voting of our method requires 5.7 ms per image on the full resolution and 3.5 ms using the baseline method. This overhead of 2.2 ms per image is negligible in comparison to the overall computational time of 326 ms for the localization of a full resolution image. As shown in Figure 4.18, the overhead for the scoring is over-compensated by the reduced time for feature matching.

Our pre-verification step as outlined in Section 4.2.2.3 reduces the number of required verifications drastically, especially if the number of inlier is very low or if no valid pose can be found. As shown in Figure 4.19, 90% of the hypotheses can be discarded if the inlier ratio is below 10%. If the ratio of inlier increases, our applied hypothesize-and-verify algorithm [13] already terminates after a few iterations and therefore in such cases the benefits are small. However, our pre-verification approach reduces computational complexity for images that are difficult to localize.

### Multi-Scale Localization

Since we are now able to localize low-resolution images with high probability, we show that a multi-scale localization approach can drastically reduce the overall localization effort on datasets that are representative for robotics and AR applications. We start with a downscaled version of the full-resolution image, try to localize it and if it fails, we iteratively consider increasingly larger scales.

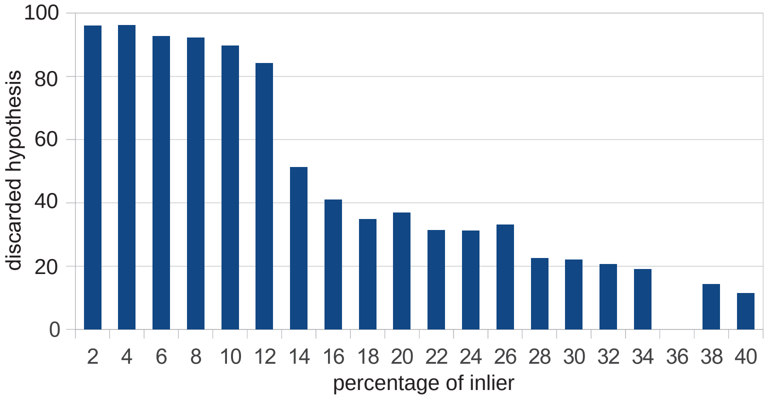


Figure 4.19: Percentage of hypothesis discarded by the pre-verification step according to the percentage of inlier.

Derived by Figure 4.14, we introduce three scales for the localization with scale factors 0.25 (level 0), 0.5 (level 1) and 0.75 (level 2) of the original image size. So we have at maximum three cycles for the localization of a single image. Each cycle comprises rescaling, feature extraction, ranking of the database images, 2D–3D matching and finally absolute pose estimation.

For the *Outdoor* dataset we obtain an average runtime of 73 ms per image which corresponds to a frame rate of more than 13 frames per second which is close to a frame rate that is required for tracking in robotics or AR applications. We found for 225 images a valid pose, where 163 images are localized on level 0, 60 frames on level 1 and 2 frames on level 2. The average number of inlier is 31.63 per localized frame. In contrast, the baseline method was only able to register 211 images on all three scales, where 94 images are localized on level 0, 110 on level 1 and 7 on level 2 with an average runtime for each image of 202 ms (5 frames per second) which is a reduction of 64%. Since our approach is able to register more images than the baseline, this demonstrates that the reduction of computational effort is not at the expense of the localization performance.

For the *City-of-Sights* dataset we register all 8 available image sequences (FARO 4 - FARO 12, except FARO 6) that show the complete scene. Each sequence consists of 800-1 000 images summing up to 7 338 in total. Our scale-space approach as well as the IR approach register 6 605 images (90%). In Figure 4.20 the average timings required for registering a certain image within the scene is shown. Our approach requires between 44ms up to 120ms to register a single image (green). The average time over the whole dataset is 87 ms whereas the baseline approach used in a multi-scale approache (IR, scale-space) requires 192 ms on average which shows that we can save more than 55% of the time.

For comparison we also show the timing when using the baseline approach on images of level 2 (IR, level 2). On average, it has the same complexity as the baseline approach integrated into a multi-scale setup. This shows that our contributions are required to reduce the computational effort of the localization.

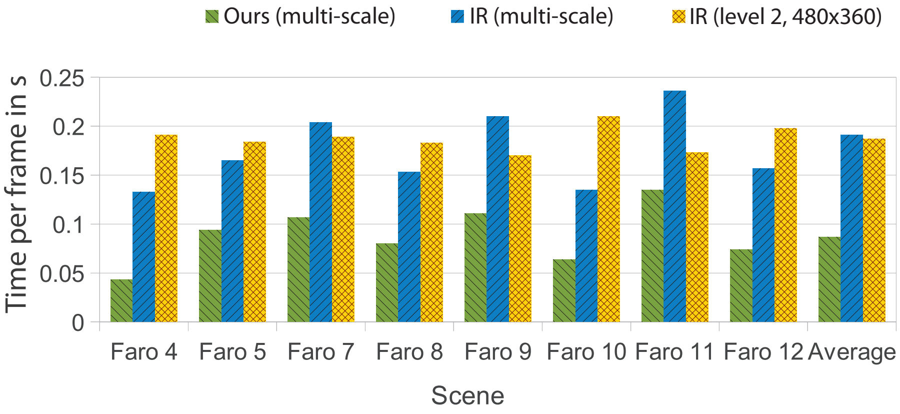


Figure 4.20: Average time required for registering frame within the scene.

### 4.6.2 Evaluation of Incremental Surface Extraction

The second important part of our interactive SfM pipeline is the incremental surface reconstruction using the sparse point cloud. Similar to the experiments of the efficient image-based localization, we individually evaluate this method. In the first experiment we show on a well-known groundtruth dataset that our proposed energy function reaches the same quality as the computational more complex state-of-the-art function proposed in [56]. We then demonstrate that our formulation is suited to incrementally extract a surface from a continuously growing point cloud in real-time and that the complexity typically is independent from the overall scene size. For all experiments we set the costs as follows: $\alpha_{free} = 10^3$, $\alpha_{occ} = 10^3$, $\beta_{init} = 10^3$ and $\beta_{vis} = 10^{-3}$. For the optimization we use the dynamic graph cut implementation of [52] and for the Delaunay Triangulation, we use the CGAL [1] software package because it reports which tetrahedra are deleted and created due to the insertion of a new point.

### Comparison to State-of-the-Art

In this experiment, we show that our novel energy achieves the same accuracy on sparse as well as on dense SfM point clouds as the more complex energy of Labatut et al. [56]. For accuracy evaluation, we use the dataset *Fountain* provided by Strecha et al. [90]. The dataset provides 11 high-resolution images and ground truth for camera positions and depth maps for each image. The sparse reconstruction is performed by an approach similar to Bundler [89] and results in 7 123 sparse 3D points. Each point is connected to 4.8 cameras on average.

We apply the surface extraction method of [56] as well as our proposed method on the provided data in a batch-based manner, i. e. we add all 3D points to the DT, setup and minimize the energy function only once. Figure 4.21(a) and 4.21(b) show the surfaces obtained by [56] and our method. Both the error maps and a visual comparison demonstrate that the surfaces are very similar. The accumulated histogram of depth map errors in Figure 4.21(e) quantifies the error in metric scale and gives evidence that both surfaces are very similar. We repeated this experiment with a densified point cloud obtained by PMVS2 [22] (370 000 points). The two upper curves in Figure 4.21(e) again show that the extracted surfaces are very similar.

On a second dataset we compare our result to [56] as well as to raycasting. This dataset consists of 77 300 3D points each connected to 4.4 rays on average but also around 20% of the points are triangulated by only two image features and therefore contain significant noise. Figure 4.22(c) shows that raycasting yields a noisy surface and hence is not suitable for such data. In contrast the surfaces obtained by our method and [56] are nearly identical (Figure 4.22(a) and 4.22(b)) but the computational complexity is very different: [56] requires 79 seconds for defining their energy function and solving it by graph cuts, whereas our approach needs only 32 seconds on a Intel Core i7-960 processor. The difference in computational effort is mainly caused by the definition of the energy function. While [56] has to perform a full raycast for each ray, we only have to identify the tetrahedra in front and behind the vertex and the first triangle that is intersected by the ray. Furthermore, our energy can be solved faster by the graph cut. While the optimization of [56] requires 740 ms, our energy is fully optimized in 430 ms.

Note that beside visibility information Labatut et al. [56] also include two further terms, a photo consistency and a smoothness term. Both can be integrated into our

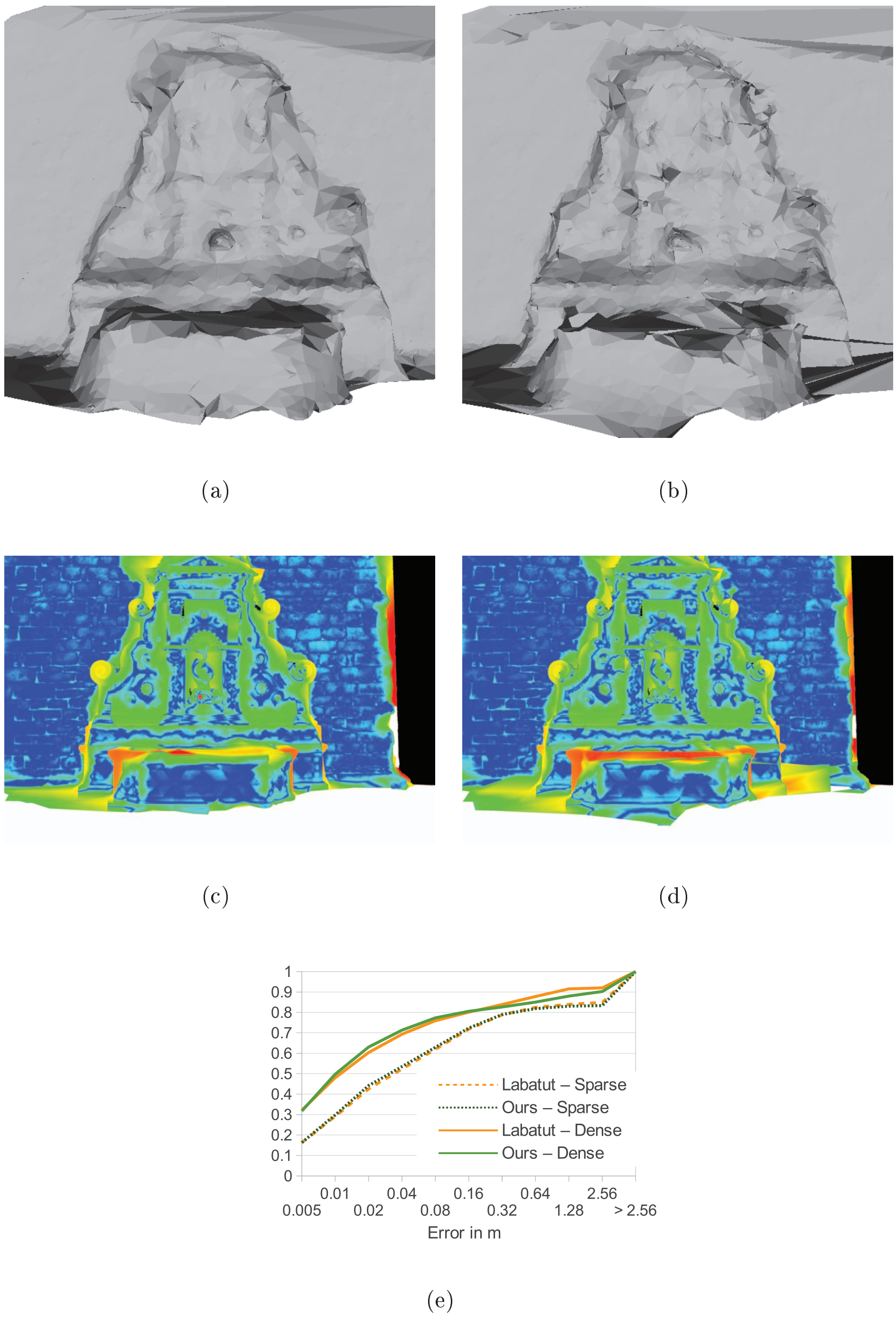
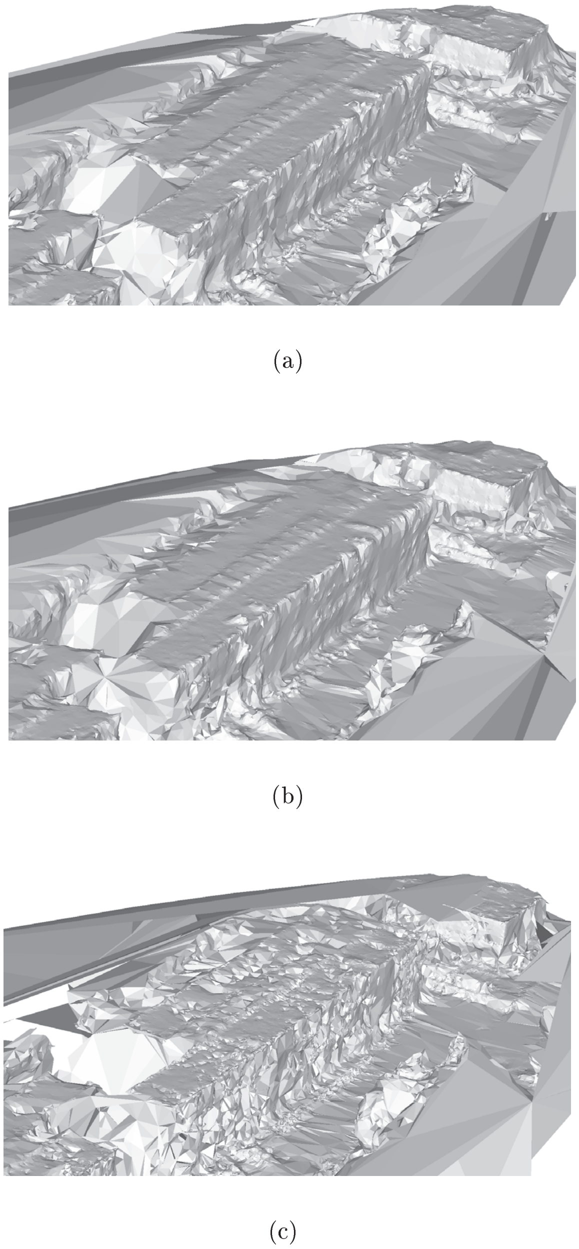


Figure 4.21: Fountain mesh. (a) Mesh extracted by our approach. (b) Surface extracted by [56] using only the visibility term of their energy function. (c) Color coded depth map error of our reconstruction of image 6. (d) Same evaluation for the result obtained by [56]. Blue indicates an error of less than 5 mm whereas distances above 2.56 m are coded in red (best viewed in color). (e) Accumulated errors for surface extraction from sparse as well as dense data using both approaches.

energy function without violating the incremental fashion of our method since they only depend on triangle properties of neighboring tetrahedra.



(a)

(b)

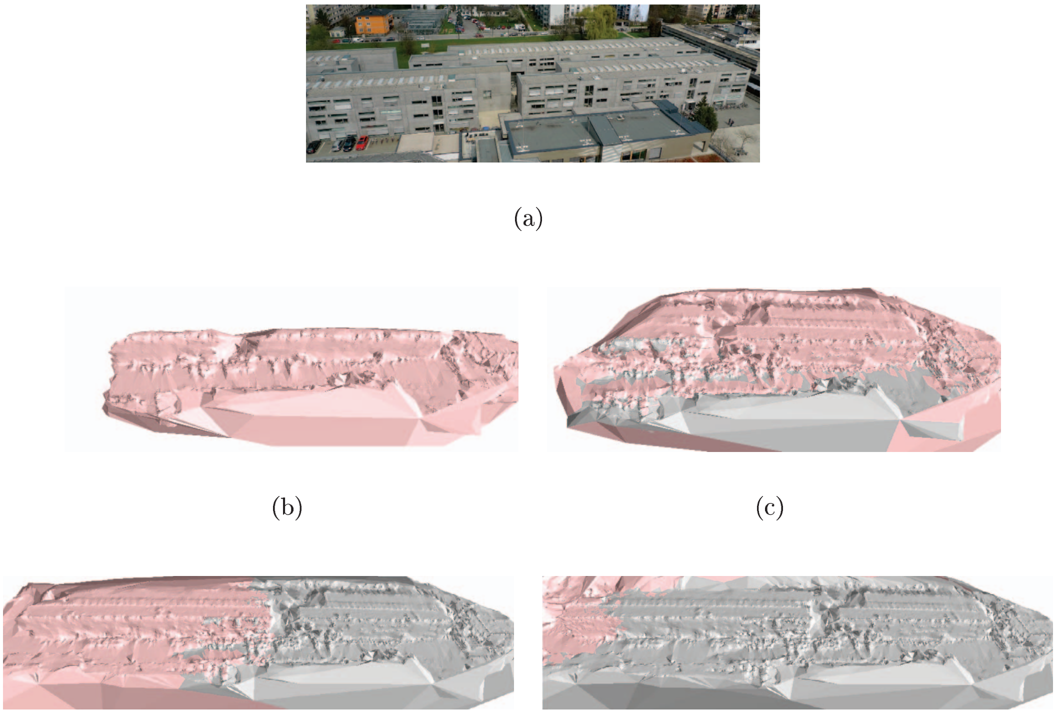
(c)

Figure 4.22: (a) Surface extracted by the method of Labatut et al. [56]. (b) Ours. (c) Space carving.

