PhD Thesis

Scalable Scene Reconstruction and Image Based Localization

Arnold Irschara

Graz University of Technology
Institute for Computer Graphics and Vision

Thesis supervisors
Prof. Dr. Horst Bischof
Prof. Dr. Jan-Michael Frahm

Graz, February 2012

Contents

Es ist nicht genug zu wissen - man muss auch anwenden. Es ist nicht genug zu wollen - man muss auch tun.

Johann Wolfgang von Goethe

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.

Place

Date

Signature

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 entnommene Stellen als solche kenntlich gemacht habe.

Ort

Datum

Unterschrift

Acknowledgement

It would have been impossible for me to do this work without the help of several people. First of all, I would like to thank my supervisor Professor Horst Bischof for giving me the opportunity to perform research on the exciting field of computer vision and 3D reconstruction. Horst supported my research and gave me the full freedom to develop my own ideas, while arranging all the necessary conditions for doing good research. Special thank goes to Jan-Michael Frahm for being my second thesis supervisor. The research internship at his Department in 2008 has broaden my horizon and triggered new research directions. I am deeply indebted to Christopher Zach for his collaboration, constant encouragement, guidance and help during the first two years of my PhD studies. I also owe my deepest gratitude to all those people who contributed to this thesis, Manfred Klopschitz, Christoph Hoppe, Markus Rumpler, Andreas Wendel, Stefan Kluckner, Katrin Pirker, Thomas Pock, Philipp Meixner, Clemens Arth, Michael Donoser, Ernst Kruijff and Franz Leberl. I would like to thank all other colleagues of the ICG's vision group for providing a pleasant and inspiring atmosphere.

Most of all, I would like to thank my parents, for giving me the opportunity to do my studies and for their support in everything I did. Finally, I would also like to thank my sister, brother and friends who enriched my everyday life. Thank you.

Abstract

In this thesis two fundamental problems in computer vision are addressed: robust and scalable structure from motion and efficient localization from images. These two problems are highly interrelated tasks with several industrial applications, like mapping, navigation and augmented reality. The main contribution of this thesis is in building a complete, robust and scalable image based reconstruction and localization system that is suitable for large scale, real world problems. Within this work we describe the core components of a structure from motion pipeline that automatically builds a three-dimensional model of a scene given a set of unordered images. Improvements to current state-of-the-art reconstruction systems are made for several processing components including scalable image matching, geometric verification and robust structure from motion estimation. In particular we introduce an algorithm for non-monotone reasoning about view triplets which enables to identify mismatches caused by scene repetitions. The design of our system is based on algorithms that efficiently utilize modern graphics processing units to speed up several processing steps. The reconstruction system presented in this thesis is generic and allows photorealistic 3D modeling of cities from user contributed terrestrial data, dense modeling from large aerial images and fully automatic reconstruction of scenes from images taken by micro aerial vehicles (MAVs).

Furthermore, we present a new and fast location recognition technique based on structure from motion point clouds. Our proposed approach leverages the recent progress of 3D scene reconstruction and image retrieval techniques for efficient and robust view registration to a known 3D model of a scene. Vocabulary tree-based indexing of features directly returns relevant fragments of 3D models that allows full six degrees of freedom (6DoF) localization from images/videos in real time. Additionally, we propose a compressed 3D scene representation which improves recognition rates while simultaneously reducing computation time and memory consumption. Our localization framework is suitable for efficient registration of community photo collections to known landmark reconstructions and can be used for visual navigation and outdoor robot localization. A variation of our localization framework is also capable to run on modest hardware such as smart-phones and allows hand-held augmented reality.

The employed algorithms are extensively evaluated on a variety of datasets. Our experiments demonstrate robustness, scalability and high geometric accuracy of the proposed algorithms.

Kurzfassung

Diese Arbeit beschäftigt sich mit zwei zentralen Problemen in der Bildverarbeitung: zum einen mit der 3D Rekonstruktion von Szenen aus ungeordneten Bildern und zum anderen mit der effizienten Lokalisierung von Bildern zu einem bestehenden 3D Modell der Szene. Beide Probleme sind eng miteinander verknüpft und ermöglichen eine Reihe von industriell nutzbaren Anwendungen wie Kartographie, Navigation und Augmented Reality. Der zentrale Beitrag dieser Arbeit besteht in der Entwicklung eines skalierbaren, robusten und vollautomatischen System zur effizienten, bildgestützten 3D Rekonstruktion und Lokalisierung. Hierfür werden die Grundkomponenten eines "Structure from Motion" Systems beschrieben, das aus ungeordneten Bilddaten automatisch ein dreidimensionales Modell generiert. Zusätzlich werden verschiedene Lösungen vorgeschlagen die konventionelle Methoden erweitern und verbessern. Das Hauptaugenmerk liegt auf der Skalierbarkeit von Algorithmen zur effizienten Korrespondenzsuche zwischen Bildern, der geometrischen Verifikation und der robusten und vollautomatischen Orientierungsberechnung. Das System ist auf die effiziente Nutzung moderner Grafikkarten (GPUs) ausgelegt um rechenintensive Verarbeitungsschritte zu beschleunigen. Im Speziellen wird ein neuer Algorithmus vorgestellt der in der Lage ist über Bild-Tripel und fehlenden Korrespondenzen falsche geometrische Relationen zu detektieren. Das entwickelte Rekonstruktionssystem ist generisch und erlaubt die fotorealistische 3D Modellierung von Städten aus ungeordneten Bildern, dichte Oberflächenrekonstruktion aus großen Luftbildern und die 3D Rekonstruktion von urbanen Gebieten mittels Bildaufnahme von unbemannten Luftfahrtsystemen (MAVs).

Darüber hinaus wird ein neues und effizientes Bild-Lokalisierungssystem vorgestellt, welches mittels "Structure from Motion" Punktwolken arbeitet. Das vorgestellte System kombiniert Ansätze aus der 3D Rekonstruktion mit Algorithmen zur Bildsuche für die effiziente und robust Registrierung von Bildern zu einer bekannten 3D Szene. Ein hierarchischer Ansatz über ein Baumbasiertes Ähnlichkeitsverfahren ermöglicht die direkte Indizierung von 3D Punkt-Fragmenten und erlaubt die dreidimensionale Echtzeitlokalisierung von Bildern und Videos. Zusätzlich wird eine Methode vorgestellt welche die Szene komprimiert, dadurch wird die Lokalisierungsrate maximiert und gleichzeitig werden Speicher- und Rechenaufwand minimiert. Das Verfahren kann zur effizienten Registrierung von Internet Fotos zu bekannten 3D Stadtmodellen herangezogen werden und eignet sich für die visuelle Navigation und Lokalisierung von Robotern. Eine modifizierte

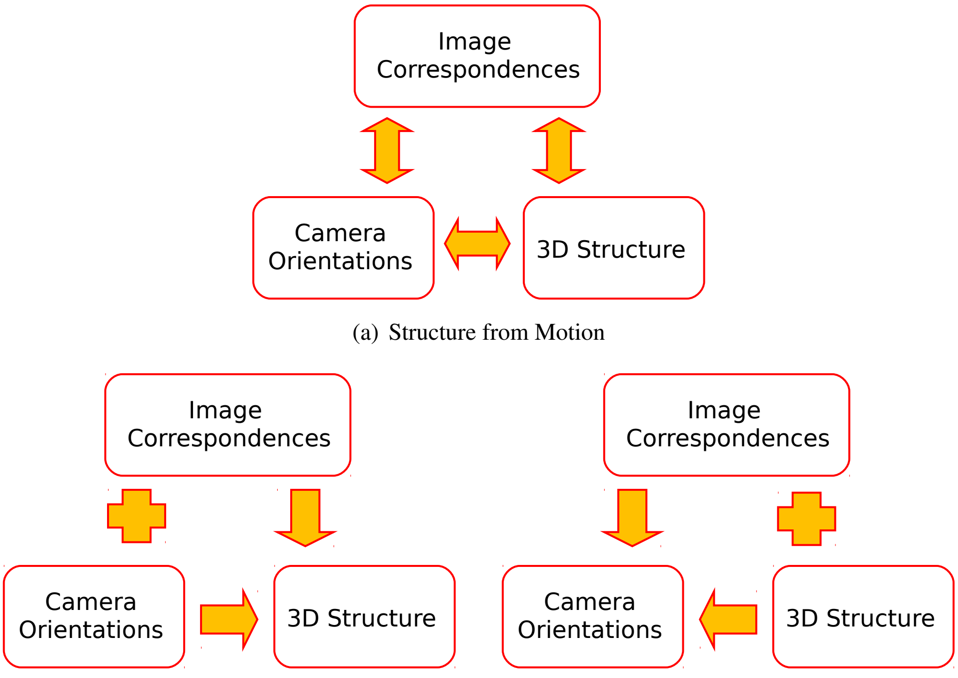
Variante erlaubt darüber hinaus die Ausführung auf Mobiltelefonen und ermöglicht Augmented Reality Anwendungen.