### Incremental Surface Reconstruction

In this experiment, we investigate the computational complexity of our proposed method in an incremental scenario. Similar to SLAM applications, we incrementally add new 3D points and visibility information and update the surface mesh after the integration of several hundred points. We determine the surface of two reconstructions that both consist of around 70 000 3D points. The first sequence was acquired by a Micro Aerial Vehicle showing an elongated building of 200m length. The second scene shows a medieval entrance where two figures are integrated into the wall.

We initialize our method with 1 000 3D points, calculate the unary and pairwise costs, extract the surface and incrementally add new points according to their creation time within the SfM pipeline. Our energy is updated each time a new 3D point is added to the



(a)

(b)

(c)

(d)

(e)

Figure 4.23: Incremental surface extraction over time. (a) Overview image of the reconstructed scene. (d) Reconstruction obtained at two different points in time. The gray part has been extracted from 40 000 3D points while additional 20 000 points create the red part of the reconstruction (best viewed in color).

DT and optimized after a defined number of points, e. g. 10 000 points, have been added. Figure 4.23 shows the surface of the building at different points in time after the integration of 40 000 and 60 000 points respectively. The red marked triangles indicate the part of the surface that has been changed since the last optimization. Since the images are recorded by a camera with forward motion, we can observe the growing of the surface over time.

The acquisition ordering of the second scene is quite different. Here, the photographer started the image acquisition with overview images and then took more detailed views of the two figures. This sequence demonstrates that our approach makes no assumption about the camera motion and is able to refine parts of an already extracted surface (Figure 4.24). After 10 000 points, only the basic structure of the scene is observable. With the integration of more and more sparse points at the figures, the details become more and more visible.

In our approach we assume that camera positions and 3D points are fixed and not modified after the insertion. When integrating our method into a keyframe-based SLAM system like PTAM [47] which uses local bundle adjustment for map optimization this assumption may be violated. To attenuate this problem a *late* integration step can be implemented, i. e. new 3D points are not integrated into the mesh directly after their triangulation but at the time when they have been optimized several times by local bundle adjustment. This decreases the probability that the structure is drastically changed.

For the evaluation of the computational complexity, we compare our approach to an incremental implementation of [56]. Since such an implementation is not yet available, we combine their method with the incremental space carving approach of [61]. We store for each tetrahedron a list of rays that pass through it. When the DT is changed, we update the energy function of [56] and minimize the new energy with the dynamic graph cut. For adapting the energy to a new DT, we have to intersect all rays going through deleted tetrahdra with all new created tetrahdra which is computationally expensive. Furthermore, we have to store the ray to tetrahdra assignment which requires a large amount of memory.

The incremental adaptation of [56] and our method consist of basically two parts: Update of the energy function according to new 3D points and the optimization using the dynamic graph cut. Figure 4.25 quantifies the complexity difference when updating the energy to a changed DT for the building sequence. The blue bars show the number of rays that are involved in updating the energy of 1 000 points. In our approach, we have to determine for each ray the tetrahedron in front and behind the destination vertex of the ray

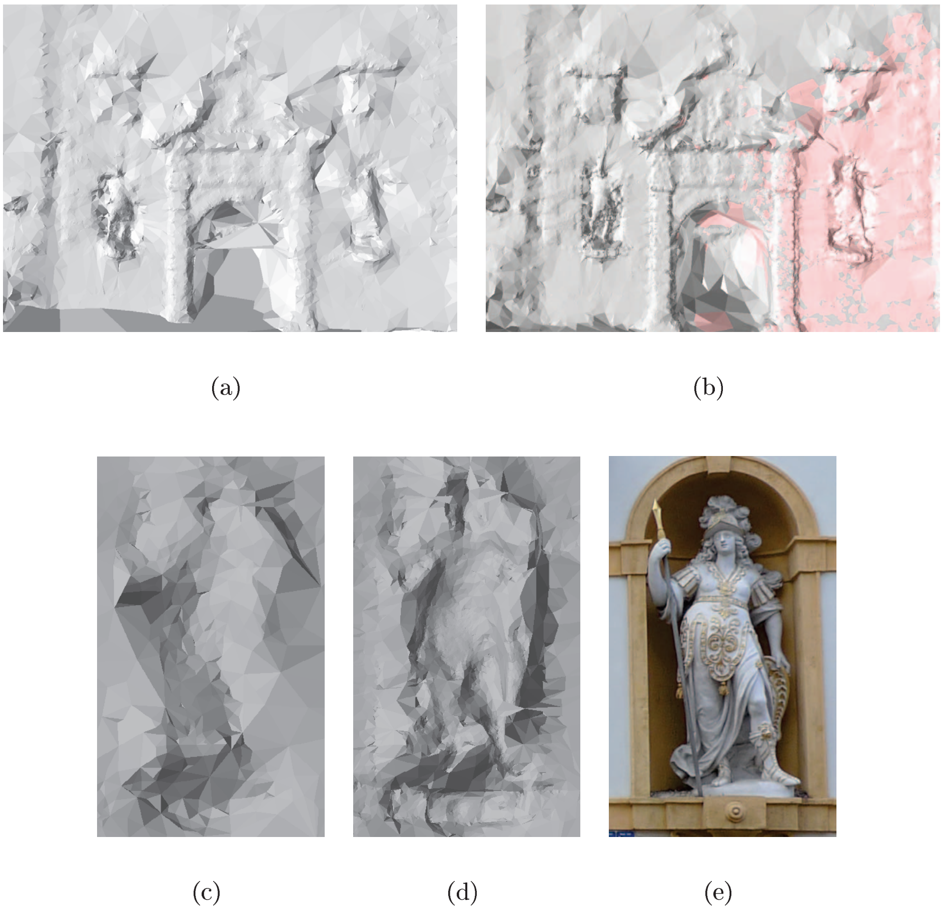
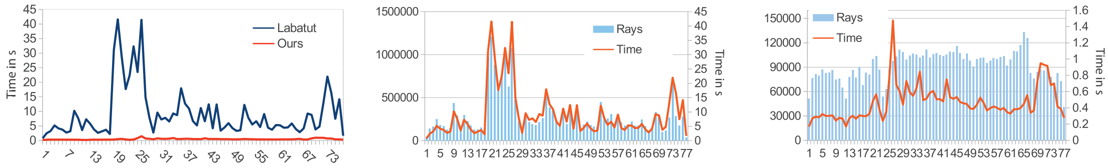


Figure 4.24: Surface evolution of the entrance sequence after the integration of (a) 10 000 and (b) 40 000 points. Consecutively added 3D points incrementally increase the quality of the right figure. (c) and (d) Cut-out of the right figure. (e) Right figure.

and the intersected interface in front of the vertex. On average, the adaption of the energy to the modified DT structure requires 0.44 ms per integrated 3D point. Typical SLAM applications like [15] generate a few hundred 3D points per second which can be integrated into the surface in the same time with our approach. The incremental implementation of [56] has to test for each ray which of the modified tetrahedra are intersected by which ray. The number of rays involved in the energy update of [56] is an order of magnitude higher than in our approach. Since for each ray [56] has to determine the set of tetrahedra that are intersected by the ray, the absolute time is on average more than 20 times higher (9159 ms vs. 440 ms). Another important fact for real-time applications is the variance of the complexity. The large deviations in [56] are caused by the following problem. If a tetrahedron is modified that is intersected by large number of rays, all of these rays have to be taken into account to update the pairwise energy term. For example, in the building sequence several tetrahedra are passed by more than 50 000 rays and if one of these is modified the integration time rises drastically.



(a) Runtime

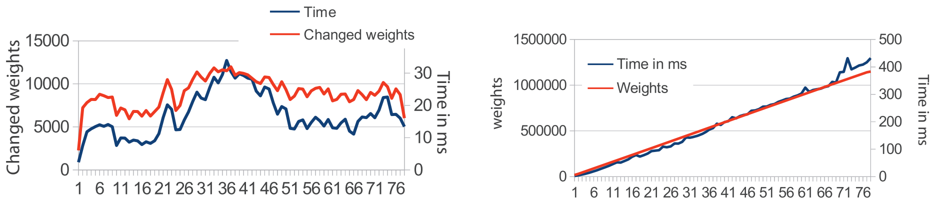
(b) Labatut

(c) Ours

Figure 4.25: (a) Difference in runtime for updating the energy function for 1 000 3D points. The x-axis shows the total number of 3D points in the mesh. (b) and (c) Time for energy update and the number of rays involved in the update. Please note, that scaling differs significantly.

The second part is to solve the labeling problem by minimizing the energy function. Standard graph cut methods are designed to solve static problems, i. e. the energy is once defined and the minimum is calculated. In contrast, our approach generates a series of energies with an increasing number of terms. Figure 4.26(b) shows that the number of terms grows nearly linear in the number of points integrated in the DT. When using a static graph cut solver like [11], the time for solving also increases linearly and requires 430 ms for the final energy. In contrast, the time for solving the dynamic graph cut largely depends on the number of changed terms (Figure 4.26(a)) and does not depend on the overall problem size. In the building sequence, typically between 10 000 to 15 000 terms are updated when integrating 1 000 points into the reconstruction. The time required for the optimization varies between 20 ms and 30 ms. Compared to the time for the energy update which is around 440 ms for 1 000 points, the time for optimization is relatively small. This comparison gives evidence that the dynamic graph cut reduces the computational complexity and is independent of the overall scene size.

To summarize, our experiments demonstrate that our approach achieves the same accuracy as state-of-the-art methods for sparse SfM point clouds with a reduced computational effort. Our energy is suited to work in an incremental manner and in combination with



(a) Dynamic graph cut

(b) Static graph cut

Figure 4.26: Dynamic graph cut vs. static graph cut. The runtime for solving the dynamic graph cut depends on the number of changed terms in the energy function, whereas in the static case, the runtime depends of the overall number of terms.

| | SfM Runtime | # sparse points | avg measurements per 3D point | # dense points |
|---|---|---|---|---|
| Bundler | 930 s | 28 215 | 3.33 | 2 416 144 |
| real-time SfM | 129 s | 23 218 | 3.85 | 2 399 999 |

Table 4.2: Comparison Bundler vs. our real-time SfM

the dynamic graph cut, computation time for energy minimization largely depends on the number of changed terms in the energy function.

### 4.6.3 Evaluation of Interactive SfM Feedback System

After evaluating the individual methods of our interactive SfM pipeline, we perform experiments with the overall system in this section. We first show that the real-time SfM reconstructs images up to 7 times faster when assuming that images are not taken randomly. We also tested the accuracy of the SfM result and demonstrate that the accuracy is comparable to batch-based SfM methods. Furthermore, we show that the instant feedback helps even a very experienced user to increase the number of images that are suited for a reconstruction. Finally, we demonstrate the versatile application areas our approach can be used in.

#### Incremental Real-time SfM vs. Offline Batch-based SfM

To compare the accuracy of real-time SfM to a state-of-the-art batch-based SfM approach, namely Bundler [89], we acquired an image sequence of 74 outdoor images (10 Mpx) of a church entry (Figure 4.28) and reconstructed them using both SfM methods. Because it is difficult to generate ground truth data for a large-scale outdoor dataset, a good indicator is the result of the dense matching step. Small errors in the camera alignment have large effects in the final dense reconstruction.

For both methods, we extract 4 000 SIFT features per image that have largest scale. The features are calculated by the SIFTGPU [105] implementation. Table 4.2 shows the comparison between both methods. The visual result is shown in Figure 4.27. Our approach requires 129 seconds which is 7.2 times faster than Bundler. Our approach generates less 3D points which is because we only find matches between the current image and the visually closest $n = 6$ that are obtained from the image-based localization. In contrast, we obtain an increased number of measurements per 3D point compared to Bundler. This allows to conclude that our cameras are connected more densely and therefore we can expect a similar accuracy. On average, our approach requires 1.8 seconds to integrate a new image into the map and to expand the map. Since we fix the number of images for matching, the insertion time is independent of the map size. The undistortion and feature extraction are dependent on the image size whereas the subsequent feature matching and map extension steps only depend on the number of features used for matching. Hence, these two parameters can be adjusted if less computational power is available or if faster image integration is needed.

In order to compare the accuracy of the real-time approach, we perform a densification using PMVS2. Since our approach performs only local bundle adjustment during the integration, we optimize the whole reconstruction before the densification. Figure 4.28 shows the densified SfM point cloud when using the sparse SfM result obtained by our real-time approach and the result when using the off-line reconstruction of Bundler. Both point clouds have nearly the same number of 3D points and their visual appearance is also very similar. This demonstrates that the real-time SfM result is accurate and can be used directly for subsequent processing steps like dense matching.

### User Support

To demonstrate that our approach supports the user during image acquisition, we performed an experiment with a user that has deep knowledge about SfM methods and is familiar with image acquisition for 3D reconstruction. We asked the user to acquire images of a church entrance that are processed by our real-time SfM algorithm. We advised

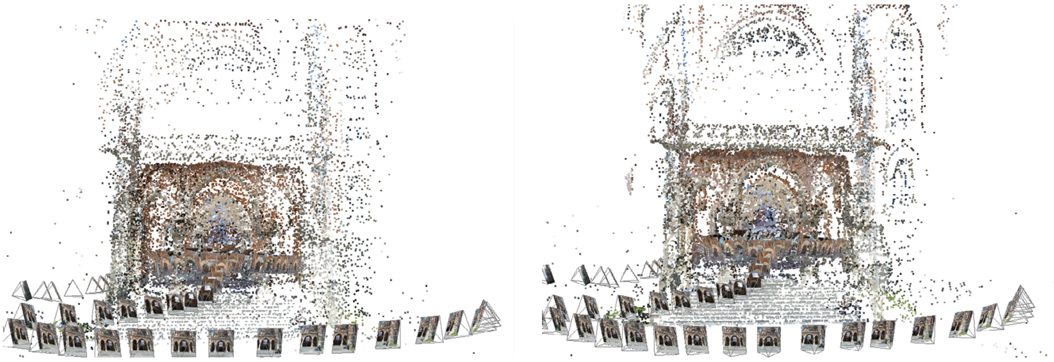

(a) (b)

Figure 4.27: Comparison of sparse reconstruction obtained by (a) the real-time SfM and (b) the batch-based method.

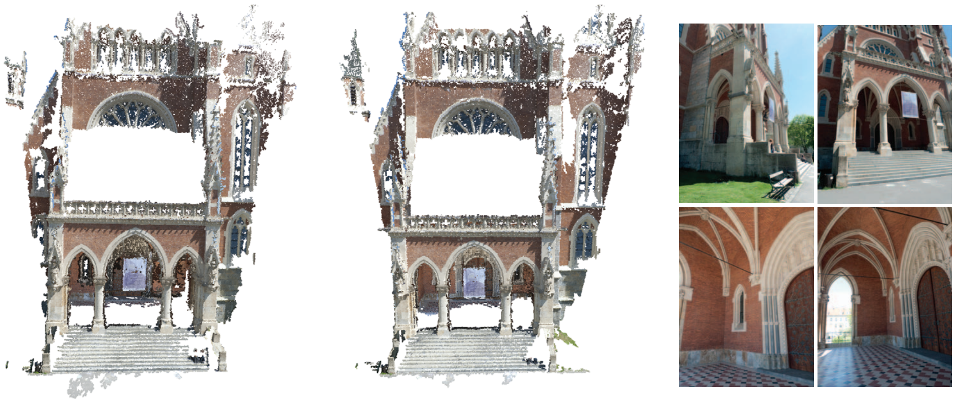
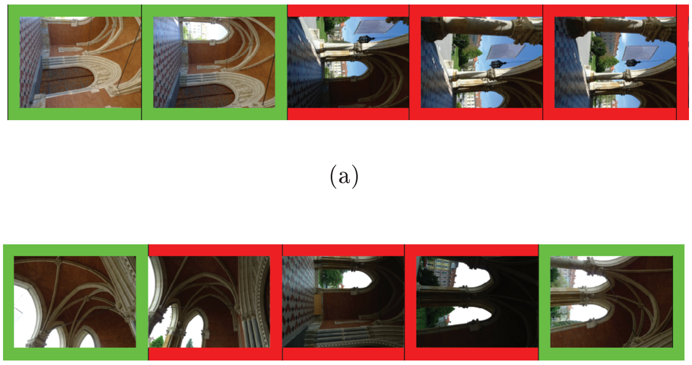

(a) (b) (c)

Figure 4.28: Comparison of dense results obtained by PMVS [22] when using the sparse result calculate by our online SfM (a) and the result achieved by Bundler (b). Both are visually similar and also the number of reconstructed 3D points is comparable. (c) Example of input images.

him to take images that have enough overlap to be integrated into the existing reconstruction. We were interested in an outside reconstruction as well as on the reconstruction of the vaulted ceiling, which made image acquisition more complicated because of the non-convex object's shape (see Figure 4.29). We performed the experiment twice: Without feedback and with real-time feedback. During the first experiment without feedback he was advised to acquire 100 images, in the second experiment with feedback he should stop



(a)

(b)

Figure 4.29: Sequence of acquired images. (a) If acquiring images without real-time feedback, the user did not recognize that images cannot be aligned to the reconstruction (red). The next 20 images could not be added incrementally. (b) With feedback, the user recognized the problem and reacted on that. Therefore, the following images are inserted correctly.

image acquisition once 100 images were integrated into the reconstruction by our system.

Without feedback, our method successfully integrated 74 of 100 images into a consistent reconstruction. Most images that cannot be reconstructed were acquired when looking from the inside of the entrance to the brighter background. The dynamic range of the camera is insufficient and parts of the vaulted ceiling are underexposed. Since the user did not recognize this, a sequence of 20 images are missing in the reconstruction. When incorporating feedback, the user recognized this problem after 3 images because our algorithm reports that the underexposed images could not be integrated into the reconstruction and the user adjusted the exposure settings. Figure 4.29 illustrates the difference between both experiments. He captured 118 images to achieve that 100 images are integrated into the reconstruction,which is a rate of 15% missed images compared to 26% in the experiment without feedback.

| | Images | 3D points | SfM time | Triangles |
|---|---|---|---|---|
| City-of-Sights | 61 | 22 752 | 110 s | 18 810 |
| Atrium | 127 | 28 374 | 238 s | 22 798 |

Table 4.3: Reconstruction results for the two different scenes. The meshing time includes the time needed for calculating the GSD and the redundancy information.