Die verwendeten Algorithmen sind generisch einsetzbar und wurden ausführlich auf verschiedenen Daten getestet. Unsere Experimente bestätigen die Robustheit, Skalierbarkeit und hohe geometrische Genauigkeit der vorgeschlagenen Algorithmen.

Chapter 1

Introduction

The visual system is the most important sensory system by which humans perceive the world. Vision enables us to navigate through previously known or unknown environments and allows us to interpret and build a representation of the surrounding [Marr, 1982]. Humans are well trained to sense the environment using stereo vision, motion and shape cues to get a three-dimensional impression and model of the environment with apparent ease. Such information is valuable for tasks such as localization, navigation and obstacle avoidance. Given that 3D is an integral part of our visual experience it is no surprise that recovering the 3D scene structure from images or video is one of the core problems in computer vision. While this process often involves simultaneously estimating both 3D scene geometry (structure) and camera pose (motion), this problem is commonly known as Structure from Motion (SfM). Based on tracked and matched features the relations between multiple views can be automatically computed [Hartley and Zisserman, 2000]. As shown in Figure 1.1 structure from motion consists of two interrelated tasks, namely triangulation and localization. On one hand, given the exact 3D position and orientation of the cameras, recovering the 3D structure of a scene can be achieved straight forward by triangulation of image correspondences. On the other hand, an existing 3D model of the scene allows the determination of the image pose directly by camera resectioning (localization) using 2D to 3D correspondences. In this thesis both problems are addressed and robust and efficient solutions are provided to solve those tasks. First, we present methods and applications for scalable and robust scene reconstruction from images. In particular we focus on offline batch based reconstruction methods for unorganized image collections that allow simple and flexible image acquisition at low cost. Second, we introduce a method for efficient and robust localization of new input images based on known 3D scene structure. Application domains range from environment mapping for e-commerce and real estate, games, film industry and simulation, to change detection, navigation, augmented and virtual reality. Figure 1.2 gives a summary of potential applications.



(a) Structure from Motion

(b) Triangulation

(c) Localization

Figure 1.1: (a) Structure from Motion problem and (b) Triangulation and (c) Localization subproblems.

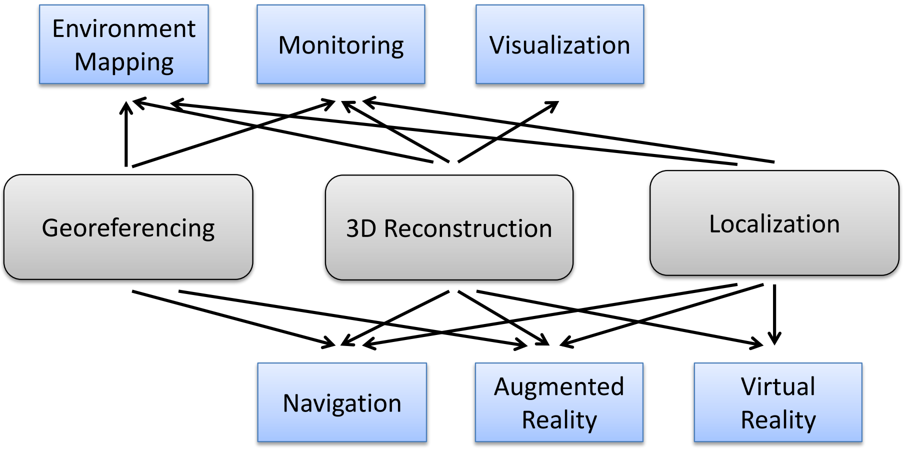
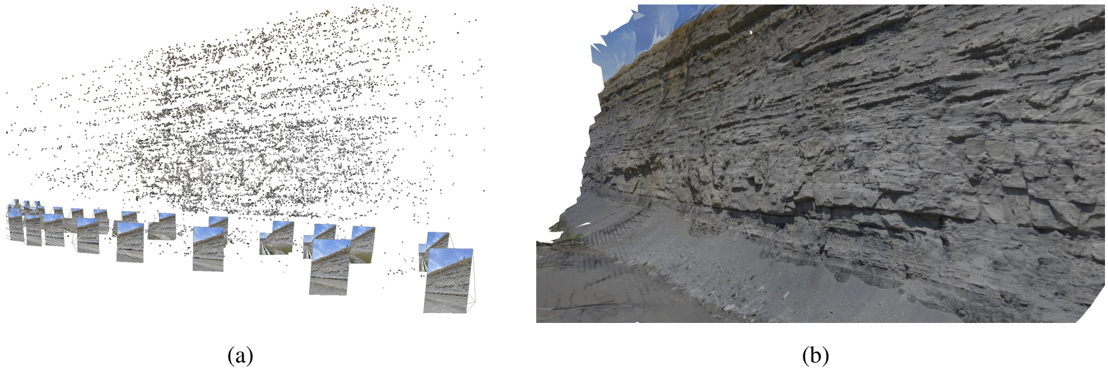


Figure 1.2: Potential application domains of image based 3D reconstruction and localization techniques.

1.1 Image based 3D Modeling

We observe an ever increasing demand for 3D models of single objects and the world. The main objective in 3D modeling are photorealism, cost efficiency, scalability and high geometric accuracy. Furthermore, there is a demand for simple and flexible acquisition procedures that can be rapidly deployed and are easy to perform by end-users. Today, several technologies are available to determine the 3D shape of an object. Those technologies can be classified into active and passive methods. While active sensing techniques such as Light detection and ranging (LiDAR) or time-of-flight cameras directly collect depth information about surfaces within its field of view, passive image based methods rely on the photogrammetric principle of triangulation. In general one can regard LiDAR as a more reliable technique for dense depth estimation since it works also for texture-less scenes and achieves favorable depth precision compared to small baseline stereo. However, the pixel density of cameras is typically much higher than the LiDAR measurements and novel automated photogrammetry technologies are capable to produce one or two magnitude denser point clouds than LiDAR [Leberl et al., 2010]. Additionally, LiDAR sensors require external tracking and instrumentation to determine the absolute position and orientation of the sensor. Such devices generally include a Global Positioning System (GPS) receiver and an Inertial Measurement Unit (IMU). Overall, 3D information computed by multi-view photogrammetry compares well with direct LiDAR-methods in terms of accuracy and photogrammetric image acquisition is cheaper and more flexible to apply. Furthermore, these techniques provide the advantage that texture (images) and geometry (depth) are produced from the same source of data, thus photorealism is naturally achieved and no additional calibration effort between camera and depth sensors is required. Figure 1.3 shows a 3D reconstruction of a highwall computed from multiple photographs. The combination of exact depth extraction using multiple view geometry combined with texture information of original images allows photorealistic 3D models of an environment.


(a)

(b)

Figure 1.3: Coal mine reconstruction form 72 images taken by a digital consumer cameras. (a) Orientation result by structure from motion and 1.3(b) oblique view of the respective dense 3D model represented as texturized mesh.

Multi-view reconstruction has matured during the past decades [Hartley and Zisserman, 2000] and led to fully automatic reconstruction systems from video and still images. Many reconstruction systems in computer vision are based on images from a moving video camera. These video based systems can use uncalibrated [Beardsley et al., 1996, Pollefeys et al., 2004] or calibrated cameras [Mouragnon et al., 2006, Nistér et al., 2004]. Applications include cultural heritage modeling, odometry, robost localization and city modeling. These methods are particularly appropriate to create large sparse reconstructions of continuous movements in real time. Current state-of-the art video-based 3D reconstruction [Nistér, 2001, Nistér et al., 2004, Pollefeys et al., 2004] allows detailed real time modeling of the environment into a dense textured polygonal mesh. The fusion of the structure from motion output with data from an Internal Navigation System (INS) and Global Positioning System (GPS) allows drift free 3D modeling [Pollefeys et al., 2004].

In the robotic literature, video based online structure from motion is also denoted as Simultaneous Localization and Mapping (SLAM) [Davison et al., 2007, Eade and Drummond, 2006]. [Newcombe and Davison, 2010] present a method which enables rapid and dense reconstruction of scenes from a single live camera in real time. The system relies on point-based real-time structure from motion provided by the Parallel Tracking and Mapping (PTAM) system of [Klein and Murray, 2007]. Recently, the same authors demonstrated a Dense Tracking and Mapping (DTAM) approach that performs camera pose tracking directly on the dense 3D model [Newcombe et al., 2011].

While sequential image acquisition by video enables frame-to-frame feature tracking [Shi and Tomasi, 1994], 3D reconstruction from unordered still images requires wide baseline image matching techniques [Pritchett and Zisserman, 1998]. Current advances in feature extraction [Lowe, 2004] and wide baseline matching [Mikolajczyk and Schmid, 2004, Mikolajczyk and Schmid, 2005] lead to fully automatic 3D reconstruction systems from unorganized images such as Internet photo collections [Snavely et al., 2008a]. Reconstruction systems based on still images (e.g. [Brown and Lowe, 2005, Kamberov et al., 2006, Martinec and Pajdla, 2007]) are in general designed to operate in batch mode. Photo Tourism [Snavely et al., 2008a] (and the related PhotoSynth[1] web-system) is probably the most-well known application for automatic structure from motion computation from a large set of unordered images. A collection of supplied images is analyzed and correspondences are established, from which a relevant subset of views and the respective 3D structure is determined. As the problem size becomes larger, i.e. hundred thousands of images, scalability becomes a key problem. Current state of the art uses efficient image search methods [Nistér and Stewenius, 2006] to battle down the initially quadratic matching complexity to sublinear search. Furthermore, massive parallel computing resources such as multi-processors [Agarwal et al., 2009] are employed to speed up feature extraction, matching and geometric estimation. In [Frahm et al., 2010] graphic processing units are used to handle even millions of images. Recent approaches in large scale structure from motion uses hierarchical methods [Strecha et al., 2010] or a combination of discrete and continuous optimization techniques [Crandall et al., 2011] to tackle the huge optimization problem. [Gherardi et al., 2010] presents a hierarchical reconstruction scheme based on balanced agglomerative clustering of unorganized images. In [Snavely et al., 2008b] the problem of efficient structure from motion for large, unordered, highly redundant, and irregularly sampled photo collections is addressed. The proposed approach computes a small skeletal subset of images that covers the whole scene and approximates the accuracy of the full image set. The authors show that this method can improve efficiency by up to an order of magnitude and more, while little or no loss in accuracy. An analo-

[1] *http://labs.live.com/photosynth*

gous idea is described in [Havlena et al., 2010], where a fast polynomial algorithm determines an approximated minimal connected dominating set in the view graph. The algorithm uses a prioritized approach which is able to avoid matching of highly redundant images with low change of reconstruction success.

One core challenge in structure from motion is large scale bundle adjustment [Triggs et al., 2000, Hartley and Zisserman, 2000]. Many variations of bundle adjustment exist, from algorithms that exploit the sparse structure of the problem [Lourakis and Argyros, 2009, Konolige, 2010] to large scale solutions [Agarwal et al., 2010] exploiting out-of-core processing [Ni et al., 2007] and parallelized multi core CPU/GPU implementations [Wu et al., 2011]. Wu et. al report that the induced runtime acceleration is up to thirty times fast then current state of the art methods, while maintaining comparable convergence behavior.

1.1.1 Image Acquisition Techniques

In image based reconstruction the scene can only be recovered up to a scale factor, unless the baseline of the camera motion or the dimension of at least one element in the scene is known. Hence metric reconstruction cannot be achieved directly. While this may seem to be a limitation, it implies that image based reconstruction methods have a wide range of applications, from the reconstruction of small objects from microscopy images to large satellite images capturing the whole Earth. Current work on image based modeling is concerned with the reconstruction of small objects [Hernández, 2004] and office workspace environments [Klein and Murray, 2007] to the reconstruction of buildings [Pollefeys et al., 2004] and whole cities from community photo collections [Agarwal et al., 2009, Frahm et al., 2010] and large area modeling from aerial [Zebedin, 2010] and satellite images [Kim and Muller, 2002]. Today an ever increasing amount of image data becomes available that is used for mapping, inspection and navigation. Worldwide people are taking a lot of photos[1], up from about 50 billion a year in 2007 to about 60 billion in 2011. Furthermore, more than 100 million photos are uploaded to the web every day[2] and this number shows no signs of slowing down. Table 1.1 gives a summary of the estimated number of images taken worldwide. The following sections describe different acquisition techniques and image sources that are available today for image based modeling.

Image source	# Images	avg. Resolution	Memory
Digital Photos taken in 2011	$>$ 60 billion	$\sim$ 1 Megapixel	$\sim$ 180 Petabyte
Number of Photos on Flicker (2010)	$>$ 5 billion	$\sim$ 2 Megapixel	$\sim$ 30 Petabyte
Photos per month Facebook (2012)	$>$ 3 billion	$\sim$ 0.5 Megapixel	$\sim$ 4.5 Petabyte
Google Street View Images (2012)	$\sim$ 1 billion	$\sim$ 5 Megapixel	$\sim$ 15 Petabyte
Microsoft / DigitalGlobe Clear30	$\sim$ 150 million	$\sim$ 109 Megapixel	$\sim$ 50 Petabyte

Table 1.1: Estimated number of images taken worldwide.

[1] http://www.itfacts.biz/50-bln-digital-photos-taken-in-2007-60-bln-by-2011/8985

[2] http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers

Terrestrial Mapping

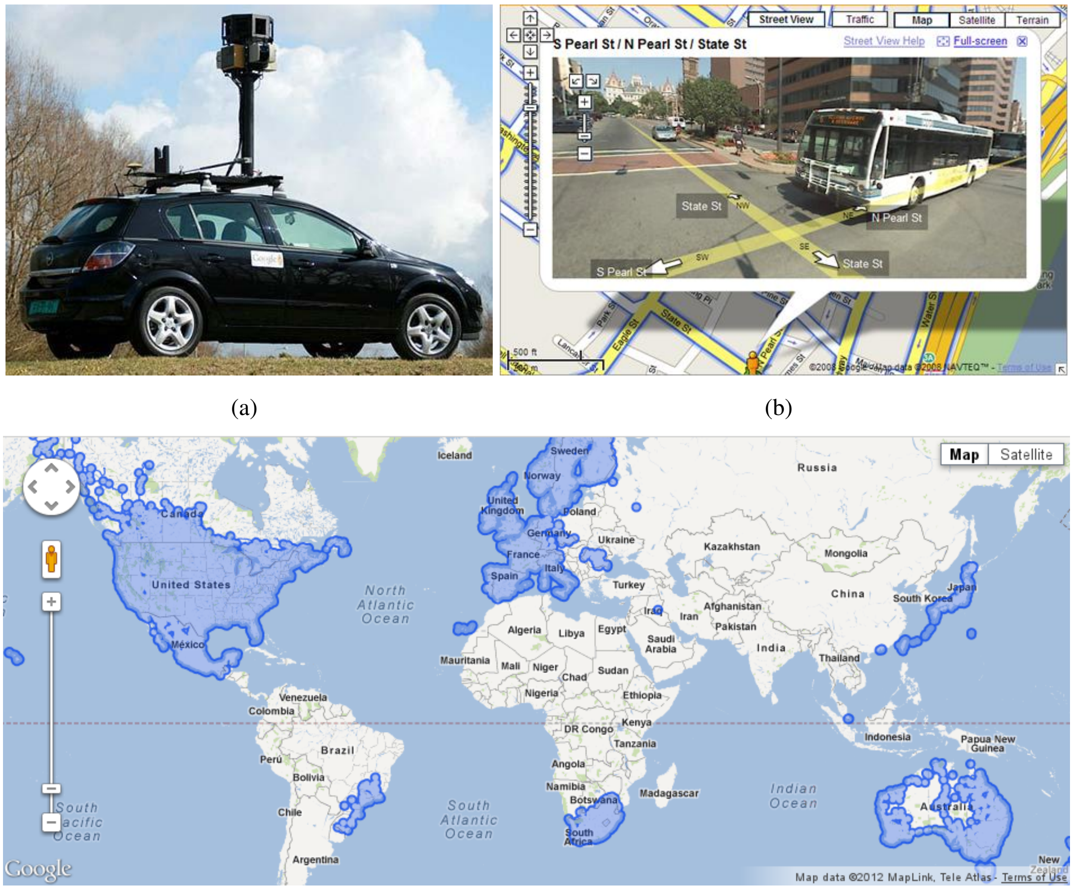
In the past ten years we have seen an explosion of consumer digital photography and a large growth of community photo-sharing websites (e.g. *flickr.com*). Today, there exist billions of photographs of sights and cities taken from a variety of cameras, viewing positions and angles. Mobile phones are equipped with high resolution cameras and provide a continuous growth of images available of the world. Even though, these photo collections are taken from a variety of different cameras and illumination conditions, fully automatic structure from motion modeling is feasible as shown by the authors of [Snavely et al., 2006, Agarwal et al., 2009, Frahm et al., 2010]. On the other hand, Google and Microsoft aim at large scale terrestrial mapping by systematic image acquisition using 360° camera systems mounted on cars that drive around and photograph each location (see Figure 1.4). So far, Google has collected tens of millions of street-side panoramic still images from a substantial part of the populated world, as shown in Figure 1.4(c). Given that the total length of all public roads in the United Sates is about 6.5 million kilometers, acquiring one shot every 5 meters results into more than one billion images. To figure out exactly where each image was taken, panoramic images are combined with signals from several sensors on the car, including global positioning device and monitors that measure speed and direction. A coarse 3D model of the scene is then used for navigation and to render smooth transitions between successive panoramas as shown in Figure 1.4.