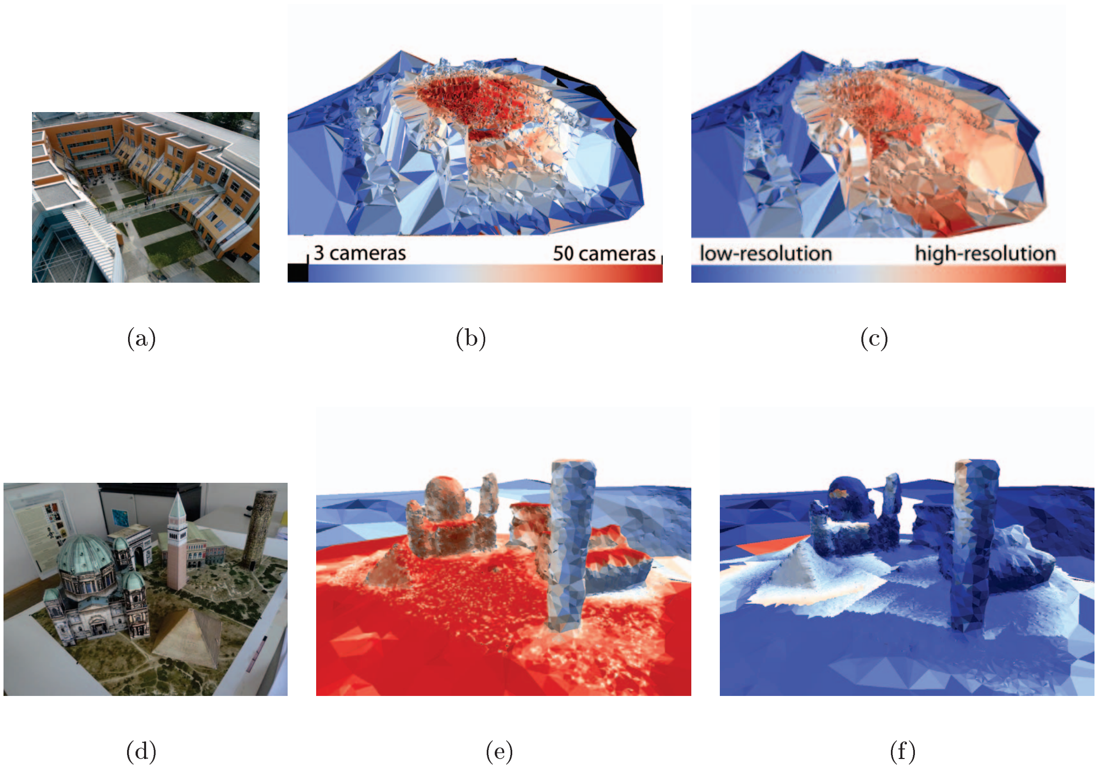
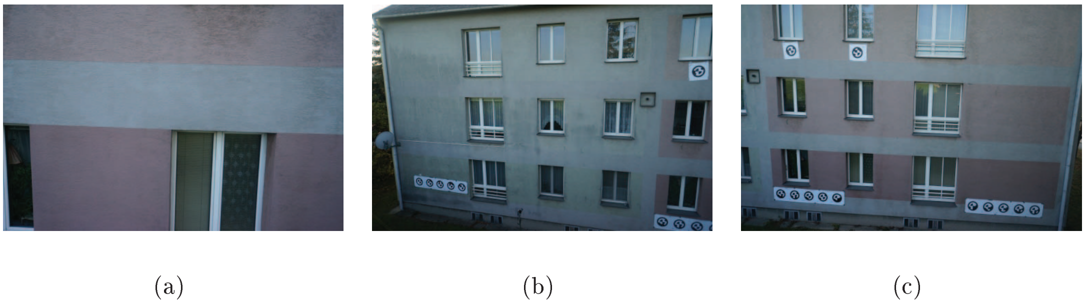


Figure 4.30: Results of the Atrium and the City-of-Sights scene. (a) and (d) Sample image of the two datasets. (b) and (e) Redundancy map. (c) and (f) Visualization of the GSD. Best viewed in color.

### Application Areas

We performed experiments on a large number of different scenes and also with different acquisition methods. The datasets vary between small paper models of the City-of-Sights [27] to very large-scale reconstructions of open pits. We collected the images with different modalities like a multi-copter, a fixed wing plane, a high-flying ultralight plane and by a handheld camera.

We observed that our method is useful for all listed types of scenes as well as for all acquisition methods. The largest benefit is obtained if the object of interest is geometrically complex like the church entrance in Figure 4.28 or if the type of acquisition allows large variations between individual pictures. This is especially a problem when acquiring images with a multi-copter because here, the user often changes the viewing angle between images drastically which prevents from successful feature matching. The instant feedback of the



(a) (b) (c)

Figure 4.31: Reconstruction of a weakly-textured façade at different scales. The instant feedback is important to determine the distance where even the fine details become visible and matchable across images. Furthermore, when taking images at different distances it often happens that images cannot be registered across the different scales. With the real-time SfM the feedback reports this issues instantly

real-time SfM calls the attention of the user in such cases.

The major issue when dealing with cameras that are attached to an MAV, is the transmission of the high-resolution images in real-time to the laptop that is located on the ground. Right now, it only exists a few hardware devices that are able to transmit the images wireless over a few meters in near real-time. For our experiments, we used a Secure Digital card that is able to transmit the images stored on the card using WiFi. The card can be installed in all consumer grade cameras. Since the range of this card is limited to 3 to 5 meters, we amplify the signal with an external repeater that is also mounted on the MAV. This allows the transmission of a 10 Mpx image over 30 meters in about 2 to 3 seconds.

One case study where we used our method is the reconstruction of a façade that is weakly textured. In order to guarantee that enough features can be found, one has to be very close to the surface. Here, our system instantly reports if the resolution is sufficient for the reconstruction. It is also helpful to check if images can be registered that are acquired at different distances of the façade. Figure 4.31 shows sample images of this challenging object. The mesh that evolves over time and the corresponding quality values are shown in Figure 4.32

Another application scenario is the documentation of a changing environment like construction sites. Such applications require that the reconstructions obtained at different points in time are aligned in the same coordinate system. For large reconstructions where significant parts changed over time, this is even for a human a non-trivial task. Here, our method can ease this task by assuring that images of parts are acquired that are static over time. The user performs an real-time SfM reconstruction when he or she collects images of the changing environment for the first time. If the user captures the environment for the second time, he or she can integrate the new images directly into the old reconstruction. Since the system gives feedback if a new image can be registered within the old reconstruction, the user can take images until a new image is localized within the old reconstruction. This ensures that both datasets can be registered automatically in a common coordinate system.

## 4.7 Conclusion

In this chapter, we presented an interactive SfM that couples the image acquisition process and the SfM pipeline tightly. The basis of our method is the ability to perform the SfM reconstruction from high-resolution still images in an incremental manner in real-time. We achieved this by splitting the reconstruction process into an image-based localization and a structure expansion part. To reduce the computational complexity of the image-based localization part, we developed a new method that takes account the scale of an image feature to improve a state-of-the-art image-based image-based localization method. In particular, we improved the image ranking, reduced the 2D–3D feature matching costs and implemented a pre-verification step in the RANSAC to discard erroneous hypotheses. In the experiments, we showed that this reduces the computational complexity and increases the robustness if images with different resolutions are localized against each other. The real-time processing of the images allows us to provide the user instant feedback if an image can be registered within the existing reconstruction. This helps the user to recognize possible problems of the acquired image already during the image acquisition.

Besides the camera poses, the real-time SfM provides a continuously growing set of triangulated sparse feature points. To obtain a surface representation, we developed a new algorithm that incrementally extracts a surface mesh given theses points. Based on a 3D Delaunay Triangulation of the sparse points, we formulated the meshing as an optimal binary labeling problem of tetrahedra which is formulated as a submodular optimization problem. The specialty of our formulation is that the energy can be easily adapted to

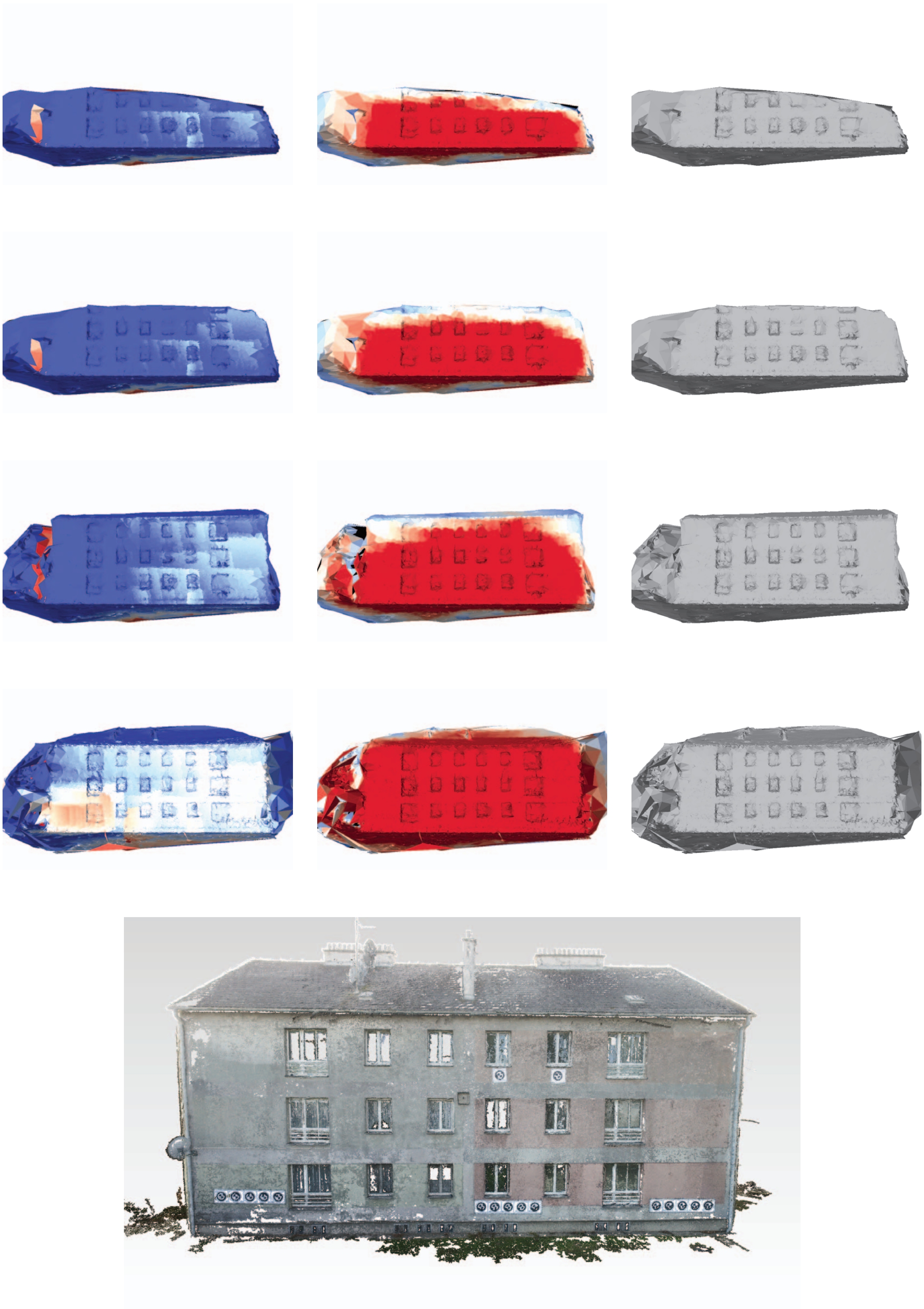


Figure 4.32: Evolution of the reconstructed façade at different points in time. The first column shows the color coding of the GSD where blue indicates a low resolution and red a high resolution. The images in the second column is the image overlap of the reconstruction. Red colored parts are seen by 30 and more cameras. The last column shows the reconstructed mesh without any color. The meshes are extracted after 50, 100, 150 and 200 integrated images.

the growing point cloud. By using a fully dynamic graph cut to solve the optimization problem, we ensure that the computational complexity is independent of the overall scene size.

The surface mesh allows us to derive important reconstruction parameters like the image overlap or the Ground Sampling Distance directly for individual parts of scene. By visualizing these parameters by color coding the mesh, the user gets an instant feedback which allows to assess the reconstruction's quality. Since this is also performed in real-time, the user can react immediately on that and can adapt the image acquisition strategy.

The integration of the real-time SfM, the incremental surface extraction and the quality visualization in a common framework that makes use of multi-core processors which allows to run the application on a standard laptop. We used the overall method to support the image acquisition in diverse scenarios, e.g. that collection from octo-copters, fixed wing MAVs or by a handheld camera. We experienced that the instant reconstruction on-site is a very useful tool to ensure that the image dataset is suited for an off-line batch-based reconstruction and therefore increases the reliability of SfM

# Chapter 5

# Large-scale View Planning using Geometric Priors

The quality and completeness of an image-based 3D reconstruction method largely depends on the spatial distribution of the input images around the object of interest. While interactive methods as proposed in the previous chapter are useful for the manual acquisition by a human user, remotely operated image acquisition systems like Micro Aerial Vehicles (MAVs) require a different strategy to be time- and cost-efficient. They either need pre-defined positions where images should be acquired or they explore the environment and autonomously decide where useful image positions are located.

The latter approach is well-known in robotics under the name of Next-Best-View (NBV) planning. Here, an autonomous device like a robot has to explore an unknown environment and to simultaneously build a map of the environment. Those approaches are related to Simultaneous Localization and Mapping methods and therefore are often computationally complex and require powerful processing capabilities. Since todays Micro Aerial Vehicles often have a limited flight time as well as limited computational resources, NBV approaches for acquiring images of a large-scale outdoor object like a complex of buildings are not yet directly applicable.

Generating a universally valid plan of viewpoints independent of the object of interest is in the field of close-range photogrammetry often not possible, since requirements like complete coverage largely depend on the scene geometry. Hence, the scene geometry has to be taken into account already during the view planning. Although 3D reconstruction

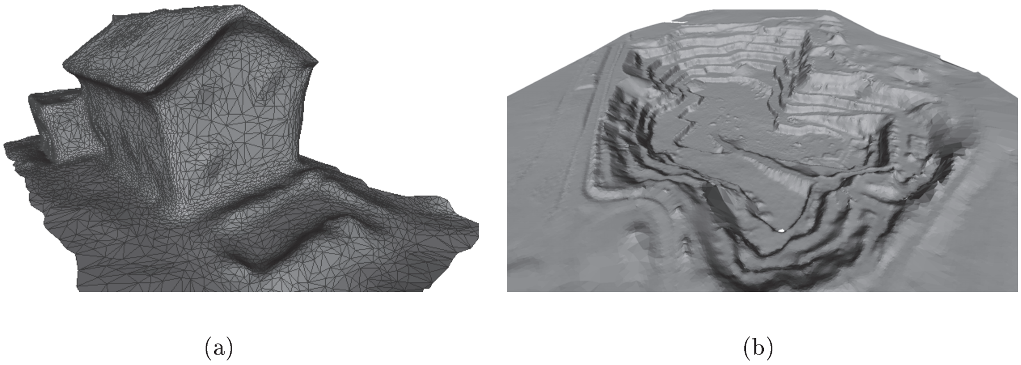

(a) (b)

Figure 5.1: A-priori information for view planning. (a)Mesh of a house during construction obtained from a previous flight. (b) DSM of a 1 km × 1 km open pit reconstructed by nadir views acquired by a microlight plane.

specifically aims to obtain the object's geometry, there are many applications where the objects geometry is (roughly) known in advance. For example, when acquiring images of a construction site the footprint is often available as prior knowledge. Or in case of reconstructing an open pit for topographical survey, terrain models, like the NASA SRTM data [19] or digital elevation models (DEM) of national mapping agencies, are available. If no a-priori knowledge is available, one can generate the rough geometry from nadir images acquires by a high flying MAV using the sparse incremental meshing proposed in Section 4.3.

Even if the geometry is known, acquiring a set of images that allow to sufficiently reconstruct the object of interest remains a complex task due to the large number and partially competing requirements of the problem. First, view planning should deliver a small number of views that cover the entire object to guarantee short flight- and processing times. Second, redundancy is necessary to achieve an accurate reconstruction. Third, the algorithm has to satisfy constraints like overlap and viewing angle between images to facilitate vision–based similarity computations [96]. Finally, the epipolar graph should be connected, otherwise disjoint reconstructions are obtained.

In our view planning approach, we embed the most important requirements for a successful reconstruction into a *multi-coverage set* problem. Given a-priori information about the object of interest as a triangular mesh, we first create a large set of hypothetical camera positions which are then reduced such that each part of the surface is at least covered by

$k$ cameras. Since redundancy is a necessary but not a sufficient condition for accuracy, we extend the multi-coverage approach taking the relative spatial distribution of cameras into account. The multi-coverage approach is a submodular maximization problem, but in contrast to a submodular minimization problem which can be solved in polynomial times, the submodular maximization problem is known to be NP-hard. However, the property of submodularity guarantees that a simple greedy optimization schema gives reasonable results with theoretical bounds on the solution's quality [55, 97]. In [21], the theoretical bounds for the quality of the approximated solution are derived.