Aerial Mapping

Novel digital aerial cameras produce high resolution images at no cost increase of overlaps that are readily suitable for photogrammetric end products like Digital Surface Models DSM, ortho-photo creation and 3D city modeling [Zebedin, 2010]. Fully automated image based creation of dense point clouds with an elevation measurement at each pixel makes the technology competitive with LiDAR-based surface measurements [Leberl et al., 2010]. It can be argued that the image-based approach offers many advantages over LiDAR, and that practically all aerial mapping scenarios will need digital images, even with LiDAR [Leberl and Gruber, 2003]. Current aerial flight missions are normally based on 80% forward and 60% sideward overlap. Large camera systems can achieve a Ground Sampling Distance (GSD) of less than 2cm at 12bit radial resolution (e.g. UltraCamXp, Microsoft Vexcel). Due to the the high image overlap, on average a point in the scene is visible in ten to twenty images which allows to employ fully automatic processing. The point cloud produced by multi-view dense matching can then be used for city modeling and true orthophoto generation, as Figure 1.5 shows. Currently the large scale data acquisition program Clear30 from DigitalGlobe[1] and Microsoft Vexcel[2] aims at collecting 30-cm aerial imagery of the entire contiguous United States and Wester Europe for the production of high resolution orthophoto mosaics. This translates into several Petabytes of high resolution aerial image data.

[1] http://www.digitalglobe.com

[2] http://www.vexcel.com



(a)

(b)

(c)

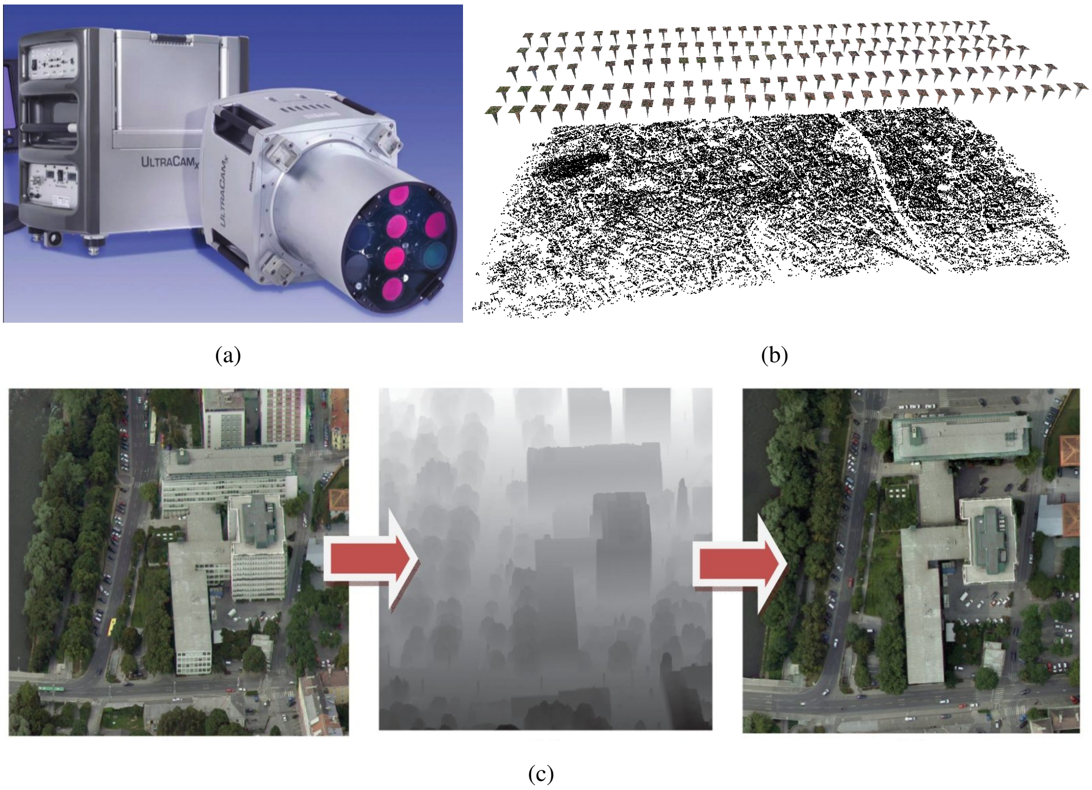
Figure 1.4: (a) Google Street View car with mounted 360° camera system for large scale terrestrial mapping and (b) Street Maps Sphere viewer. (c) Current world wide coverage of street view data (Januray 2012).

Micro Aerial Vehicles

Aerial photography has been the workhorse of remote sensing. Satellite imagery has augmented the remote sensing tool box since the launch of Landsat in 1972. Both aerial and satellite imaging result in very ordered and industrially planned image datasets. Recently, one can see a diversification of the image inputs for remote sensing [Eissenbeiss et al., 2009]. Photography from handheld amateur cameras, from balloons and Micro Aerial Vehicles (MAVs), all are subject to intensive research into their applicability to tasks previously reserved to industrial solutions.

Fully autonomous, light-weight MAVs have recently become commercially available at reasonable cost for civil applications. Equipped with consumer grade digital cameras, such systems allow rapid and cost efficient image acquisition from unconventional viewpoints around a scene. For instance the micro-drone md4-200[1] depicted in Figure 1.7 has the ability for vertical take off

[1] http://www.microdrones.com



(a)

(b)

(c)

Figure 1.5: (a) The eight-lens UltraCam aerial camera system with an in-flight image storage and processing unit produces panchromatic image tiles through four lenses linearly arranged along the flight direction. (b) Oriented image block of 155 UltraCam images with resolution 11500×7500 and ground sampling distance 8cm. (c) From left to right, aerial image with corresponding depth map and orthophoto generated by warping and stitching multiple images together using the reference digital surface model.

and landing, provides position hold and autonomous way-point navigation and is equipped with a standard digital consumer camera that can be tilted (up to 90°) to capture images from different angles.

MAVs are suitable to capture medium scale scenes and buildings which are in the range of some hundred meters at very high geometric resolution [Colomina et al., 2008, Schmid et al., 2012]. The obtained models provide an accuracy which is in the scale of centimeters, as shown in Figure 1.6. A MAV is often capable to provide visual information about an object which otherwise cannot be obtained. Mapping using images taken by MAVs has been addressed by many authors, e.g. in the context of digital surface model (DSM) extraction [Förstner and Steffen, 2007], archaeological preservation [Scaioni et al., 2009] and agricultural survey [Grenzdörffer et al., 2008]. The temporal (4-dimensional) analysis of local areas, as for instance the monitoring of building reconstruction sites (see Figure 1.7), becomes affordable because of the reduced cost of the hardware. Expensive helicopters or airplanes are replaced by ultra-light MAVs. The automated processing on the other hand reduces the labor cost substantially and makes such projects feasible.

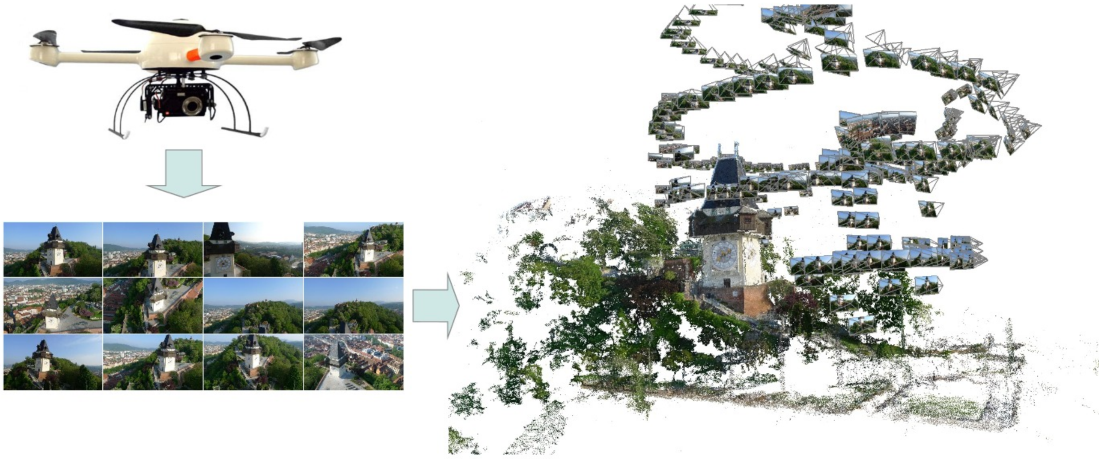


Figure 1.6: Micro-drone md4-200 with attached PENTAX Optio A40 and reconstruction of the clocktower of Graz from 420 images.

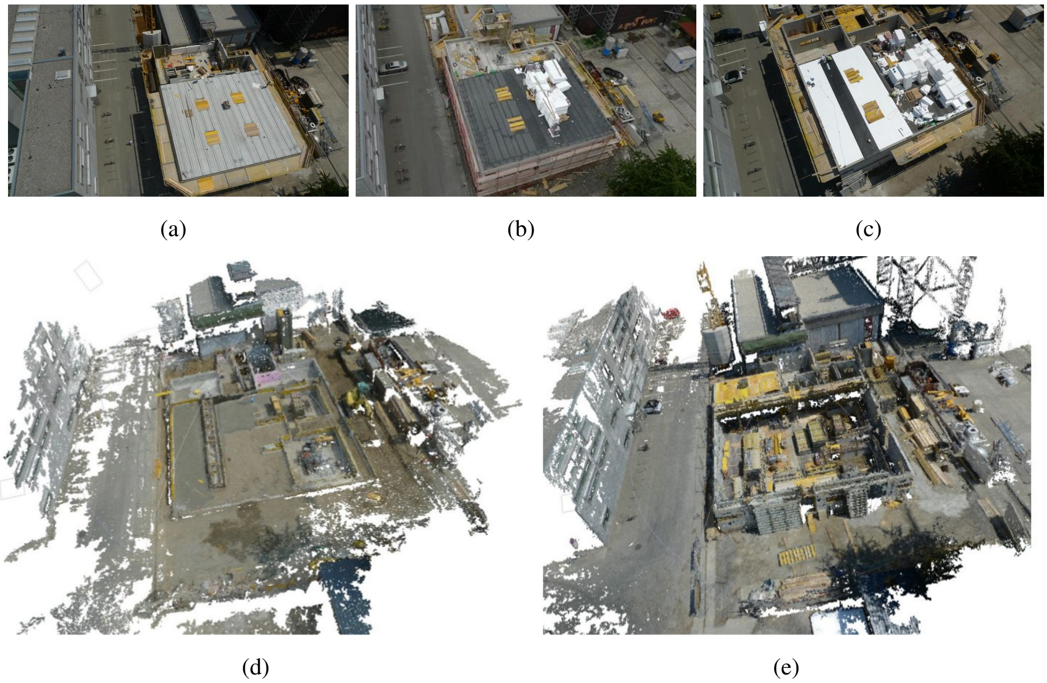


Figure 1.7: Monitoring changes of a reconstruction site using images taken from an autonomously flying MAV system. (a)-(c) Still images taken over a period of three days and (d),(e) 3D reconstruction results obtained by multi-view dense matching.

1.2 Image based Localization

Image based localization has many applications including navigation for robots [Se et al., 2002] and pedestrians [Robertson and Cipolla, 2004,Zhang and Kosecka, 2006], augmented reality [Arth et al., 2009, Gordon and Lowe, 2006] or 3D browsing and visualization of photo collections [Sattler et al., 2011, Li et al., 2010]. Unlike to Global Positioning Systems (GPS) that only provide positioning information, view registration of a camera with respect to a 3D scene delivers full six degree of freedom (6DOF) pose information and is also capable of working indoors and in urban canyons. Fast and accurate image alignment to a given scene is especially useful for Augmented Reality (AR) applications [Azuma et al., 1999] which aims for registration of 3D content to the live view of the world as captured from a camera. In order to get a realistic photo overlay without offset, pixel accurate pose estimation is required. In addition, if the 3D scene is registered to the World Geodetic System (WGS84), global geo-registered position and orientation estimation is possible. Such models can then be used for global localization tasks at the scale of the world. Furthermore, being a passive device, a camera requires low energy and instant, real time localization is possible through high frame rates. This is in contrast to GPS sensors that usually have a position update of about 1Hz. Image based localization is non-intrusive and conceptually appealing since accurate 3D pose and orientation information can be computed from image and video data, only.

A prerequisite of a fast and accurate image based localization system is the availability of an exact 3D model of the environment. The ideal case would be a precomputed visual map of the environment that encodes the exact illumination and viewing conditions from any desired viewpoint. A fast and flexible method to build such photo-realistic 3D models from an environment is image based 3D reconstruction as describe in the first part of the thesis (Chapter 2). Figure 1.8 depicts a 6DOF real-time localization result of a handheld video using our proposed image based localization technique as described in Chatper 5.

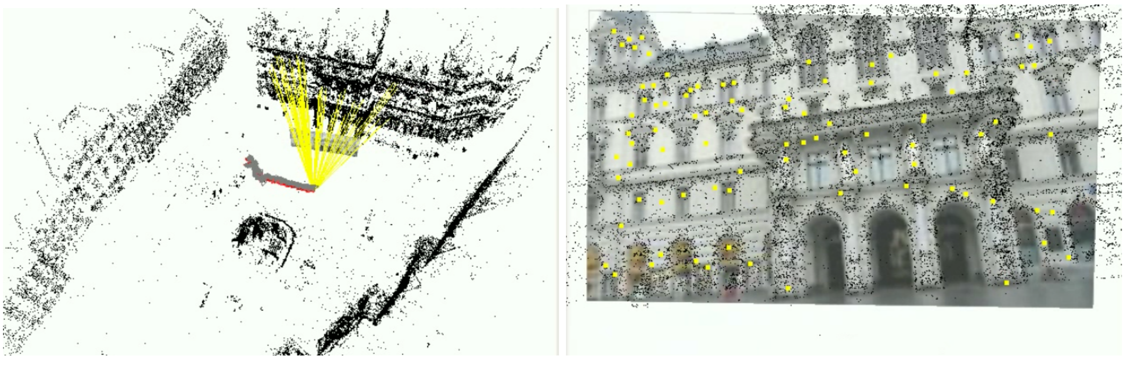

(a) (b)

Figure 1.8: Image based localization using a precomputed visual landmark of the environment. (a) Camera localization and (b) respective 2D to 3D aligned image content.

In the computer vision literature, the problem of location recognition has been addressed in the past by a variety of approaches [Robertson and Cipolla, 2004, Zhang and Kosecka, 2006, Zhu et al., 2008]. The most successful methods rely on wide baseline matching techniques based on sparse features such as scale invariant interest points and local image descriptors. The basic idea behind these methods is to compute the position of a query image with respect to a database of registered reference images [Schindler et al., 2007a], planar surfaces [Robertson and Cipolla, 2004] or 3D models [Najafi et al., 2006]. Assuming a static scene, geometric verification can be used to determine the actual pose of the camera with respect to the exemplar database. Different viewpoints or illumination changes are largely handled by robust features like SIFT [Lowe, 2004] and SURF [Bay et al., 2008] that act as descriptors of local image patches.

[Schindler et al., 2007a] present a city scale location recognition approach based on geotagged video streams and specific trained vocabulary trees using SIFT features. The vocabulary tree concept and inverted file scoring as described in [Nistér and Stewenius, 2006] allows sublinear search of large descriptor databases requiring low storage space. In contrast [Lepetit et al., 2005] recast matching as a classification problem using a decision tree and trade increased memory usage for expensive computation of descriptors at runtime. [Skrypnyk and Lowe, 2004a] present one of the first systems for 3D scene modeling, recognition and tracking with invariant image features. First, a sparse 3D model from the object of interest is reconstructed using multi-view vision methods. Second, SIFT descriptors associated with the sparse 3D points are organized into a kd-tree structure, and a best-bin first search strategy is employed to establish putative 2D-3D correspondences. A robust pose estimation algorithm is used for geometric verification and delivers the accurate pose of the query image with respect to the 3D model. Self-localization in indoor and smaller-scale environments using image or video data is also addressed by the visual SLAM (simultaneous localization and mapping) literature. [Eade and Drummond, 2008] propose a vocabulary tree-based approach for real-time loop closing, using a reduced SIFT-like descriptor.

Related work in the augmented reality context includes [Gordon and Lowe, 2006, Reitmayr and Drummond, 2006, Klein and Murray, 2007]. [Reitmayr and Drummond, 2007] proposed an accurate localization technique on modeling the GPS error with a Gaussian process for fast outdoor localization without user intervention. Recently, [Li et al., 2010] presented a location recognition approach based on prioritized feature matching exploiting stable scene features. Combining bag-of-features approaches with geometric verification to improve the precision of object recognition was proposed by [Xiao et al., 2008]. Visibility prediction of known 3D points with respect to a query camera was investigated in [Alcantarilla et al., 2011].

Based on the same fundamental concepts, in Chapter 5 we present a location recognition system with full 6DOF localization which runs in real-time for very large scenes. Our approach achieves competitive registration rates than current state-of-the-art view registration techniques [Sattler et al., 2011] and is more efficient in terms of processing time. Furthermore our proposed approach is fully scalable and prior pose information can be easily integrated. A precomputed set of synthetic views provides 3D point fragments that cover the space of admissible viewpoints. The 3D point fragments are globally indexed with a vocabulary tree data structure that is used for coarse matching. Real time performance for view registration on a desktop PC is achieved by utilizing modern graphics processing units for feature extraction and matching.