In our experimental evaluation we demonstrate the effectiveness on synthetic data as well as on a real-world experiment with an MAV. Our comparison to a state-of-the-art view planning approach that directly minimizes the reconstruction accuracy shows that the view plan generated by our method is better suited for SfM.

## 5.1 View Planning as Constrained Set Multi-Covering

Calculating view points around a known 3D geometry can be casted as a sensor placement problem. In the sensor placement problem, a large number of potential sensor locations are defined and each sensor covers a limited part of the scenery. The goal is to select the minimum number of locations such that the whole scenery is captured by sensors.

Mathematically, this is described by the *set-coverage* problem. Given a set $\mathcal{U}$ of elements $\mathcal{U} = \{u_1, \cdots u_N\}$ that have to be covered and an overlapping partition $\mathcal{A}$ of $\mathcal{U}$. The partition is given as a set $\mathcal{A} = \{S_1, \cdots S_L\}$ where each $S_l$ consists of elements $u \in \mathcal{U}$: $S_l = \{u_{l_1} \cdots u_{l_k}\}$. The union of all elements in $\mathcal{A}$ is again $\mathcal{U}$. Please note that the sets $S_l \in \mathcal{A}$ may overlap. The goal is to find set $\mathcal{A}^* = \{S_i, \dots S_m\}$ such that each element of $\mathcal{U}$ is in $\mathcal{A}^*$ and $|\mathcal{A}^*|$ should be as small as possible. In case of our view planning problem, the set $\mathcal{U}$ corresponds to the faces of the prior mesh or points that are equally sampled on the surface mesh. A set $S_i$ contains all faces or points $u$ that are visible from a camera that is positioned at $K_i$. Figure 5.2 illustrates the different sets in case of view planning and Table 5.1 summarizes the definitions.

However, the formulation as a standard set coverage problem is not suitable for SfM using images from a monocular camera because the greedy optimization algorithm simply finds the most disjoint sets $S_i$ to minimize $|\mathcal{A}^*|$. In contrast, for SfM we require that

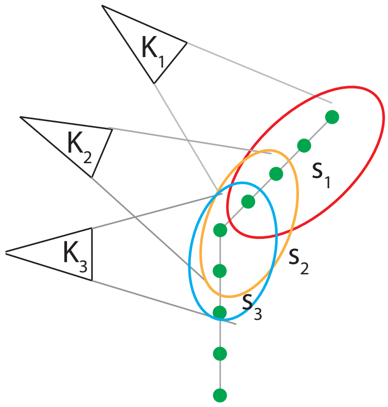


Figure 5.2: Definition of the different sets for the set covering. The green dots $\mathcal{U}$ are the equally sampled points on the prior knowledge surface mesh. Each potential camera position $K_i$ creates a set $S_i \subseteq \mathcal{U}$.

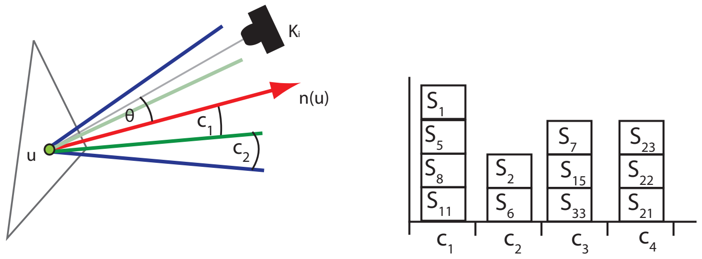
| Variable | Explanation |
|---|---|
| $\mathcal{U}$ | All faces / points of the mesh |
| $\mathcal{K}$ | Set of all possible camera locations |
| $S_i$ | All points that are visible in a specific camera $K_i$ |
| $\mathcal{A}$ | $\{S_1, \cdots S_L\}$ |

Table 5.1: Definition of view planning in set-coverage notation

each $u \in \mathcal{U}$ is visible in at least two cameras in order to triangulate a certain point. This requirement is equivalent to the property that each $u$ is visible in at least two subsets $S_i$. But for being robust and accurate, it is desired that the point has a higher redundancy, i.e. is visible in $k \geq 2$ cameras. Adding the constraint that a point is contained in more than a single set $S_i$ leads to an instance of a multi-cover problem which is also submodular [21].

But the accuracy does not only depend on the redundancy but also on the spatial distribution of the cameras used for triangulation: the larger the triangulation angles, the higher is the expected accuracy [9]. But even finding an optimal subset of $k$ cameras that maximizes the triangulation angle for certain 3D points is computationally expensive as a short numerical example illustrates: Assuming a point $u$ is visible in $R$ cameras we can create $\binom{R}{k}$ sets containing $k$ elements. If we assume that $R = 150$ and $k = 4$ this results in 20 M possible camera sets. Finding the set that maximizes the triangulation angle is therefore computationally not feasible for a large number of points.

Furthermore, maximizing the triangulation angle increases the triangulation accuracy but it complicates feature matching because the feature descriptors are often only stable up to a certain degree, e.g. SIFT is said to be stable up to out-of-plane rotations of about 30 degrees. Furthermore, the larger the viewing angles are, the more problematic are occlusion effects. Hence, we want to find a trade-off between large viewing angles and relative camera orientations that allows feature matching.



(a) (b)

Figure 5.3: Camera clustering. The set of cameras $K_i$ that observe $u$ is partitioned into different clusters $c_i$ according to angle between $K_i$ and $n(u)$. (a) illustrates the clustering for a point $u$. (b) exemplary shows the assignment of $S_i$ to the clusters of a point $u$.

In order to cope with these competing requirements, we propose to build clusters according to the spatial distribution of the cameras. Given a point $u$ and all sets $S_i$ that contain $u$, we cluster the sets $S_i$ according to the viewing angle of the associated cameras $K_i$ with respect to the normal of $u$. Figure 5.3 illustrates the clustering. We then enforce that the final set $\mathcal{A}^*$ contains at least one set of each cluster. This guarantees (a) a minimum triangulation angle, (b) a minimum redundancy for each 3D point, and (c) a limited viewing angle between selected view points such that feature matching is possible.

Beside the constraints for the reconstructed surface, there are also constraints on the set of selected cameras $\mathcal{A}^*$. Since we are interested in a reconstruction that contains all images, we have to ensure that the epipolar graph is not fragmented into several subgraphs. We obtain this by enforcing that cameras are selected such that pair-wise matching leads to an epipolar graph that contains a subgraph that connects all images. We show later, how these constraints can be easily integrated in the optimization.

Before going into the details of the view planning, we first start to explain how the camera hypotheses $\mathcal{K}$ are created.

### 5.1.1 Camera View Hypotheses

Before selecting the set of camera views $\mathcal{A}^*$, we have to create the set of potential camera views $\mathcal{K}$. We assume that our image acquisition device can access each point in space, hence $\mathcal{K}$ contains infinitely many elements. Since a regular discretization of all possible view points results in a very large set $\mathcal{K}$, we propose a different sampling strategy. This strategy is based on the assumption that the surface of the object of interest has to be reconstructed with a user-defined Ground Sampling Distance (GSD). The ground sampling distance defines the resolution of the reconstruction. Assuming a pinhole camera, the GSD $g$ is related to the camera's focal length $f$ and the distance $d$ from the camera plane to the object as follows:

$$gf = d. \tag{5.1}$$

Hence, in order to ensure a certain GSD, the cameras have to be positioned at a certain distance $d$ from the surface. Motivated by this requirement of a specific GSD, we propose the following sampling strategy.

Given our prior knowledge represented as a triangular mesh $M$, we assume that the object is textured in a way that yields equally distributed keypoints on its surface. Since our goal is to recover each potential feature point, these points build our set $\mathcal{U}$. In other words, $\mathcal{U}$ consists of 3D points that are equally sampled from the objects surface. Finally, we want to cover $\mathcal{U}$ by our potential camera set $\mathcal{K}$. Therefore, we create for each point $u \in \mathcal{U}$ a camera that observes $u$ fronto-parallel with the distance $d$ that depends on the user defined GSD $g$.

The density of $\mathcal{U}$ heavily depends on the texture of the object's surface, e.g. a pure white wall generates much weaker keypoints than a well-textured facade of bricks. In order to get a number of features per image that is independent of the texture, SfM pipelines typically tune the sensitivity of the keypoint detector such that the number of extracted keypoints per image is constant. Another solution to obtain a constant number of keypoints that are equally distributed over the image plane is to divide the image into an $n \times n$ grid and extract a single feature from each grid cell. Given the expected number of features per image and the distance $d$ that depends on the required GSD, we can determine the density $v$ of the potential 3D points on the objects surface as follows:

$$v = \frac{|F|}{whg}, \tag{5.2}$$

where $|F|$ denotes the number of extracted features per image, $wh$ the number of pixels in the image and $g$ the GSD. For example, we are given an image with 12 Mpx, $|F| = 5000$ expected features per image, and a GSD of 5 mm per pixel, this results in a density of 16.67 feature points per square meter.

When thinking of non-convex geometric structures, self–occlusions may occur and the optimal camera position for a certain $u$ might by located within a part of the object. In that case, we try to find a position along $n(u)$ whose distance is close to $d$. To guarantee positions that can be approached safely by the MAV, we define a margin around each obstacle. Camera positions within this margin are marked as infeasible and are removed.

### 5.1.2 Generating Classes of Equivalent Cameras

Having defined the set of potential views $\mathcal{K}$, we have to determine the sets $S_i$ which contain the points that are visible from camera $K_i$. Because the surface is represented as a triangular mesh, we perform a simple raytracing to determine the points that are visible in $K_i$. Since it is very unlikely that a point $u$ is matched reliably in images that have a very oblique view, we discard such highly distorted points from $S_i$. More formally, if the angle $\alpha = \sphericalangle(\overrightarrow{n(u)}, \overrightarrow{K_i u})$ is larger than a certain value $\lambda$, we exclude $u$ from $S_i$. Here, $\overrightarrow{K_i u}$ denotes the ray emitted by $K_i$ and passing $u$ and $n(u)$ denotes the surface normal of $u$.

In order to build sets of equivalent cameras that guarantee a certain triangulation angle while restricting the relative view point changes such that feature matching is still possible, we propose the following method. We group the sets $S_i$ that contain $u$ according to their viewing angle $\alpha$ into $k$ clusters $C = \{c_i, \dots c_k\}$ and force that at least a single camera of each cluster is contained in the final solution $\mathcal{A}^*$. This formulation has several advantages. First, clustering according to the viewing angle prevents that only cameras are selected that have a similar viewing angle with respect to each other and therefore we can guarantee a minimum triangulation angle for $u$. Second, the $k$ selected cameras for $u$ have roughly equally distributed viewing angles to $n(u)$. This is beneficial for automatic feature matching because the viewing angles between neighboring views cannot be arbitrarily large.

Finally, this formulation can be casted to a submodular coverage problem and therefore can be efficiently optimized as we show in the next section.

The clustering can be obtained for example by an equal discretization of the viewing angle into a histogram of $k$ bins. Alternatively, a standard cluster algorithm that provides a constant number of clusters like k-means [63] can be applied.

### 5.1.3 Finding Optimal Viewpoints

Finding the smallest set $\mathcal{A}^* \subseteq \mathcal{A}$ such that each point $u$ is covered from $k$ cameras is NP-hard and therefore only approximately solvable. Mathematically, this can be written as follows:

$$\mathcal{A}^* = \underset{\mathcal{A}_s \in P(\mathcal{A})}{\arg\min} |\mathcal{A}_s| \;\; s.t. \;\; F(\mathcal{A}^*) = \sum_{u \in U} C_u(\mathcal{A}^*) = |\mathcal{U}|k, \tag{5.3}$$

where $P(\cdot)$ generates the power set of $\mathcal{A}$ and therefore contains all possible sub-sets of $\mathcal{A}$. The function $C_u(\mathcal{A}^*)$ returns the number of the covered clusters of point $u$ given $A^*$. Hence, the maximum value for each point is $k$ and the sum over all $u \in \mathcal{U}$ is $|U|k$. Figure 5.4 gives an example for $C_u(\mathcal{A})$ for a single point $u$.

Although the problem is a submodular maximization problem an therefore NP-hard, it is approximately solvable by a greedy optimization approach. Despite its simplicity, the greedy optimization guarantees an upper bound on the solutions quality. Hence, we re-formulate Equation 5.3 to an optimization problem, where the objective function $F(\mathcal{A}^*)$ has to be maximized with the constraint that $|\mathcal{A}^*|$ is small:

$$F(\mathcal{A}^*) = \sum_{u \in U} C_u(\mathcal{A}^*) \geq \delta(|\mathcal{U}|k). \tag{5.4}$$

Starting with an empty set $\mathcal{A}^* = \{\emptyset\}$, the greedy optimization adds the element $s \in \mathcal{A}$ to $\mathcal{A}^*$ that in each iteration maximizes $F(\mathcal{A}^*)$. The algorithm iterates until $F(\mathcal{A}^*) \geq \delta(|\mathcal{U}| * k)$ where $\delta$ is typically set to 0.95 which means that 95% of all points have to be covered. For finding the next best element $s$, the naive implementation has to evaluate $F(\mathcal{A}^* \cup \{s\})$ for each $s \in \mathcal{A}$ which can be computational expensive if $\mathcal{A}$ is very large. Thanks to the submodular nature of $F(\mathcal{A}^*)$, the evaluation can be sped up using lazy evaluations [79].

Maximizing Equation 5.4 guarantees that each point $u$ is visible in at least $k$ cameras from different view points but it does not ensure that the resulting epipolar graph of $\mathcal{A}^*$

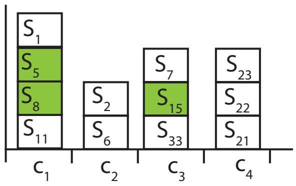


Figure 5.4: Given a set $\mathcal{A}^*$, $C_u(\mathcal{A}^*)$ returns the number of clusters that are occupied by elements in $\mathcal{A}^*$. Lets assume in this example $\mathcal{A}^* = \{S_5, S_8, S_{15}, S_{40}\}$ then $\mathcal{A}^*$ covers two of four clusters and therefore $C_u(\mathcal{A}^*) = 2$.

is not fragmented into subgraphs. To ensure this important property, we constrain the viewing angle and the overlap of the selected cameras in the greedy optimization: instead of adding $s \in \mathcal{A}$ that maximizes $F(\mathcal{A}^* \cup \{s\})$ directly, we require that $s$ has a certain image overlap to an element in $\mathcal{A}^*$ and at the same time does not exceed a certain viewing angle. This formulation has the advantage that each additional view can be registered within the previously chosen subset $\mathcal{A}^*$. Hence, incremental reconstruction algorithms like our method proposed in [36] can be used to perform the reconstruction already during acquisition.

From an algorithmic point of view, these constraints can be easily included in the greedy optimization. Instead of testing each $s \in \mathcal{A}$ if it maximizes $F(\mathcal{A}^* \cup \{s\})$, we build in each iteration a subset $A_c \subseteq \mathcal{A}$ that fulfills our requirements concerning overlap and maximum angle. Then $s_c \in \mathcal{A}_c$ is selected that maximizes $F(\mathcal{A}^* \cup \{s_c\})$. Adding the constraints to the optimization problem violates the property of submodularity, where we will discuss the consequences later in the experimental section.

The exact formulation of the constraints depends on the implementation details of the underlying SfM pipeline. In SfM pipelines like [89] that are designed for unordered images, an epipolar graph is calculated to determine the spatial relationship between images. Therefore, the constraints must be chosen such that it is possible to estimate for each image at least a single fundamental matrix. The requirement therefore is that a minimum number of putative feature correspondences can be determined between image pairs. This depends on the texture of the surface (which cannot be controlled by us) but also on the overlap and on the viewing angle between the image pairs. Hence, we force that a camera $s \in \mathcal{A}_c$ must have at least $n$ points in common with at least a single camera in $\mathcal{A}^*$ which have a triangulation angle lower than $\beta$ . In the experiments, we set $\beta = 30$ degrees which is motivated by the maximum angle that is matchable using SIFT. The number of common points is determined as a ratio of common visible points as follows. Given camera $K_i$ and $K_i'$ that observe $S_i$ and $S_i'$ points respectively, then $n$ is set to $n = \alpha\ min(|S_i|, |S_i'|)$ where $\alpha$ describes roughly the common overlap between both images and therefore is a value between 0 and at most 1. In the experiments we set $\alpha = 0.33$. But also other methods like [69] can be used to measure the pair-wise image overlap.

Algorithm 1 outlines the selection procedure in pseudo code. After initializing $\mathcal{A}^*$ as an empty set (Line 1), in each iteration the set $\mathcal{A}_c \in \mathcal{A}$ is selected that fulfills the defined constraints (Line 3). Then the element $s' \in \mathcal{A}_c$ is identified that maximizes the utility function $F(\mathcal{A} \cup \{s'\})$ (Line 4). The maximum element is finally added to $\mathcal{A}^*$ (Line 5). This procedure is repeated until $F(\mathcal{A}^*) \geq \delta(|\mathcal{U}|k)$ is reached. In Algorithm 2, it is outlined how the set $\mathcal{A}_c$ is determined.