1.3 Contribution of the Thesis

This section summarizes the six key contribution of this thesis.

1) Scalable, efficient and flexible structure from motion pipeline This thesis focuses on making 3D reconstruction scalable and feasible for real world problems and applications. We extend current state-of-the-art by combining efficient image search with robust matching and 3D reconstruction methods. The use of highly parallel general purpose GPU (GPGPU) techniques based on the CUDA (Compute Unified Device Architecture) framework is a core component of all our design decisions. This includes a GPU accelerated vocabulary tree implementation, dense feature matching and geometric estimation. The achieved speedups on the GPU are about $10 - 20\times$ compared to single CPU processing. The development of a robust, scalable and fully automated structure from motion pipeline that processes unorganized real world images into 3D models is one of the main contribution of this thesis. In contrast to current state-of-the-art structure from motion systems that operate in batch mode, our pipeline enables incremental reconstruction. In our system, the 3D models can evolve over time and are all stored in a global repository. In addition, we present a calibration method based on coded markers that is very accurate and easy and fast to employ for end-users.

2) Globally optimal multi-view matching We present a new multi-view matching approach to generate accurate 2.5D dense depth maps from large aerial images. We use a global optimization algorithm based on a continuous energy minimization framework that delivers globally optimal solutions. Furthermore we demonstrate the benefit of using multi-view dense matching compared to standard stereo in terms of achievable geometric accuracy. From our synthetic experiments on a typical aerial camera network we conclude that true multi-view matching/triangulation outperforms two-view stereo approaches by about one order of magnitude.

3) View selection method We present an approach that leverages prior information from global positioning systems and inertial measurement units to speedup structure from motion computation. We propose a view selection strategy that advances vocabulary tree based coarse matching by also considering the geometric configuration between weakly oriented images. Furthermore, we introduce a fast and scalable reconstruction approach that relies on global rotation registration and robust bundle adjustment. The method is scalable and computationally more efficient than previous approaches.

4) Robust 3D reconstruction by Bayesian reasoning We introduce a novel algorithm that is able to detect incorrect two view geometries by reasoning about missing correspondences retrieved from view triplets. Our method allows to detect and disambiguate wrong epipolar geometries that often occur in scenes with duplicate scene structure. The algorithm can be used to augment existing 3D reconstruction systems with little computational effort.

5) Efficient localization framework We introduce a new algorithm and framework to register a single image and videos to large structure from motion reconstructions. The method performs real-time tracking by detection, hence it is not prone to drift and can automatically recover from tracking failures. We demonstrate the first large scale system that is capable to do full 6DOF global localization at $15fps$ on large structure from motion point clouds (e.g. 1.5 Million 3D points at a recognition rate of more than $> 90\%$). Since each frame is individually matched to the whole 3D database the method is very robust and automatically recovers from registration failures. The core component of our system is a fast indexing method based on 3D point fragments (structure from motion points) that allows fast view registration. We introduce the concept of synthetic views for the registration of images that are beyond the viewpoints of original images. A scene compression strategy further reduces matching costs and the amount of required memory.

6) Detailed evaluation of potential applications The principal algorithms and methods described in this thesis have practical benefits and provide efficient solutions for various problems. We demonstrate photorealistic 3D modeling of cities from user contributed terrestrial data, dense modeling from large aerial images and the reconstruction of urban areas using images obtained by micro aerial vehicles. Our localization framework allows efficient registration of community photo collections to known landmark reconstructions. Furthermore, it extends to 3D and is suitable for outdoor robot localization and delivers more robust pose estimates than current state of the art systems. We demonstrate that our algorithm can be used at modest platforms such as smart-phones that allows mobile augmented reality applications. Our reconstruction and localization solutions are generic, extensively evaluated on a variety of datasets and demonstrated scalability, accuracy and robustness.

1.4 Outline

The thesis is organized in two parts. The first part is concerned with fast end efficient creation of 3D models from our environment using image based 3D reconstruction methods. The second part deals with efficient localization of images with respect to the reconstructed 3D models.

Chapter 2 gives theoretical background for multi-view geometry and structure from motion. The individual processing blocks and algorithms of a structure from motion pipeline are described in detail. We present implementation methods based on graphic processing units to speed up several processing steps.

Chapter 3 discusses applications and evaluation for 3D scene reconstruction from multiple images. We present a Wiki-base reconstruction approach that is capable of reconstructing a scene from an unordered image collection. Furthermore, an algorithm for globally optimal multiview dense matching for aerial images is presented. We conduct reconstruction experiments using micro aerial vehicles and give a detailed comparison to a semi automatic structure from motion approach based on the PhotoModeler software. Moreover a reconstruction algorithm that effectively takes advantage of GPS/IMU information for matching and view selection is presented. The proposed algorithm considerably speeds up structure from motion computation.

Chapter 4 presents a robust method to tackle wrong epipolar geometries for 3D reconstruction. We propose an algorithm for non-monotone reasoning about view triplets which enables to identify epipolar geometries that are caused by scene repetitions. The method allows the detection and identification of wrong geometric relations and leads to more robust reconstruction results.

Chapter 5 introduces a novel image based localization approach based on structure from motion point clouds. Vocabulary tree based indexing of features directly returns relevant fragments of 3D models instead of documents from the image database. This makes the approach scalable and allows efficient registration of images with significantly different viewpoints than the original views used the reconstruction of the 3D model.

Chapter 6 describes three potential applications for image based view registration. We show the wide applicability of our approach, from the registration of community photo collections to known 3D landmarks, to robot navigation and localization and camera tracking for augmented reality applications.

Chapter 7 concludes the thesis with a summary of the outcomes and discusses open issues and ideas for future work.

Chapter 2

Structure from Motion

Reconstructing a scene from a set of 2D images is one of the core problems in computer vision. In particular, Structure from Motion (SfM) deals with the problem of estimating the 3D structure of a scene and camera orientations from 2D image measurements only. This problem has been extensively studied from a theoretical viewpoint in the past decades [Hartley and Zisserman, 2000,Faugeras and Luong, 2001]. Recently, computer vision methods for 3D scene reconstruction became robust enough to be used by non-vision experts. Today, there exists fully automated reconstruction systems that are able to reconstruct a scene from unordered images such as online photo collections downloaded from the Internet. In this chapter fundamental concepts and notations for multi-view modeling are introduced and individual processing components of our reconstruction system are described in detail. In particular, we study the problem of (i) fast and scalable image matching of unordered images, (ii) how to determine and deal with mismatches (i.e. incorrect correspondences) and (iii) how to efficiently determine a scene structure from correspondences. Several data-parallel parts of the pipeline are implemented for the execution on Graphic Processing Units using the Compute Unified Device Architecture (CUDA)[1] framework. The algorithms allow large scale reconstruction of thousands of still images.

2.1 Pinhole Camera Model

The perspective projection from a point in 3-space $\mathcal{R}^3$ onto an image plane $\mathcal{R}^2$ can be represented by the ideal model of a pinhole camera [Hartley and Zisserman, 2000]. The pinhole camera model is described by a center of projection $\mathbf{C}$ and an image plane Π of distance f, denoted as the focal length. Let $\mathbf{X} = [X, Y, Z]^\top$ be a point in space, the line joining $\mathbf{X}$ and the camera center $\mathbf{C}$ is projected to the point,

$$\mathbf{x} = \left(f\frac{X}{Z}, f\frac{Y}{Z} \right) \tag{2.1}$$

[1] http://developer.nvidia.com/category/zone/cuda-zone

onto the image plane Π. Introducing homogeneous coordinates, the projection equation can be written in matrix notation,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = P\mathbf{X} \tag{2.2}$$

where the point $\mathbf{x}$ has the 3-space coordinates $[u, v, 1]^\top$ with $u = x/z$ and $v = y/z$. As a result, a linear equation system is obtained. The assumption up to now was that the camera coordinate system coincides with the world coordinate system and is totally aligned with the image coordinate system. However, in general the image coordinate system is defined in pixel and thus the principal point $[0, 0, 1]^\top$ in camera coordinates is at location $[u_0, v_0, f]^\top$ in the image. The focal length f denotes the distance from the projection center to the image plane. Moreover, the metric pixel size may be different for x and y directions, as a result different scale factors $\alpha_x = fs_x$ and $\alpha_y = fs_y$ for both directions are obtained, where s_x and s_y are the number of pixels per unit distance for image columns and rows, respectively. For some very particular imaging situations, e.g. non-orthogonal pixel or images of images, a fifth parameter s_θ exists, referred to as skew parameter. However, for most cameras $s_\theta = 0$ is satisfied. This relations are expressed by the camera calibration matrix K, which describes the transformation between the image coordinate system and the camera coordinate system,

$$K = \begin{bmatrix} \alpha_x & s_\theta & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.3}$$

The calibration matrix K describes the imaging system, together with a rigid body transformation between the world coordinate system and the camera position, a general 3×4 projection matrix P is described by,

$$P = \begin{bmatrix} \alpha_x & s_\theta & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = K[R \mid \mathbf{t}] \tag{2.4}$$

where R is a 3×3 rotation matrix and $\mathbf{t} = [x, y, z]^\top$ a translation vector. In total therefore P depends on 11 parameters, five for the intrinsic relations and another six for the external parameters, denoted as the exterior orientation. The projection matrix can be expressed by the camera center $\mathbf{C}$,

$$P = KR[I \mid -\mathbf{C}] \tag{2.5}$$

with $\mathbf{C} = -R^\top \mathbf{t}$ and $\mathbf{I}$ is the 3×3 identity matrix. The pinhole camera model and the different coordinate systems involved are shown in Figure 2.1.

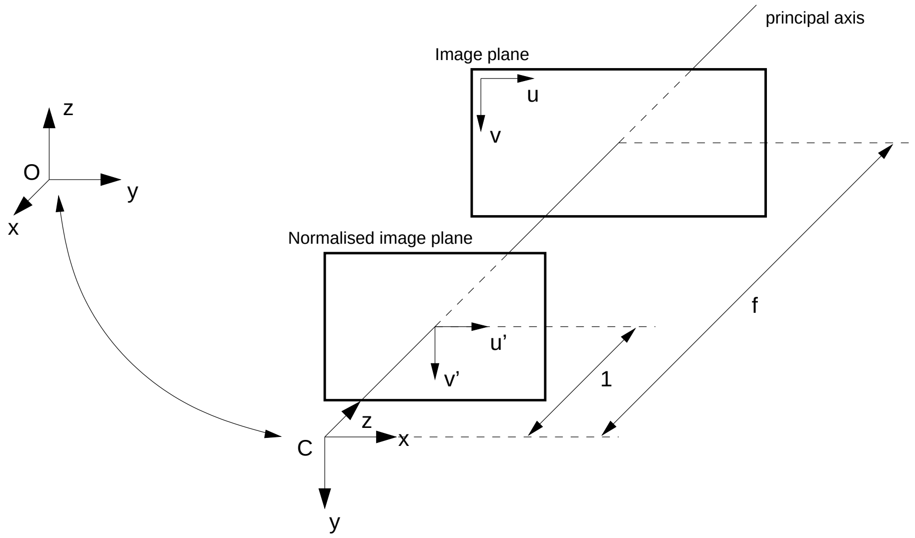


Figure 2.1: Camera geometry. Image coordinate to camera coordinate transformation together with an euclidean rigid body transformation between the world and camera coordinate frame.

2.1.1 Radial Distortion

The pinhole camera model assumes a linear model of the imaging process, thus world point, image point and optical center are collinear. For real (non-pinhole) lenses this assumption is normally not valid since non-linear deviations exists. There can be found two types of distortions, radial distortion and tangential distortion. However, only radial distortion has a significant influence on the image geometry and can be seen as a deficiency in straight lines transmission [Devernay and Faugeras, 2001]. Tangential distortion is usually insignificant and is not considered in the camera model. The effect of radial distortion is that straight lines are bended as general curves and points are moved in the radial direction from their correct position. Especially when working with non-metric digital cameras, the radial distortion reaches significant values and a correction of this distortion should be the first step in image processing. Normally, the radial lens distortion is modeled as,

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \tag{2.6}$$

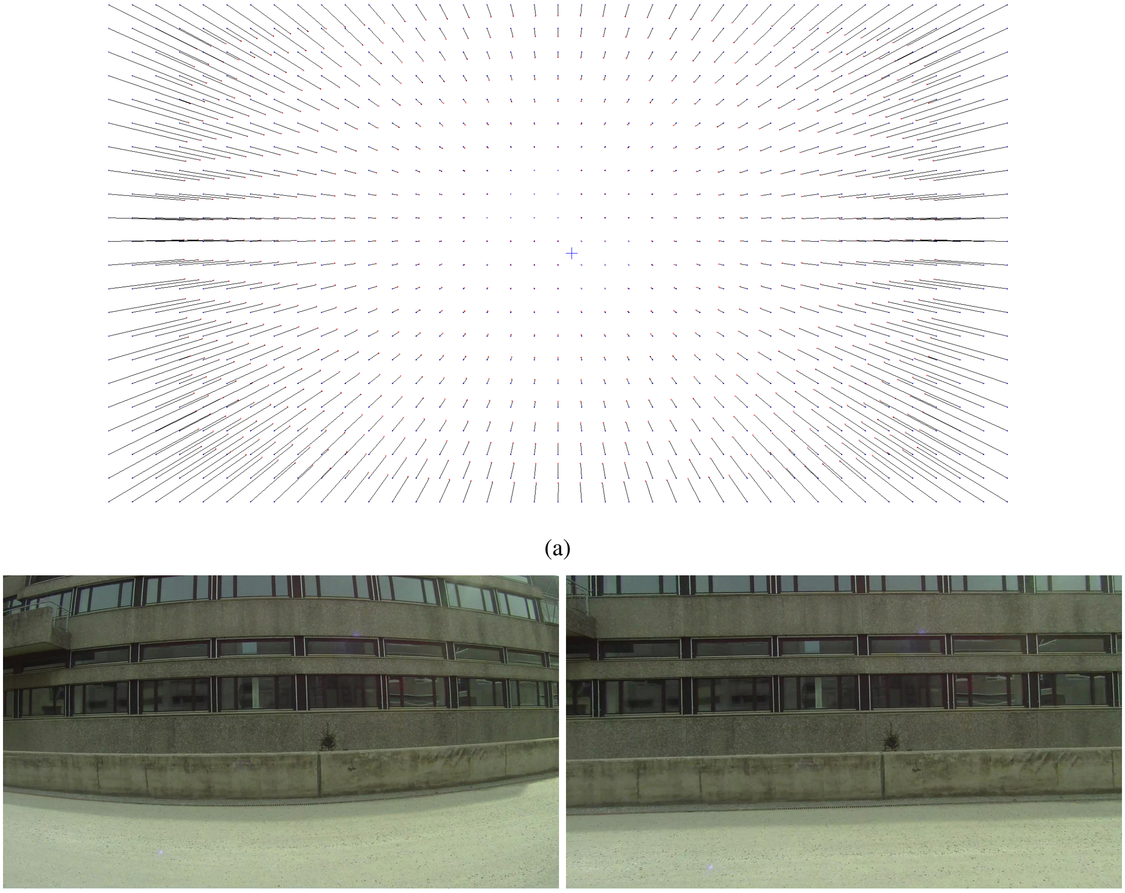
where (x, y) are the the measured coordinates, $(\hat{x}, \hat{y})$ are the corrected coordinates and (x_c, y_c) is the center of radial distortion, with $r^2 = (x - x_c)^2 + (y - y_c)^2$.

The function $L(r)$ is usually approximated by a Taylor series expansion,

$$L(r) = 1 + k_1\, r + k_2\, r^2 + k_3\, r^3 + \ldots \tag{2.7}$$

where the coefficients for radial correction $\{k_1, k_2, k_3, \ldots, x_c, y_c\}$ are considered part of the interior calibration of the camera. This correction together with the camera calibration parameters specifies the mapping from an image point to a ray in the camera coordinate system. Figure 2.2

depicts the distortion map of a wide angle lens. Note, the distortion at the fringe of the image is considerable and the projection substantially deviates from the ideal linear pinhole camera model.



(a)

(b)

(c)

Figure 2.2: (a) Radial distortion map of a wide angle lense. The non-linear distortion is significant, the deviation from a linear-pinhole camera model is about 20% at the fringe of the image. (b) Raw image and (c) undistorted result after applying the inverse distortion function and bilinear interpolation.

2.2 Epipolar Geometry

The geometric relation between two images, taken from different viewpoints is based on the well established epipolar geometry [Faugeras and Luong, 2001, Ma et al., 2003, Hartley and Zisserman, 2000]. A 3D point $\mathbf{X}$ captured from two different camera positions is projected to image location $\mathbf{x}$ in the first and to $\mathbf{x}'$ in the second image. The intrinsic relation of the point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ is known as the epipolar constraint, a point visible in one image is restricted to lie on a line in the second image. The epipolar geometry can be expressed by the rotation and translation of the first camera P to the second camera P', with known intrinsic K,

$$\mathbf{x}'^{\top}(K'^{-1})^{\top}S(\mathbf{t})R(K)^{-1}\mathbf{x} = 0 \tag{2.8}$$

where R is a 3×3 rotation matrix and $S(\mathbf{t})$ is a translation matrix of the form,

$$S(\mathbf{t}) = [\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} . \tag{2.9}$$

The epipolar constraint (2.8) is described by the essential matrix E and encodes the relative pose between the two cameras,

$$E = S(\mathbf{t})R \, . \tag{2.10}$$

Analogous for the uncalibrated case (i.e. unknown camera intrinsics), the epipolar constraint is based on the fundamental matrix,

$$\mathbf{x}'^{\top}F\mathbf{x} = 0 \tag{2.11}$$

and thus E can be rewritten as,

$$E = K'^{\top}FK \, . \tag{2.12}$$

As illustrated in Figure 2.3, the camera centers $\mathbf{C}$ and $\mathbf{C}'$, the 3-space point $\mathbf{X}$, and its images $\mathbf{x}$ and $\mathbf{x}'$ lie in a common plane π also denoted as epipolar plane. The camera centers are connected by the baseline, which virtually intersects the image planes at the epipols $\mathbf{e}$ and $\mathbf{e}'$. The intersection of an epipolar plane with the image plane is called epipolar line. Each point $\mathbf{x}$ in the left image corresponds to the epipolar line $\mathbf{l}' = F\mathbf{x}$ in the right image and vice versa. The epipolar line in one image is a projection of the straight-ray from the 3-space point $\mathbf{X}$ to the other camera center $\mathbf{C}$ and $\mathbf{C}'$, respectively.