**Algorithm 1** Constrained view selection algorithm.

```
1: $\mathcal{A}^* = \{\emptyset\}$
2: repeat
3:     $\mathcal{A}_c = constraints(\mathcal{A}^*, \mathcal{A})$ {see Algorithm 2}
4:     $s' = \arg\max_{s \in \mathcal{A}_c} F(\{\mathcal{A}^* \cup s\})$
5:     $\mathcal{A}^* = \{\mathcal{A}^* \cup s'\}$
6: until $F(\mathcal{A}^*) = \delta(|\mathcal{U}|k)$
```

**Algorithm 2** *constraints*$(\mathcal{A}^*, \mathcal{A})$. Formulation of the pairwise constraint function *constraints*$(\mathcal{A}^*, \mathcal{A})$. The function parameters are the current set of selected cameras $\mathcal{A}^*$ and the set of all cameras $\mathcal{A}$. The function return $\mathcal{A}_c$ which consists of all cameras that have enough overlap to another camera in $\mathcal{A}^*$. *commonpoints*() returns the number of common points of $s$ and $b$ whose triangulation angle is smaller than $\beta$.

```
$A_c = \{\emptyset\}$
for all $s \in \mathcal{A} \setminus \mathcal{A}^*$ do
    for all $b \in \mathcal{A}^*$ do
        if $commonpoints(s, b, \beta) > n$ then
            $\mathcal{A}_c = \{\mathcal{A}_c \cup s\}$
        end if
    end for
end for
```

## 5.2 Experimental Evaluation

We show in this section that our algorithm determines a set of camera positions that allows a state-of-the-art SfM algorithm to compute an overall connected reconstruction. We analyze how the constraints on the camera positions concerning overlap and the maximum viewing angle, impact the optimization result. Furthermore, we show that the greedy optimization of our proposed objective function is beneficial in several aspects. Furthermore, we compare our method to a state-of-the-art view planning approach that directly minimizes the uncertainty of the reconstruction.

We tested our method on two synthetically generated datasets but also in a real-world outdoor experiment which shows the limitation of our method.

### 5.2.1 Dataset

For evaluating our method, we selected two buildings of medium size. The first mesh represents a house that has been reconstructed by an SfM approach [41] using 273 images captured by a manually controlled MAV. From the resulting semi-dense point cloud [22], we determine a surface mesh containing 14 871 faces using the Poisson surface reconstruction algorithm [46]. As the second object of interest, we use the surface mesh of a medieval tower as shown in Figure 5.5(a). Both meshes are given in metric scale.

Since we generate for each point $u \in \mathcal{U}$ a potential camera view that is fronto-parallel to the underlying triangle, it may happen that cameras are generated at sharp edges that do not overlap. For example, at the corner of a house where two walls intersect, the resulting camera views are perpendicular to each other. To bypass this problem, we smooth the prior mesh using an Laplacian kernel as illustrated in Figure 5.5(a). For all experiments, we consider a point $u$ as visible in a camera $K_i$ if $\lambda < 30$ degree and $u$ is not occluded by any part of the mesh. For each point $u$, we build $k = 4$ clusters of cameras with respect to the angle between the surface normal and viewing angle of the camera. We use k-means [63] to find the clusters in the one dimensional data.

### 5.2.2 Constrained vs. Unconstrained View Planning

In the first experiment, we investigate how strong our constraints that guarantee a single, connected reconstruction influence the number of selected cameras. Since the constraints

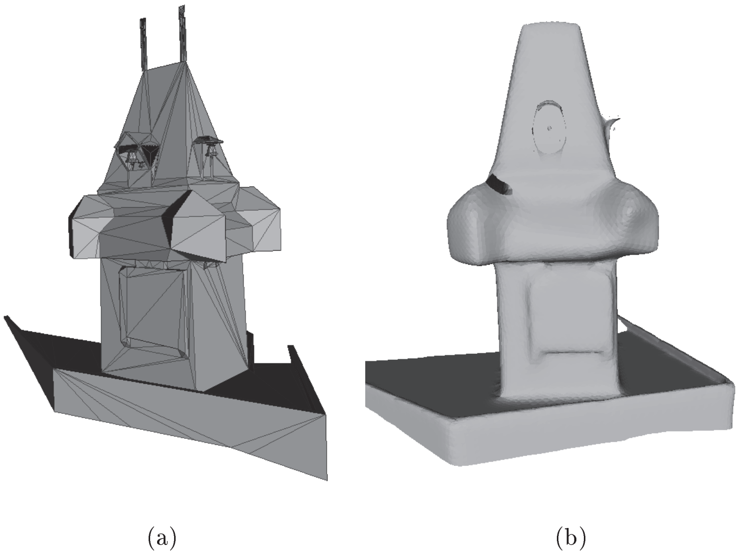


Figure 5.5: Input mesh medieval tower. (a) Original input mesh. (b) Smoothed input mesh that is used for planning. The smoothing avoids the creation of cameras whose viewing angles are too large to allow feature matching.

violate the submodularity property of $F(A^*)$, unfortunately the greedy optimization gives no theoretical guarantees on the quality of the solution. However, we show that the practical impact of the constraints is relatively small.

For this experiment, we create cameras that are located at 30 m distance to the objects surface. As intrinsic parameters we choose a standard wide angle consumer grade camera with an aperture of 76 degree. The potential camera positions $\mathcal{K}$ for the detached house are shown in Figure 5.6. For each triangle of the mesh we computed the number of cameras that observe it and visualize it as a color coded mesh (Figure 5.8(a)). Here, we only take cameras into account where the viewing angle and the normal of the triangle is smaller than thirty degrees. Obviously, planar parts are seen by a very large number of cameras whereas parts with a high curvature are seen by less cameras. However, as can be seen in Figure 5.8(b), even those high curvature parts are visible in more than 10 cameras. Only parts below the roof overhang are seen by less cameras which is because we restrict the cameras to be located above the ground plan.

In our first experiment, we run our method as proposed in Algorithm 1 without using the constraints that guarantee pairwise overlap, i.e. we set $A_c = A \setminus A^*$. Hence, the next best camera $s'$ is selected from all potential view points $\mathcal{K}$. In this configuration,

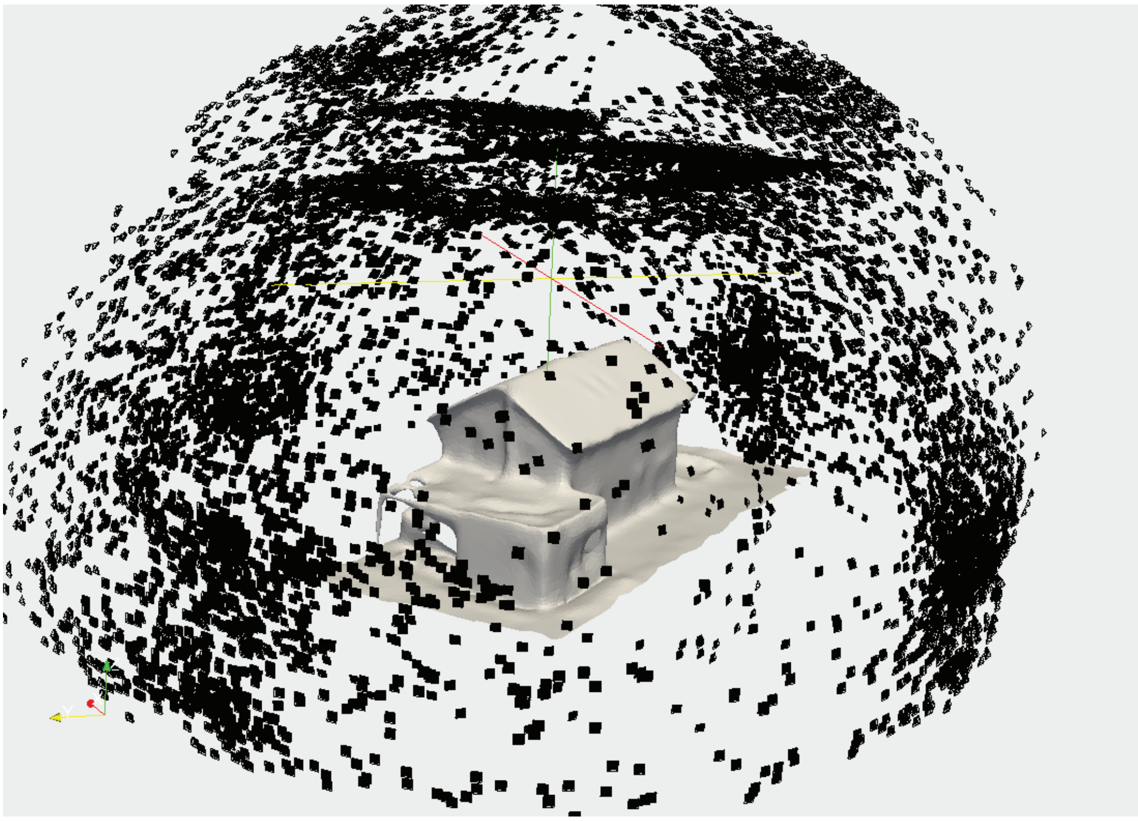


Figure 5.6: 12 000 potential camera locations around the house are generated.

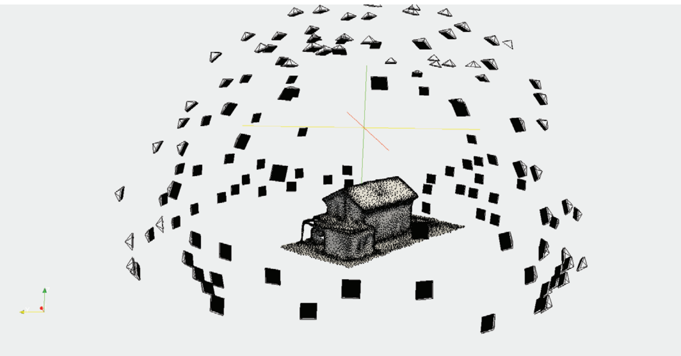


Figure 5.7: Selected view points for the detached house.

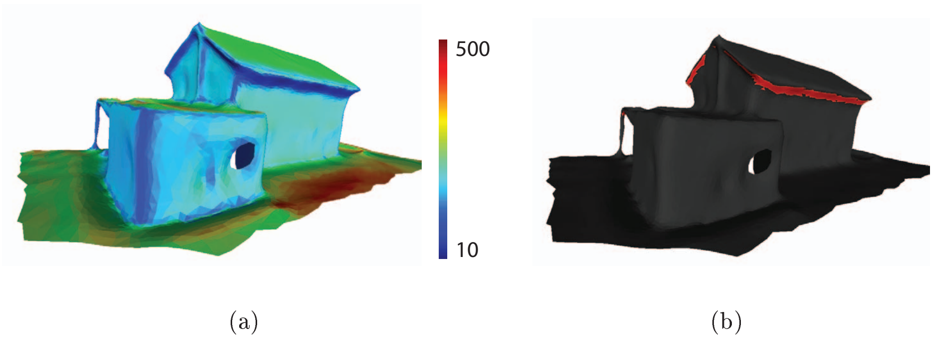
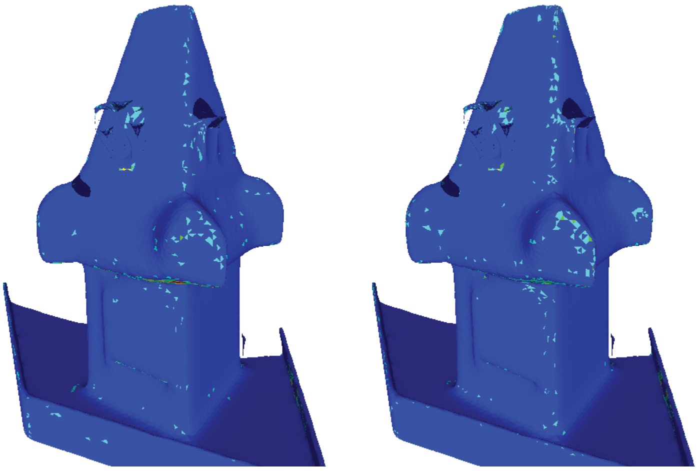


Figure 5.8: Redundancy of visibility. (a) Number of cameras that observe a certain triangle color-coded. Red parts are seen by more than 500 cameras and dark blue parts are seen by less than 10 cameras. (b) explicitly marks triangles that are seen by less than 10 cameras in red.

95% of all points are covered by selecting only 133 cameras from around 12 232 potential view points in $\mathcal{K}$. In the second experiment, we include our constraints as described in Algorithm 2 into the selection process. Since the additional constraints reduce the size of the solution set, we expect that the number of required cameras to reach 95% coverage, increases. However, we found that the number stays constant in this example, i. e. again 133 cameras are selected to reach the expected coverage. The resulting set of cameras is shown in Figure 5.7.

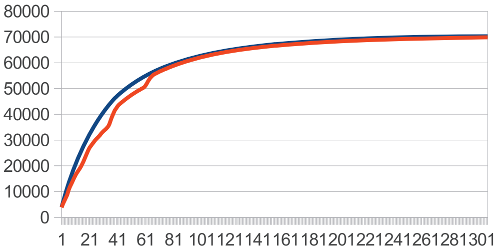
We run the same experiment also for the medieval tower. Here, our set of potential view points contains around 4 732 cameras. In the unconstrained case, 292 camera view points are selected whereas in the constrained case 322 view points are required to cover 95% of the points. Although the constraints cause an overhead of about 10%, the impact in practice is relatively small because the last 30 selected cameras only increase the coverage rate from 94.21% to 95%. Figure 5.9 shows the coverage for the constrained and unconstrained case after selecting 292 camera views. As the figures also express, the difference in coverage is very low.

Figure 5.10 shows how the constraints influence the evolution of the objective function $F(A^*)$. In the unconstrained case, the objective function is submodular whereas in the constrained case, the objective function is only monotone. The monotonicity shows that even in the constrained case a full coverage is obtained under the assumption that their exists a graph that connects all cameras.



(a) (b)

Figure 5.9: Final coverage with (a) constraints and (b) without constraints. Blue parts are completely covered, triangles where three of four clusters are covered are turquoise.



(a) objective function of medieval tower

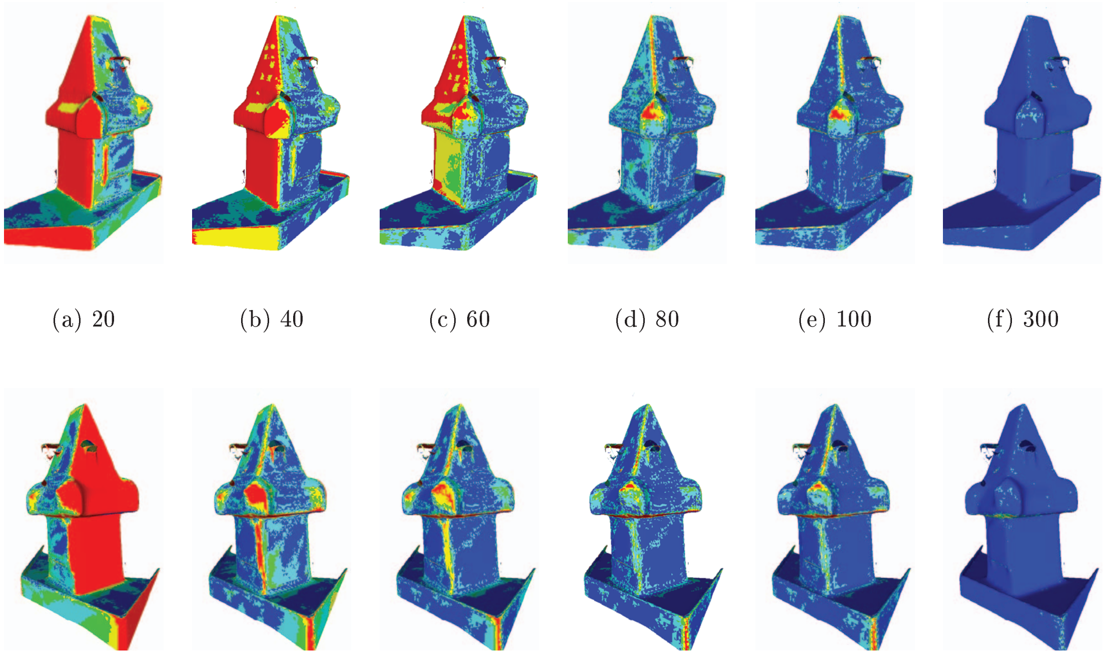
Figure 5.10: Values of objective function plotted against the number of selected cameras in the unconstrained (blue) and constrained case (red).

### 5.2.3 Greedy optimization

Optimizing our utility function with the greedy optimization scheme has various advantages which we will discuss in this section.

Since the greedy optimization is working in an iterative manner, we can stop the algorithm at any point in time and we obtain valid set of cameras, i.e. the greedy optimization is an anytime algorithm. Furthermore, the greedy optimization in conjunction with the submodular property of our objective selects cameras first, that increase the objective function at most (see Figure 5.10). Hence, we can stop the algorithm at a certain iteration when we obtain a solution that fulfills a certain stopping criterion, for example if 95% of the points are covered. Furthermore, the greedy optimization implicitly results in an importance ordering. Cameras selected first, often cover more parts than cameras that are selected at a later point. This is a nice property also for our interactive SfM method because if the images are acquired in the ordering obtained by the optimization, the first images already cover a large part of the scene. Figure 5.11 illustrates the coverage at different iterations of the greedy optimization.

To demonstrate the efficiency of the greedy optimization method, we perform the experiment of the previous section again but choosing cameras randomly. In the unconstrained case, the greedy approach selected 133 cameras to obtain the full coverage for the detached house. When selecting 133 cameras randomly from the set $\mathcal{K}$ on average 72.8% of the



(a) 20 (b) 40 (c) 60 (d) 80 (e) 100 (f) 300

(g) 20 (h) 40 (i) 60 (j) 80 (k) 100 (l) 300

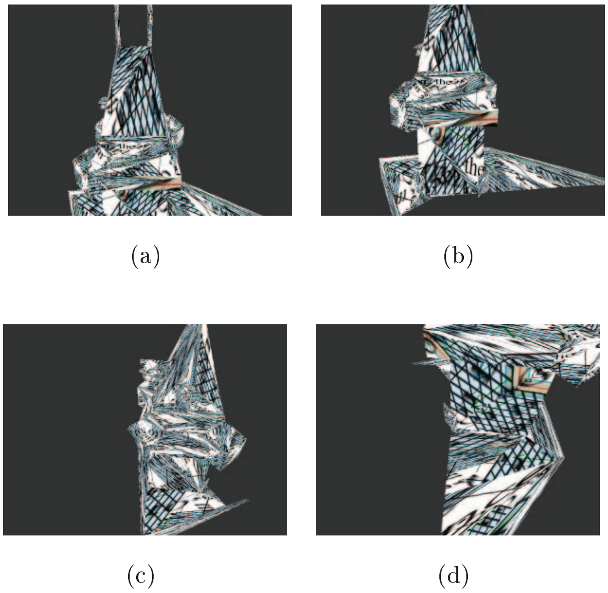
Figure 5.11: Evolution of coverage value $C_u$ of the medieval tower from two different view points. The coverage increases over time and already after 100 selected view points a large part is covered. Triangles where $C_u = 0$ are red, $C_u = 1$ are yellow, $C_u = 2$ are green, $C_u = 3$ are turquoise and $C_u \geq 4$ are blue.

points in $\mathcal{U}$ are completely covered. We additionally evaluated how many cameras we have to be select randomly to obtain a coverage rate of 95%. The result is that 418 cameras have to be selected on average, with a standard deviation of 38, to obtain the required coverage rate. Both experiments were repeated 100 times. This experiment shows that the greedy optimization method is valuable to minimize the number of required cameras.

### 5.2.4 Evaluation of Pairwise Constraints

In order to demonstrate that our pairwise constraint meets the requirements of a state-of-the-art SfM pipeline to obtain a single non-fragmented reconstruction, we run the following simulation. Given the a-priori mesh, we texture it randomly and render the model from the calculated view points. The random texture ensures that our assumption that the surface yields in equally spaced feature points holds. The rendered images serve as input to a standard SfM pipeline for unordered images. Figure 5.12 shows some rendered example view points of the medieval tower.

The epipolar graph in Figure 5.13 shows for the medieval tower that our proposed pairwise criterion is sufficient to generate an epipolar graph that allows to integrate all images into a common non-fragmented reconstruction. Furthermore, the constraints generate an



(a)

(b)

(c)

(d)

Figure 5.12: The a-priori mesh of the medieval tower rendered with random texture from the selected view points.

epipolar graph such that the images can be processed by our interactive SfM approach. This can be seen from the lower triangular part of the matrix. This part encodes the estimated epipolar geometries of an image to previously selected images. Because for each image at least a single epipolar geometry to a previously selected image was found, it fulfills the requirements of our interactive SfM pipeline.

### 5.2.5 Comparison to Direct Uncertainty Minimization

Since our method optimizes the accuracy only implicitly by selecting cameras from different viewing directions, we compare our method to an approach that minimizes the uncertainty of an observed point directly. As stated by Hollinger et al. [33], this is state-of-the-art for

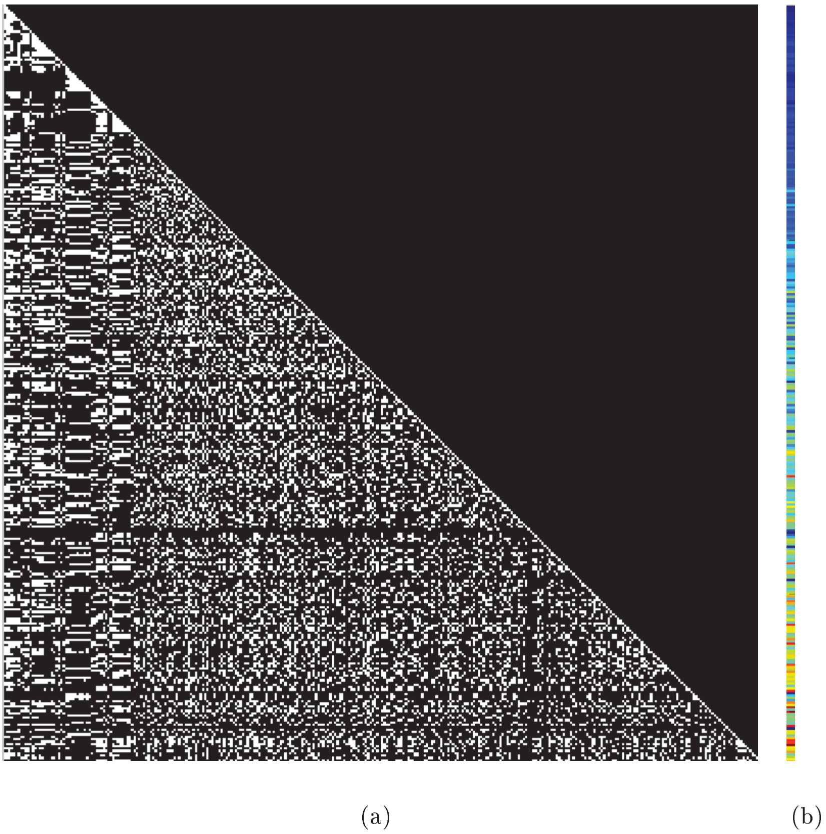
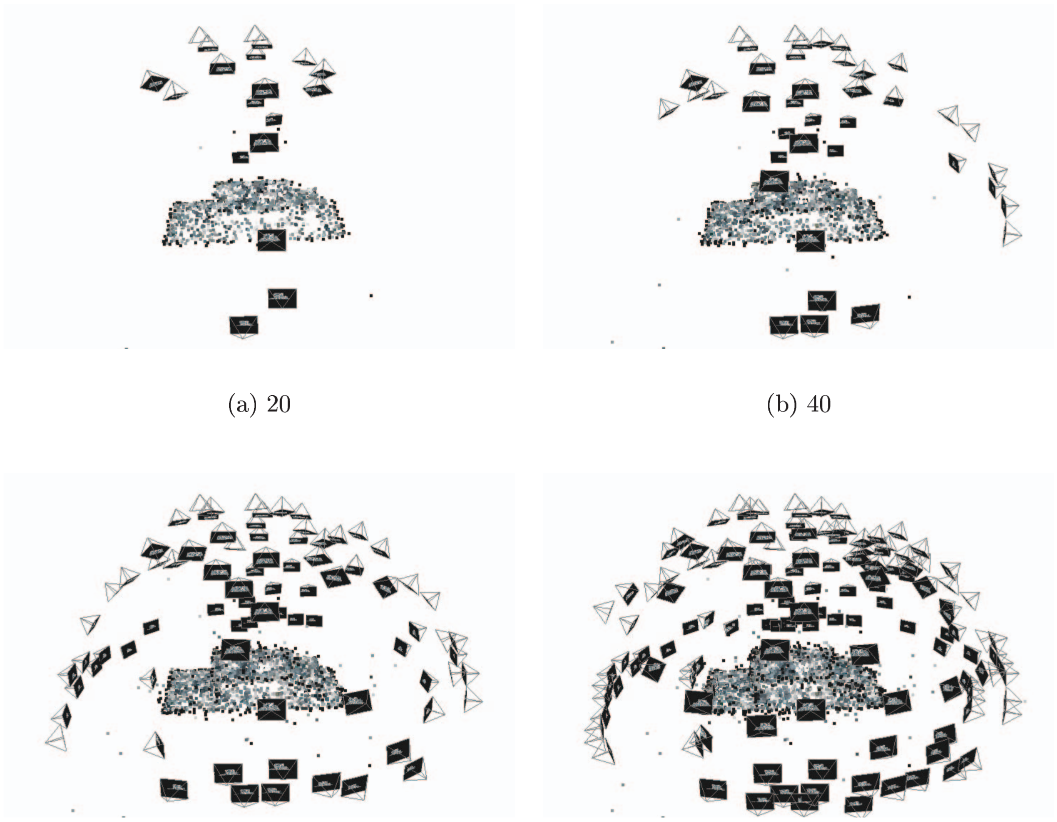


Figure 5.13: Epipolar graph of the synthetic experiments of the medieval tower.(a) Lower triangular part of the epipolar matrix. (b) The number of estimated epipolar geometries to previously selected images (color-coded). Red means a high connection to previously acquired images and blue indicates a low number.



(a) 20

(b) 40

(c) 80

(d) 152

Figure 5.14: Evolution of the interactive SfM result using the planned view points. All images can be added sequentially which shows that our pairwise overlap criterion is sufficient to get a single reconstruction.

planning methods.

The uncertainty of the position of a triangulated point is encoded in the covariance matrix. For example, Beder and Steffen [9] showed how the covariance matrix can be calculated under the assumption that the 2D matching uncertainty is Gaussian distributed and the cameras have a very low uncertainty. Given the covariance of a point, Wenhardt et al. [103] derive three accuracy criteria which are related to the determinant, the eigenvalues or the trace of the covariance matrix. Since the conclusion states that all criteria perform equally well under the assumption that cameras can only be positioned on a sphere around the object, we decided to minimize the trace of the covariance matrix. Another reason to minimize the trace of the covariance matrix is that this criterion is typically submodular

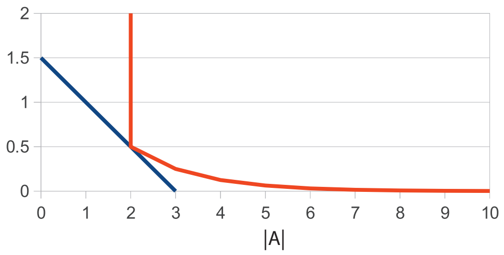


Figure 5.15: The uncertainty of a triangulated point is infinity if the point is seen by less than two cameras (red curve). In that case, we approximate the uncertainty by a linear function (blue).

as shown in [55]. Therefore, it can be directly used in our framework as an alternative objective function. Hence, our alternative objective function is

$$F'(A^*) = \sum_{u \in U} tr(cov(u, A^*)), \tag{5.5}$$

where $tr(\cdot)$ is the trace of the covariance matrix $cov(\cdot)$ of point $u$. In the following, we refer to this alternative objective as *direct uncertainty minimization*. The difficulty of minimizing this objective function is that the covariance of a 3D point can be calculated only if the point is visible in two or more images. If the point is not visible in any camera, its uncertainty is infinite. But even if the point is visible in a single camera, the uncertainty in the depth direction remains infinite and therefore the trace of the covariance is still infinite. This property violates the submodularity property of diminishing returns:

$$\forall A \subseteq B \subseteq S, s' \neq B : F(A \cup \{s'\}) - F(A) \geq F(B \cup \{s'\}) - F(B). \tag{5.6}$$

Starting with an empty set $A = \{\emptyset\}$ and adding a single camera $s'$ does not improve $F(A)$: $F(A) - F(A \cup \{s'\}) = 0$. Whereas if $|B| = 1$ and adding a second camera $s'$, this improves $F(B)$ from infinity to a bounded uncertainty: $F(B) - F(B \cup \{s'\}) > 0$. Hence, to bypass this problem, we modify Equation 5.5 as follows:

$$F'(A^*) = \sum_{u \in U} G(A^*), \tag{5.7}$$

where

$$G(A^*) = \begin{cases} \delta|A_v^*| + b & \text{if } |A_v^*| < 2 \\ tr(cov(u, A^*)) & \text{else} \end{cases} \tag{5.8}$$

and $A_v^*$ is the set of cameras where $u$ is visible in. This means if the number of cameras that observe $u$ is smaller than two, we approximate the uncertainty by a linear function. The negative slope $\delta$ must be larger than the maximum gradient that can occur in the case of $|A_v^*| \geq 2$. We can easily find such a $\delta$ by estimating the worst-case accuracy when triangulating a point with two cameras with a Monte Carlo simulation and then fitting a line through this point and the point $(3, 0)$ which is the ideal point where three cameras are sufficient to determine the points location with zero uncertainty, which will be impossible. Figure 5.15 visualizes Equation 5.8. However, each linear function which passes through $x = 2$ with a larger gradient than the maximum that can occur in the case $|A_v^*| \geq 2$ is valid.

We repeated the experiment with the detached house using our proposed approach and the direct uncertainty optimization. We stopped both algorithms after selecting 150 camera positions. Figure 5.16 shows the performance values for both algorithms. Figure 5.16(a) illustrates that after the selection of 150 cameras approximately the same number of points are not reconstructable, i.e. they are either never seen by any camera or by just a single camera. The histogram also shows that our approach finds a distribution of camera views such that more than 12 000 points are seen by 9 or more cameras whereas this value for the direct uncertainty optimization is only 10 000. This demonstrates that the overall redundancy obtained by our method is higher than optimizing Equation 5.8. This higher redundancy is also reflected in the point-wise accuracy that is shown in Figure 5.16(b) and Figure 5.16(c). These plots illustrate the mean trace of the covariance matrix of all points that are seen by two or more cameras, i.e. $|A_v^*| > 1$. In case of the direct uncertainty optimization, the gradient of the linear function in the case that $|A_v^*| < 2$ is set to a high negative value. This favors the greedy optimization to select cameras first, such that each point is seen at least two times. Therefore, the mean accuracy of the individual reconstructable points at the beginning is relatively high. Nevertheless, the mean accuracy of all points after selecting 150 cameras is twice as much higher than the accuracy obtained by our method.

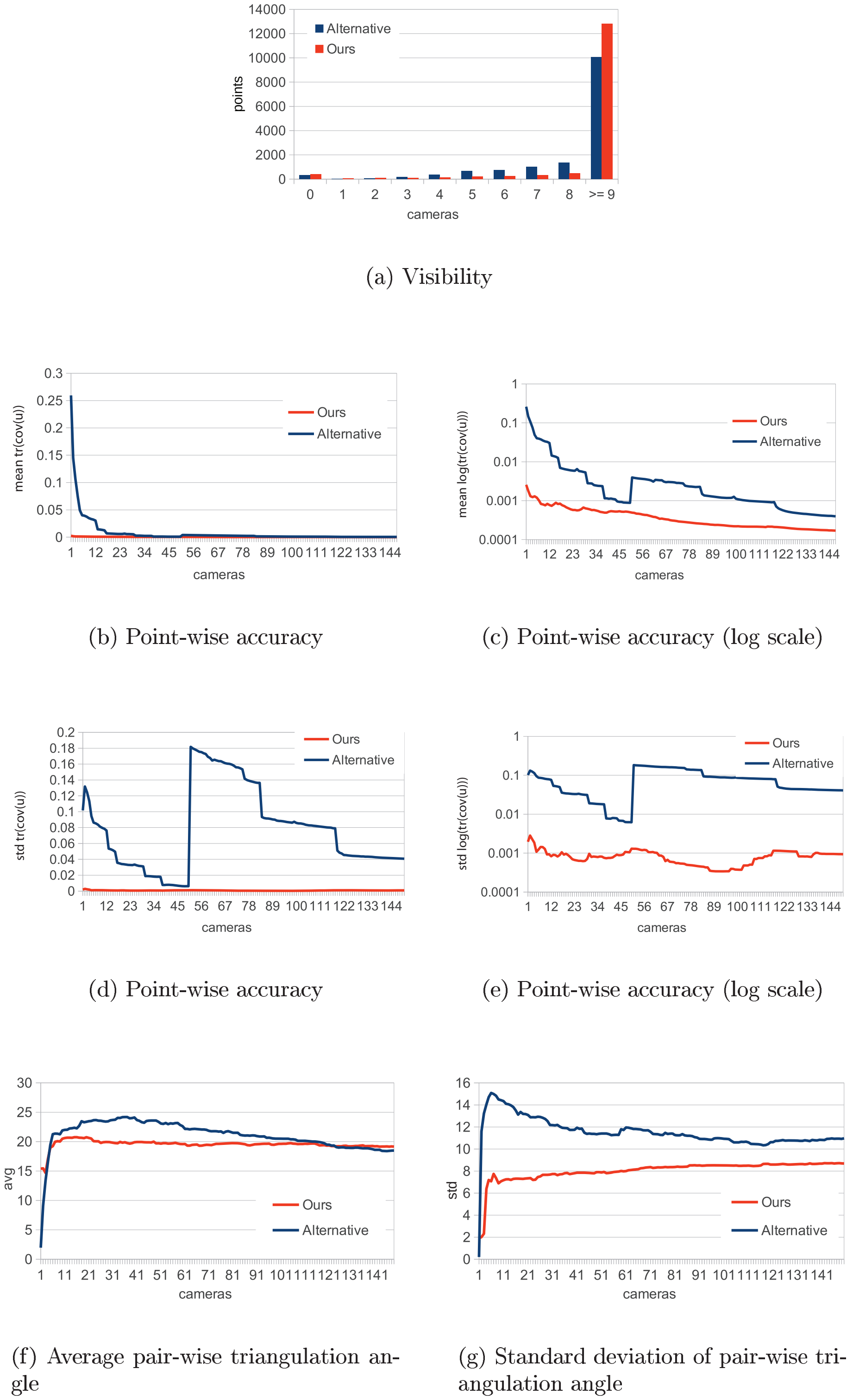
Since we are interested in reconstructions whose accuracy is equally distributed over the scene, the standard deviation of the accuracy should be low. Figure 5.16(d) and Figure 5.16(e) demonstrate that the standard deviation is much higher when performing the direct uncertainty optimization. When stopping the algorithm at 150 selected cameras, the standard deviation of the accuracy is two orders of magnitude smaller using our approach. This demonstrates that the selection of cameras that are equally distributed locally around a point reduces the mean uncertainty as well as the variance of the uncertainty.

Beside an equal distribution of the uncertainty, we are also interested in a set of cameras that is well suited for pair-wise feature matching. One factor for reliable feature matching is that the view distortions between the images should be small. To evaluate the degree of the view distortion, we perform the following experiment. Given a point $u$ and all selected cameras $\mathcal{A}_v^*$ that observe $u$, we want to determine the matching ordering of the cameras such that the sum of the pair-wise triangulation angles and therefore the pair-wise view distortion is as small as possible. Figure 5.17 shows an example of such an ordering. Finding the optimal ordering is an instance of the traveling salesman problem where the nodes are the cameras and the edge weights are the pair-wise triangulation angles. Since this is not efficiently solvable, we approximate the search by choosing each camera as starting point and determine the shortest path that includes all cameras using Dijkstras [17] algorithm. The resulting path is the ordering where the pair-wise view distortion is as small as possible. The best case is, from a feature matching point of view, if all edges in the path are zero, i. e. all cameras have the same viewing direction. To provide a quantitative result for each camera network, we calculate for each point $u$ the shortest path $P(u)$ and calculate the mean path length:

$$\frac{1}{m} \sum_{u \in \mathcal{U}} \sum_{e \in P(u)} w(e), \tag{5.9}$$

where $e$ is an edge of the shortest path $P(u)$ and $w(e)$ is the weight, i.e. the triangulation angle of the edge; $m$ gives the overall number of summed up edges. Therefore this formula calculates the mean path length.

Figure 5.16(f) shows how this value changes in relation to the number of selected cameras. In our approach the value is nearly constant at 19 degrees whereas the value of the alternative accuracy optimization behaves linearly. At the beginning the average pair-wise triangulation is relatively large and gets smaller the more cameras are added to



(a) Visibility

(b) Point-wise accuracy

(c) Point-wise accuracy (log scale)

(d) Point-wise accuracy

(e) Point-wise accuracy (log scale)

(f) Average pair-wise triangulation angle

(g) Standard deviation of pair-wise triangulation angle

Figure 5.16: Comparison between our approach and the objective function of Equation 5.8. (a) histogram of redundancy after selection of 150 cameras. In both approaches most of the points are visible in more than 9 cameras. (b) point-wise accuracy evaluation. As expected the point-wise accuracy for all points that are seen by two or more cameras is a bit lower when optimizing for the accuracy directly. However, the difference is low. (f) mean triangular distance between cameras

$A^*$. This again shows that our selected cameras are better distributed in the scene. We also evaluate the standard deviation of the average triangulation angle which is shown in Figure 5.16(g). We observe that the standard deviation from the mean triangulation angle is smaller with our method which demonstrates that the cameras are more equally distributed in the scene.

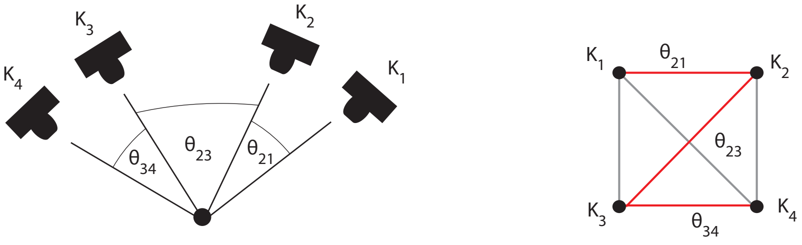


Figure 5.17: Optimal ordering of cameras such that the pair-wise view distortion is as small as possible. In this case, the following matching order minimizes the sum of the pair-wise view distortions: $\{K_1, K_2, K_3, K_4\}$.

The impact of the more equally distributed images is also measurable in the synthetic experiment. Similar to the evaluation of the pair-wise constraint in Section 5.2.4, we synthetically render 150 images of both camera networks and reconstruct them using a state-of-the-art SfM method. The epipolar graph obtained by the images of our approach contains 5 813 edges, i. e. 38.75 edges (fundamental matrices) per image on average. In contrast when using the direct uncertainty minimization, the epipolar graph contains only 5 346 edges which are roughly 8% less compared to our approach. We also observe an improved number of measurements per 3D point when using our approach (Figure 5.18). The reconstructions of both methods contain about 40% of 3D points that are visible only in two images. But the reconstruction obtained by our camera network contains 25% points that are visible in nine or more cameras compared to 20% of the alternative approach. The finding that our method results in more high-redundant 3D points coincides with the result of the planning process. During the planning process our approach also generates more high-redundant points than the alternative approach (see Figure 5.16(a)).

### 5.2.6 Real-world Experiment with a MAV

The synthetic experiments that we performed before assume that the acquired images are taken at the exact calculated camera positions. However, when acquiring a set of images

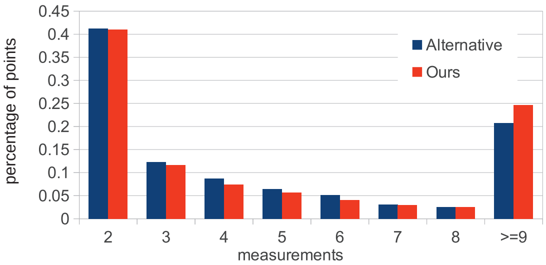


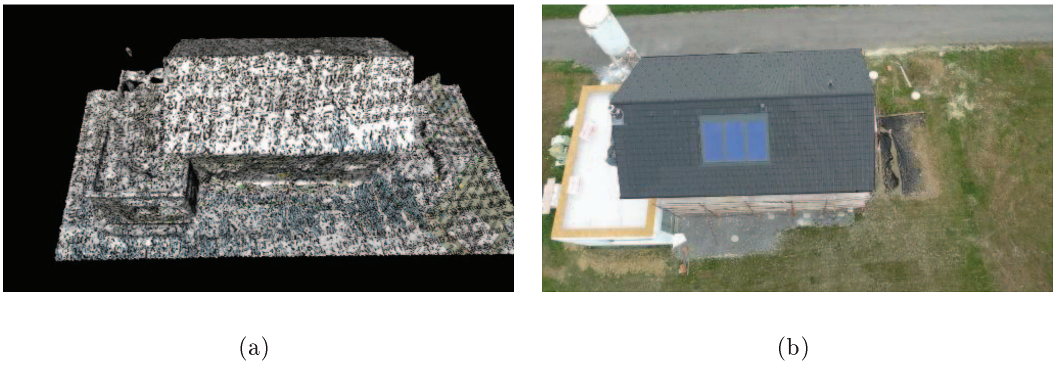
Figure 5.18: Distribution of measurements per point. The reconstructions of both camera networks contain about 40% points that are only visible in two images. However, the reconstruction of the camera network obtained by our approach contains 25% points that are visible in nine or more cameras whereas the reconstruction of the alternative approach contains only 20% of such redundant points. This supports our argument that our approach allows better feature matching.

with a Global Positioning System (GPS) controlled vehicle like an MAV, inaccuracies occur at different steps. First, in order to generate a camera plan where the camera positions are given as GPS coordinates, the prior mesh where the planning is based on, has to be geo-referenced, i.e. the coordinates of the mesh are also given in GPS coordinates. An error in the geo-referencing directly leads to erroneous camera positions. The second source of inaccuracy is the absolute position uncertainty of a standard GPS receiver which is about 5 to 10 meters. This can be potentially reduced to 1 to 2 meters by incorporating additional sensors like an IMU or using a differential GPS. Nevertheless, for the high-resolution reconstruction of small objects of interest like a detached house whose size is about 10 to 15 meters, the positioning with standard GPS is too low to guarantee that the calculated and acquired images are identical. Beside the errors in the translational positioning, rotational errors have a large impact. Especially a positioning error at the yaw axis which is controlled with a magnetic compass on a MAV, might change the view point drastically. In the worst case, the camera does not even observe the object of interest anymore.

In order to demonstrate how important an accurate positioning is, we performed the following experiment. We reconstructed a detached house from 274 images using SfM, densified the result with PMVS [22] and extracted a surface mesh using the Poisson surface reconstruction [46]. We used [101] to geo–reference our mesh. For view planning, we set the GSD to $6\,mm$ which results in a camera distance $d = 15\,m$. The camera we used for image acquisition is a Panasonic DMC-LX3 still image camera, which has a resolution of 10Mpx and a 24 $mm$ lens.

By running our view–planning algorithm, we obtain 133 camera poses that are approached autonomously by an MAV. Given the 133 acquired images, the applied SfM pipeline reconstructs 20,762 feature points. However, only 81 of the 133 camera positions can be integrated into common reconstruction. We believe that this happens because of the positioning inaccuracies. The example in Figure 5.19 shows that the expect view point and the one which has been acquired by the MAV. In horizontal direction both images differ by around 12%.

To attenuate this problems we propose two ideas. First, the positioning accuracy has to be increased by a more accurate GPS for example. Another possibility would be to register the current image against the prior knowledge mesh to get a position that is closer to the pre-calculated camera position. However, since we are dealing with an MAV that is moving in turbulent air, there will always be an uncertainty in the positioning. Hence, our second idea is to consider the expected positioning accuracy already during the view planning. The idea here is to rate the potential view points according to the robustness to location changes. For example, a view point that observes only a small part of the object is more susceptible for localization errors than a view that shows the overall object. Figure 5.20 shows two views where one is stable against view point changes but the other one will change drastically if the view point changes only slightly. By favoring stable views, the probability that the reconstruction splits into different parts will decrease.



(a)

(b)

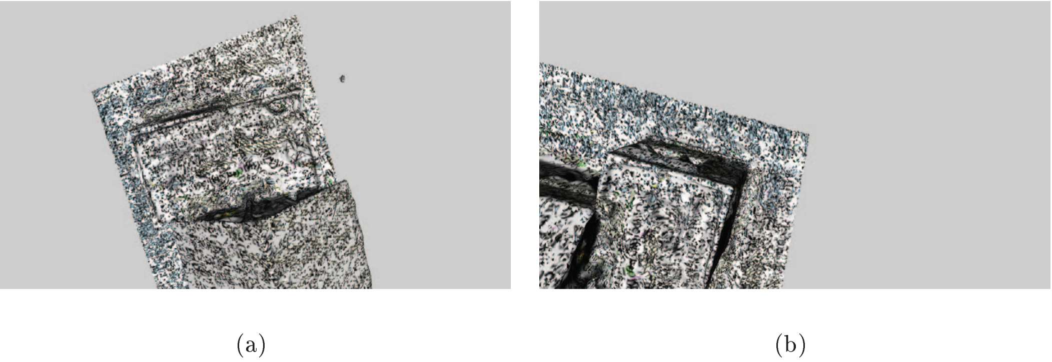
Figure 5.19: Difference between expected and acquired view. The real and the expected view are shifted around 12% of the image width.

## 5.3 Conclusion and Future Work

In this chapter, we presented a view planning approach that takes care of the requirements of SfM. These are, on the object side, redundancy and a a triangulation angle that allows feature matching. On the image side, our view planning ensures overlapping images and maximum viewing angles such that an epipolar graph is generated that is not fragmented. We integrated these constraints into submodular optimization problem which allows us to efficiently find a small set of cameras that are suited for a SfM reconstruction.

We found that the formulation as a submodular optimization problem and solving the problem with a greedy approach has several advantages. First, the greedy optimization is an anytime algorithm, i.e. even if the algorithm is stopped before the maximum value of the objective function is reached, the selected set of view points can be used to obtain a partial but a single connected reconstruction. Second, the iterative selection gives an ordering of the view points: view points selected at the beginning are more important for the exploration whereas later selected views are more important to guarantee the redundancy for each point. And finally, intermediate results like coverage can be visualized during the selection process. This helps the user understand where view points are important and which parts are difficult to capture.

Our proposed method is a first starting point for large-scale view planning for automated mobile image acquisition. As we have shown in the last experiment there are still issues to solve in the future. For example, the robustness to positioning errors of the camera with respect to the calculated camera position has to be increased as the experiment



(a)

(b)

Figure 5.20: Stable vs. unstable view points.

in Section 5.2.6 shows. Furthermore, our criterion for full coverage is a very strict criterion and often requires a large number of cameras as we have seen in the experiments. Hence, one could integrate also add additional criteria for optimization like the uncertainty of the triangulated point. Finally, the combination of the interactive SfM with this view planning method could be beneficial to consider additional impact factors like the texture of the environment in the planning.

# Chapter 6

# Conclusion

The presented thesis deals with the problem of reconstructing large-scale scenes with high-resolution still images using Structure-from-Motion (SfM). The thesis concentrates on applications where images can be deliberately acquired for the task of 3D reconstruction like scene documentation or geological surveys. This implies that the image acquisition process is a part of the reconstruction pipeline. In this theseis, two methods were introduced that exploit the property that the image acquisition is controllable.

The key findings of our work are presented in Section 6.1 and an outlook on further potential research directions in image-based reconstruction is given in Section 6.2.

## 6.1 Summary

Although SfM is a well-investigated research topic in computer vision, there exists still problems when using SfM for applications like architectural reconstruction or scene documentation in 3D where images are typically acquired deliberately for the reconstruction process. The property that the images are acquired especially for the reconstruction task, raises problems but the same property can be exploited to attenuate the problems as it is shown in this thesis. Typical problems that often occur are incomplete, fragmented or inaccurate reconstructions. The main reason for these errors is that the acquired input dataset is not sufficient for the reconstruction. Hence, to make SfM more reliable for the aforementioned applications, the reconstruction and the image acquisition process have to be coupled. Therefore, this thesis deals with the question how the image acquisition and the reconstruction can be interleaved to make SfM more reliable.

The first introduced approach replaces the standard feed-forward processing pipeline by an interactive, closed-loop method that adds images incrementally and in real-time. This method supports a user in three ways. First, the user instantly gets a feedback if an acquired images can be used to extend the existing reconstruction. Second, the surface mesh, that is incrementally extracted from the sparse triangulated feature points, visualizes the reconstruction such that even non-expert users can interpret the reconstruction's quality. And finally, quality parameters like the ground sampling distance and the redundancy can be visualized on the extracted surface.

The users that acquired image datasets with the support of the interactive SfM method, experienced that the instant feedback is very helpful, especially when the scene is geometrically complex or in case it is not obvious if the texture is sufficient for automatic image alignment. The possibility to check the reconstruction's quality on-site is a great benefit when performing reconstructions for commercial reasons because it reduces the probability that a re-take of the images is required. Furthermore, the system is not only helpful for expert users but even more for non-experts. Non-experts often acquire image datasets that are not suited for a reconstruction. In the batch-based processing, the user experiences only the final result which potentially does not meet the users requirements. In that case, a non-expert user often has no idea about the reason for the error. In contrast, the novel interactive SfM visualizes the contribution of each individual image for the whole reconstruction. Therefore, the user gets quickly an intuition about the image parameters that lead to a successful reconstruction.

From a technical point of view, the interactive SfM is based on two novel methods which both can be also used for other tasks. The novel image-based localization method registers low-resolution images to SfM results that are obtained from high-resolution images much faster and more reliable than existing methods. Beside it application in our interactive SfM approach, the method can be also used for registering low-quality video streams to 3D point clouds obtained from high-resolution data. In the future, more and more low-quality cameras will be used in devices like mobile phones or wearable gadgets. Hence, we expect that for the localization of such images, our approach also delivers better a performance then existing approaches.

The second important contribution of the interactive SfM is the fully incremental mesh extraction from sparse and noisy 3D points. Thanks to the simplicity and the low computational complexity, this approach is suited for applications in robotics or in Augmented Reality where the surface extraction can now be a lightweight process. Although the extracted mesh is not as detailed as a dense 3D reconstruction, it delivers significantly more information about a scene than a plain and sparse point cloud because the mesh considers 3D points as well as visibility information. In the future, the surface mesh can be used in AR for occlusion handling, or in robotics for path planning. Furthermore, the surface information could be used in many applications as prior knowledge. For example, the mesh can be used to easily segment the scene in planar regions. This information can be used for example to initialize a multi-directional planesweep approach like [23].

The second proposed approach in this thesis, is a view planning method. Many existing view planning approaches for image-based reconstruction mainly consider the completeness and the accuracy of the reconstruction while the problem of wide-baseline feature matching is often ignored. However, we found that in case of reconstructions from wide-baseline images, most issues are related to insufficient feature matches. Hence, the approach that is presented in this thesis considers the spatial distribution of the images such that problems related to feature matching are avoided. The formulation as a multi-cover problem with spatial constraints allows the use of efficient optimization schemes to determine a small set of camera views that is suited for SfM. The experimental evaluation shows that the proposed formulation creates camera sets that are more equally distributed over the scene, compared to related algorithms that only take into account the accuracy and the completeness of the reconstruction. However, the real-world experiment with a Micro Aerial Vehicle has shown that further research has to improve the robustness of the camera network such that small deviations from the desired camera position do not degrade the whole reconstruction. A further, more conceptual problem of off-line view planning for passive image-based reconstruction is the uncertainty that is caused by object's texture. Hence, in future research a goal is to combine the interactive SfM with the view planning approach to build a Next-Best-View method that takes into account not only the geometry of the object but also the texture.

To sum up, the integration of the image acquisition into the SfM pipeline by performing (a) SfM interactively and (b) planning the view points makes the SfM result more predictable. Since reliability is one of the major impact factors for the success of a technology, our work increases the chance that SfM will be used in the future for more and more applications. Nevertheless, for realizing a full system, several problems have to be solved in future research which will be discussed in the next section.

## 6.2 Outlook

We build numerous 3D reconstructions for various applications during the work on this thesis and meanwhile different types of issues which basically can be classified into two groups occurred. The first class of problems is related to the creation of 3D data and the second class is concerned with the processing of 3D data. In the following, we will have a closer look on both problem classes.

We presented two methods that ease the image acquisition for SfM but right now, it is still not completely automated. For example, for using SfM in the large with MAVs, the image acquisition should be performed completely autonomously. To realize such a system, the combination of our view planning with the interactive SfM into a Next-Best-View system would be a first starting point. Hence, the actual reconstruction result of the interactive SfM can be used to plan a camera path. This also allows to incorporate additional knowledge about the scene like the type of texture which is not available when performing an off-line view planning.

Another issue for some applications is the absolute accuracy of the obtained image-based reconstruction, for example, when measuring objects like windows or doors in 3D. Here, the main issue is the large number of parameters that are typically involved in a reconstruction setting, which are for example the image features used, the parameters for the bundle adjustment, the quality of the texture, the geometric complexity of the scene, etc. Once the images are taken and the camera positions are calculated, it is possible to estimate the uncertainty for camera and scene parameters. However, a practical application that supports the user on site during the image acquisition by giving hints where to take images such that a certain accuracy can be guaranteed, is still missing.

Current SfM pipelines work quite well if the scene is static and well-textured. However, a large number of applications cannot guarantee these requirements. For example, when performing SfM on scenes that contain a lot of vegetation, many of todays state-of-the-art descriptors completely fail due to the self-similarity of the structure. Furthermore, vegetation like trees is moving in the wind which violates the assumption of a static scene.

Another issue are scene parts that are non-Lambertian like glass and mirrors. Especially in the reconstruction of architectural scenes, reflecting surfaces are a major problem for nowadays SfM pipelines.

A solution for this problem could be the fusion of different sensors for 3D reconstruction. The number of sensors for 3D perception combined with the ongoing miniaturization and increasing computational power will make the integration of several sensors possible in the near future. The impact of a new 3D sensor on research has been impressively shown by the Kinect sensor. But not only new sensors can be integrated to form novel systems, but also methods of computer vision that are developed independently can be combined. For example, optical flow and sparse SfM from wide-baseline images can be fused to obtain more accurate results.

The increasing number of sensors combined with higher resolutions brings us to the second type of problem which is the processing of data. Even existing sensors like the Kinect create up to 210 000 3D measurements per image with a framerate of 30 Hz. Saving this amount of data is demanding but even more important, processing and interpretation of the data is very challenging. The aim of many applications is to extract more high-level information. For example one application is to match a CAD plan with the actual reconstruction to identify differences between both. Hence, for this task a CAD like representation of a building has to be derived. In general, for many applications the full data is not required but only a semantically meaningful approximation. Hence, one of the major research directions in the future is to find semantical descriptions of a 3D scene for various tasks. So a first step to a full semantic interpretation is to bring methods for 2D interpretation to 3D. But in further consequence, semantics obtained from the 2D representation can also improve the 3D reconstruction results.

# Appendix A

# List of Publications

## A.1 Journal Publication

- S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. Augmented reality for construction site monitoring and documentation. *Special Issue: Application of Augmented Reality, Proceedings of the IEEE*, 102(2):137–154, 2014

## A.2 Conference Publications

- C. Hoppe, M. Klopschitz, M. Donoser, and H. Bischof. Incremental surface extraction from sparse structure-from-motion point clouds. In *British Machine Vision Conference (BMVC)*, 2013

- C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online feedback for structure-from-motion image acquisition. In *British Machine Vision Conference (BMVC)*, 2012

- C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, and S. Kluckner. Photogrammetric camera network design for micro aerial vehicles. In *Computer Vision Winter Workshop (CVWW)*, 2012

- T. Holzmann, C. Hoppe, S. Kluckner, and H. Bischof. Geometric abstraction from noisy image-based 3D reconstructions. In *Proceedings of the 38th Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM)*, 2014

- A. Irschara, C. Hoppe, H. Bischof, and S. Kluckner. Efficient structure from motion with weak position and orientation priors. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 21–28, 2011

- S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr. FlyAR: Augmented reality supported micro aerial vehicle navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):560–568, 2014

- S. Zollmann, D. Kalkofen, C. Hoppe, S. Kluckner, H. Bischof, and G. Reitmayr. Interactive 4D overview and detail visualization in augmented reality. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013

- A. Wendel, C. Hoppe, H. Bischof, and F. Leberl. Automatic fusion of partial reconstructions. In *Annals of the International Society for Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 2012

- J.-S. Morard, M. Klopschitz, S. Kluckner, C. Hoppe, and H. Bischof. 3D change detection by inverted voxel spaces. In *Computer Vision Winter Workshop (CVWW)*, 2012

- S. Kluckner, J. Hatzl, M. Klopschitz, M. Jean-Severin, C. Hoppe, S. Zollmann, H. Bischof, and G. Reitmayr. Image-based as-built site documentation and analysis - applications and challenges. In *Annual Symposium of the German Association for Pattern Recognition (DAGM), Workshop Computer Vision in Applications*, 2012

- M. Maurer, M. Rumpler, A. Wendel, C. Hoppe, A. Irschara, and H. Bischof. Georeferenced 3D reconstruction: Fusing public geographic data and aerial imagery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3557–3558, 2012

- S. Kluckner, J.-A. Birchbauer, C. Windisch, C. Hoppe, A. Irschara, A. Wendel, S. Zollmann, G. Reitmayr, and H. Bischof. AVSS 2011 demo session: Construction site monitoring from highly-overlapping MAV images. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 531–532, 2011

# Appendix B

# Acronyms

**SfM** Structure-from-Motion

**6DoF** six-degrees-of-freedom

**SIFT** Scale Invariant Feature Transform

**MAV** Micro Aerial Vehicle

**AR** Augmented Reality

**SLAM** Simultaneous Localization and Mapping

**DT** Delaunay Triangulation

**GSD** Ground Sampling Distance

**GIS** Geographic Information System

**GPS** Global Positioning System

# Bibliography

[1] Computational Geometry Algorithms Library. `http://www.cgal.org`.

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[3] A. Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible*. PhD thesis, Massachusetts Institute of Technology, 2009.

[4] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

[5] K. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Pub., 1996.

[6] D. Attali, J.-D. Boissonnat, and A. Lieutier. Complexity of the delaunay triangulation of points on surfaces the smooth case. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 201–210. ACM, 2003.

[7] C. Bailer, M. Finckh, and H. P. A. Lensch. Scale robust multi view stereo. In *European Conference on Computer Vision (ECCV)*, pages 398–411, 2012.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision Image Understanding*, 110(3):346–359, 2008.

[9] C. Beder and R. Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Pattern Recognition*, volume 4174 of *Lecture Notes in Computer Science*, pages 657–666. Springer, 2006.

[10] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, 2011.

[11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

[12] O. Chum and J. Matas. Randomized RANSAC with $t_{d,d}$ test. In *Image and Vision Computing*, pages 448–457, 2002.

[13] O. Chum and J. Matas. Matching with prosac - progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 220–226, 2005.

[14] A. Dalalyan and R. Keriven. $l_1$-penalized robust estimation for a class of inverse problems arising in multiview geometry. In *Advances in Neural Information Processing Systems*, pages 441–449, 2009.

[15] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):1052–1067, 2007.

[16] B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, (6):793–800, 1934.

[17] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[18] E. Eade and T. Drummond. Scalable monocular SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 469–476, 2006.

[19] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf. The shuttle radar topography mission. *Reviews of Geophysics*, 45(2), 2007.

[20] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartogra phy. *Communications of the ACM*, 24(6):381–395, 1981.

[21] T. Fujito and H. Kurahashi. A better-than-greedy algorithm for k-set multicover. In T. Erlebach and G. Persinao, editors, *Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*, pages 176–189. Springer Berlin Heidelberg, 2006.

[22] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1362–1376, 2010.

[23] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[24] W. Gellert and V. N. R. Company. *The VNR concise encyclopedia of mathematics*. Van Nostrand Reinhold Co., 1977.

[25] M. Golparvar-Fard, F. Peña-Mora, and S. Savarese. Integrated sequential as-built and as-planned representation with D4AR tools in support of decision-making tasks in the AEC/FM industry. *Journal of Construction Engineering and Management*, 137(12):1099–1116, 2011.

[26] G. Graber, T. Pock, and H. Bischof. Online 3D reconstruction using convex optimization. In *International Conference on Computer Vision Workshops*, pages 708–711, 2011.

[27] L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The city of sights: Design, construction, and measurement of an augmented reality stage set. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2010.

[28] S. Haner and A. Heyden. Covariance propagation and next best view planning for 3D reconstruction. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 545–556. Springer Berlin Heidelberg, 2012.

[29] G. A. Hansen, R. W. Douglass, and A. Zardecki. *Mesh Enhancement*. Imperial College Press, 2005.

[30] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *International Conference on Computer Vision (ICCV)*, pages 469–476, 1998.

[31] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[32] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative evaluation of binary features. In *European Conference on Computer Vision (ECCV)*, pages 759–773, 2012.

[33] G. A. Hollinger, B. Englot, F. Hover, M. Urbashi, and G. S. Sukhatme. Uncertainty-driven view planning for underwater inspection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4884–4891, St. Paul, MN, 2012.

[34] T. Holzmann, C. Hoppe, S. Kluckner, and H. Bischof. Geometric abstraction from noisy image-based 3D reconstructions. In *Proceedings of the 38th Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM)*, 2014.

[35] C. Hoppe, M. Klopschitz, M. Donoser, and H. Bischof. Incremental surface extraction from sparse structure-from-motion point clouds. In *British Machine Vision Conference (BMVC)*, 2013.

[36] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online feedback for structure-from-motion image acquisition. In *British Machine Vision Conference (BMVC)*, 2012.

[37] C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, and S. Kluckner. Photogrammetric camera network design for micro aerial vehicles. In *Computer Vision Winter Workshop (CVWW)*, 2012.

[38] T. S. Huang and O. D. Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 11(12):1310–1312, 1989.

[39] P. J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[40] A. Irschara. *Scalable Scene Reconstruction and Image Based Localization*. PhD thesis, Graz University of Technology, 2012.

[41] A. Irschara, C. Hoppe, H. Bischof, and S. Kluckner. Efficient structure from motion with weak position and orientation priors. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 21–28, 2011.

[42] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2599–2606, 2009.

[43] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision (ECCV)*, pages 304–317, 2008.

[44] B. Joe. Construction of three-dimensional delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2):123–142, 1991.

[45] F. Kahl. Multiple view geometry and the l-infinity norm. In *IEEE International Conference on Computer Vision (ICCV)*, Beijing, 2005.

[46] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, 2006.

[47] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, 2007.

[48] M. Klopschitz, A. Irschara, G. Reitmayr, and D. Schmalstieg. Robust incremental structure from motion. In *Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2010.

[49] S. Kluckner, J.-A. Birchbauer, C. Windisch, C. Hoppe, A. Irschara, A. Wendel, S. Zollmann, G. Reitmayr, and H. Bischof. AVSS 2011 demo session: Construction site monitoring from highly-overlapping MAV images. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 531–532, 2011.

[50] S. Kluckner, J. Hatzl, M. Klopschitz, M. Jean-Severin, C. Hoppe, S. Zollmann, H. Bischof, and G. Reitmayr. Image-based as-built site documentation and analysis - applications and challenges. In *Annual Symposium of the German Association for Pattern Recognition (DAGM), Workshop Computer Vision in Applications*, 2012.

[51] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2969–2976, 2011.

[52] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(12):2079–2088, 2007.

[53] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):147–159, 2004.

[54] M. Krainin, B. Curless, and D. Fox. Autonomous generation of complete 3D object models using next best view manipulation planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[55] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Selecting observations against adversarial objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[56] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *International Conference on Computer Vision (ICCV)*, 2007.

[57] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *European Conference on Computer Vision (ECCV)*, 2012.

[58] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-DoF localization in large-scale environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[59] V. Litvinov and M. Lhuillier. Incremental solid modeling from sparse and omnidirectional structure-from-motion data. In *British Machine Vision Conference (BMVC)*, 2013.

[60] M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[61] D. Lovi, N. Birkbeck, D. Cobzas, and M. Jaegersand. Incremental Free-Space Carving for Real-Time 3D Reconstruction. In *Fifth International Symposium on 3D Data Processing Visualization and Transmission(3DPVT)*, 2010.

[62] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJVC)*, 60(2):91–110, 2004.

[63] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[64] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[65] P. Maur. Delaunay triangulation in 3D. Technical Report, 2002.

[66] M. Maurer, M. Rumpler, A. Wendel, C. Hoppe, A. Irschara, and H. Bischof. Georeferenced 3D reconstruction: Fusing public geographic data and aerial imagery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3557–3558, 2012.

[67] J.-S. Morard, M. Klopschitz, S. Kluckner, C. Hoppe, and H. Bischof. 3D change detection by inverted voxel spaces. In *Computer Vision Winter Workshop (CVWW)*, 2012.

[68] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[69] C. Mostegel, A. Wendel, and H. Bischof. Active monocular localization: Towards autonomous monocular exploration for multirotor mavs. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[70] P. Moulon, P. Monasse, R. Marlet, et al. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision (ICCV)*, 2013.

[71] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505, 2010.

[72] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[73] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011.

[74] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(6):756–777, 2004.

[75] D. Nister, F. Kahl, and H. Stewénius. Structure from motion with missing data is NP-hard. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[76] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[77] Q. Pan, G. Reitmayr, and T. Drummond. Proforma: Probabilistic feature-based on-line rapid model acquisition. In *British Machine Vision Conference (BMVC)*, 2009.

[78] M. Pollefeys, L. Van Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops. Image-based 3d acquisition of archaeological heritage and applications. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*, pages 255–262, 2001.

[79] T. Robertazzi and S. Schwartz. An accelerated sequential algorithm for producing d-optimal designs. *SIAM Journal on Scientific and Statistical Computing*, 10(2):341–358, 1989.

[80] H. Roth and M. Vona. Moving volume KinectFusion. In *British Machine Vision Conference (BMVC)*, 2012.

[81] M. Rumpler, A. Irschara, and H. Bischof. Multi-view stereo: Redundancy benefits for 3D reconstruction. In *Proceedings of the 35th Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM)*, 2011.

[82] H. Samet. *Applications of spatial data structures: Computer graphics, image processing, and GIS*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[83] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *International Conference on Computer Vision (ICCV)*, 2011.

[84] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *British Machine Vision Conference (BMVC)*, 2012.

[85] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[86] K. Schmid, H. Hirschmüller, A. Dömel, I. L. Grixa, M. Suppa, and G. Hirzinger. View planning for multi-view stereo 3D reconstruction using an autonomous multicopter. *Journal of Intelligent and Robotic Systems*, 65(1-4):309–323, 2012.

[87] W. R. Scott, G. Roth, and J.-F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, 2003.

[88] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.

[89] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJVC)*, 80(2):189–210, 2008.

[90] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[91] V. L. T. Trzcinski, M. Christoudias and P. Fua. Boosting binary keypoint descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision (IJVC)*, 9(2):137–154, 1992.

[93] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, 2000.

[94] M. Trummer, C. Munkelt, and J. Denzler. Online next-best-view planning for accuracy optimization using an extended e-criterion. In *International Conference on Pattern Recognition (ICPR)*, pages 1642–1645, 2010.

[95] R. Y. Tsai. Radiometry. chapter A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, pages 221–244. Jones and Bartlett Publishers, Inc., USA, 1992.

[96] T. Tuytelaars and K. Mikolajczyk. *Local Invariant Feature Detectors: A Survey*. Now Publishers Inc., Hanover, MA, USA, 2008.

[97] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[98] G. Verhoeven. Taking computer vision aloft–archaeological three-dimensional reconstructions from aerial photographs with photoscan. *Archaeological Prospection*, 18(1):67–73, 2011.

[99] N. J. Wade and S. Finger. The eye as an optical instrument: from camera obscura to helmholtz's perspective. *PERCEPTION-LONDON-*, 30(10):1157–1178, 2001.

[100] A. Wendel, C. Hoppe, H. Bischof, and F. Leberl. Automatic fusion of partial reconstructions. In *Annals of the International Society for Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 2012.

[101] A. Wendel, A. Irschara, and H. Bischof. Automatic alignment of 3D reconstructions using a digital surface model. In *CVPR, Workshop on Aerial Video Processing*, 2011.

[102] A. Wendel, A. Irschara, and H. Bischof. Landmark-based monocular localization for mavs. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[103] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann. Active visual object reconstruction using D-, E-, and T-optimal next best views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[104] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, 2012.

[105] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`, 2007.

[106] S. Yu and M. Lhuillier. Incremental reconstruction of manifold surface from sparse visual mapping. In *Second International Conference on 3D Imaging, Modeling, Processing, Visualization (3DIMPVT)*, pages 293–300. IEEE, 2012.

[107] C. Zach. Fast and high quality fusion of depth maps. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, volume 1, 2008.

[108] C. Zach and M. Pollefeys. Practical methods for convex multi-view reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 354–367. 2010.

[109] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision (ICCV)*, pages 666–673, 1999.

[110] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. Augmented reality for construction site monitoring and documentation. *Proceedings of the IEEE*, 102(2):137–154, 2014.

[111] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. Augmented reality for construction site monitoring and documentation. *Special Issue: Application of Augmented Reality, Proceedings of the IEEE*, 102(2):137–154, 2014.

[112] S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr. FlyAR: Augmented reality supported micro aerial vehicle navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):560–568, 2014.

[113] S. Zollmann, D. Kalkofen, C. Hoppe, S. Kluckner, H. Bischof, and G. Reitmayr. Interactive 4D overview and detail visualization in augmented reality. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.