The fundamental matrix F and essential matrix E can be determined solely from sets of matching features that satisfy the epipolar constraint [Hartley and Zisserman, 2000]. For calibrated cameras (i.e. known intrinsics K), the essential matrix can be determined from a minimal set of five point correspondences using the five-point algorithm [Nistér, 2004]. If the intrinsics are unknown, the eight-point algorithm [Hartley, 1997] offers the computationally most simple and efficient solution to determine the fundamental matrix. However, since F is a rank 2 homogeneous matrix with 7 degrees of freedom, seven correspondences are already sufficient [Hartley and Zisserman, 2000] to compute F.

2.2.1 Five-Point Algorithm

The essential matrix encodes the epipolar constraint for calibrated point correspondences $\hat{\mathbf{x}} = K\mathbf{x}$ and $\hat{\mathbf{x}}' = K'\mathbf{x}'$ and is composed by a rotation matrix R and a translation vector $\mathbf{t}$, both with three degrees of freedom,

$$E = [\mathbf{t}]_{\times}R = R[R^{\top}\mathbf{t}]_{\times} \, . \tag{2.13}$$

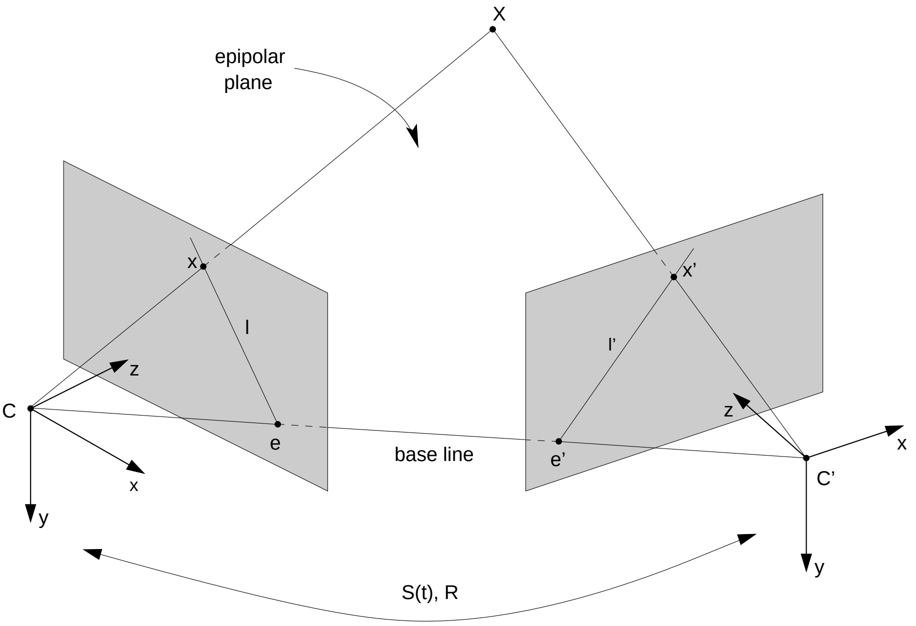


Figure 2.3: Epipolar geometry. The camera baseline intersects each image plane at the epipols $\mathbf{e}$ and $\mathbf{e}'$. A point $\mathbf{x}$ in the first image lies on an epipolar line $\mathbf{l}'$ in the second image, which is the image of the ray from point $\mathbf{X}$ to the first camera center $\mathbf{C}$.

In [Huang and Faugeras, 1989] the prove is given that beside the singularity property,

$$\det E = 0 \tag{2.14}$$

an additional algorithmic constraint,

$$\text{trace}^2(EE^\top) = 2\,\text{trace}((EE^\top)^2) \tag{2.15}$$

is satisfied, which means that two non-null singular values of E are equal. This implies two independent algebraic constraints [Faugeras and Luong, 2001]. Hence together with the up-to-scale definition and the singularity constraint, the essential matrix has in total only five degrees of freedom. The five-point problem was firstly investigated in [Kruppa, 1913], showing that up to eleven solutions exist. However, requiring that the scene points are in front of the cameras (twisted pair ambiguity), the solutions can be constricted to ten [Nistér, 2004]. Analogous to the eight-point algorithm [Hartley, 1997] a linear equation system is solved,

$$(u'u, u'v, u', v'u, v'v, v', u, v, 1)\mathbf{e} = \mathbf{0} \tag{2.16}$$

for a minimal set of five point correspondences $[u, v, 1] \leftrightarrow [u', v', 1]$. The four vectors $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$, which span the right nullspace of E, may be computed by singular value decomposition or QR-factorisation. Finally the essential matrix E can be expressed by a linear combination of the four vectors, now in matrix notation X, Y, Z, W,

$$E = xX + yY + zZ + wW \tag{2.17}$$

where x, y, z, w are scalar factors. Since E is defined up to a common scale factor, $w = 1$ is assumed. By inserting E in the trace-constraint (2.15) together with the rank constraint (2.14) gives ten third order polynomials. The solutions may be obtained algebraically performing Gauss-Jordan elimination with partial pivoting. An efficient solution to compute E is given in [Nistér, 2004] and [Li and Hartley, 2006].

2.3 Processing Pipeline

In this chapter we focus on computing structure from motion from calibrated cameras that correspond to an ideal pinhole camera model. Hence we assume that the internal parameters of the cameras are known and images are unwarped according to the radial lens distortion (see Section 2.1.1). Unlike to auto-calibration approaches [Pollefeys et al., 2004] that do a projective reconstruction first and than upgrade to Euclidean by solving for the internal camera parameters, calibrated cameras allow direct metric reconstruction. For wide angle lenses, the radial distortion on the fringe of the image can be significant and largely affect the linear pinhole camera model as shown in Figure 2.2. The radial distortion implies that image projections of straight lines in 3D are not straight any more, thus thresholds on linearity may well be erroneously exceeded in the original images. Furthermore, structure from motion using calibrated cameras can be considered as more reliable and robust since the likelihood of degenerate scene configurations (e.g. planar vs. non-planar scenes) is lower [Hartley and Zisserman, 2000]. Such configurations often occur in man-made environments and projective reconstruction fails when features common to three consecutive views are all located on a plane [Pollefeys et al., 2002]. Furthermore, for a calibrated approach, model selection [Torr et al., 1998, Frahm and Pollefeys, 2006] is not necessary in order to distinguish between scenes with and without dominant planar structure. Overall, also an increased processing speed is achieved due to the lower dimensionality of the problem (i.e. 7 DOF for the fundamental matrix computation vs. 5 DOF for the essential matrix). This is especially true for robust estimation algorithms such as RANSAC [Fischler and Bolles, 1981], since a minimal parametrization can handle more outliers with less RANSAC-iterations [Hartley and Zisserman, 2000].

In general a structure from motion pipeline for unordered sets of images consists of the five processing steps: feature extraction, coarse matching, detailed matching, geometric verification and geometric estimation.

1. **Feature Extraction:** The first processing step is in extracting distinctive feature points that act as local image descriptors. These features are necessary to establish corresponding locations in different images and are used to compute the camera pose.

2. **Coarse Matching:** Based on features, an epipolar graph is computed from pairwise image matching. Matching all potential image pairs is computationally very expensive, hence a multi-stage matching approach is normally employed. First, a coarse image similarity is computed using fast image retrieval techniques to efficiently determine image pairs that

are expected to share common scene elements. Additionally, prior information such as knowledge of sequential image acquisition or external pose information from GPS/INS can be used to limit potential matching pairs.

3. **Detailed Matching:** The next component deals with establishing correspondences between pairs of images. For video sequences local search techniques such as correlation or least squares can be used to track features between individual frames. When a large amount of motion or appearance change between images is expected (e.g. unordered still images), features are detected independently in all images and matched based on their local appearance. This operation is normally very time consuming, hence parallelization methods are often used to speed up processing.

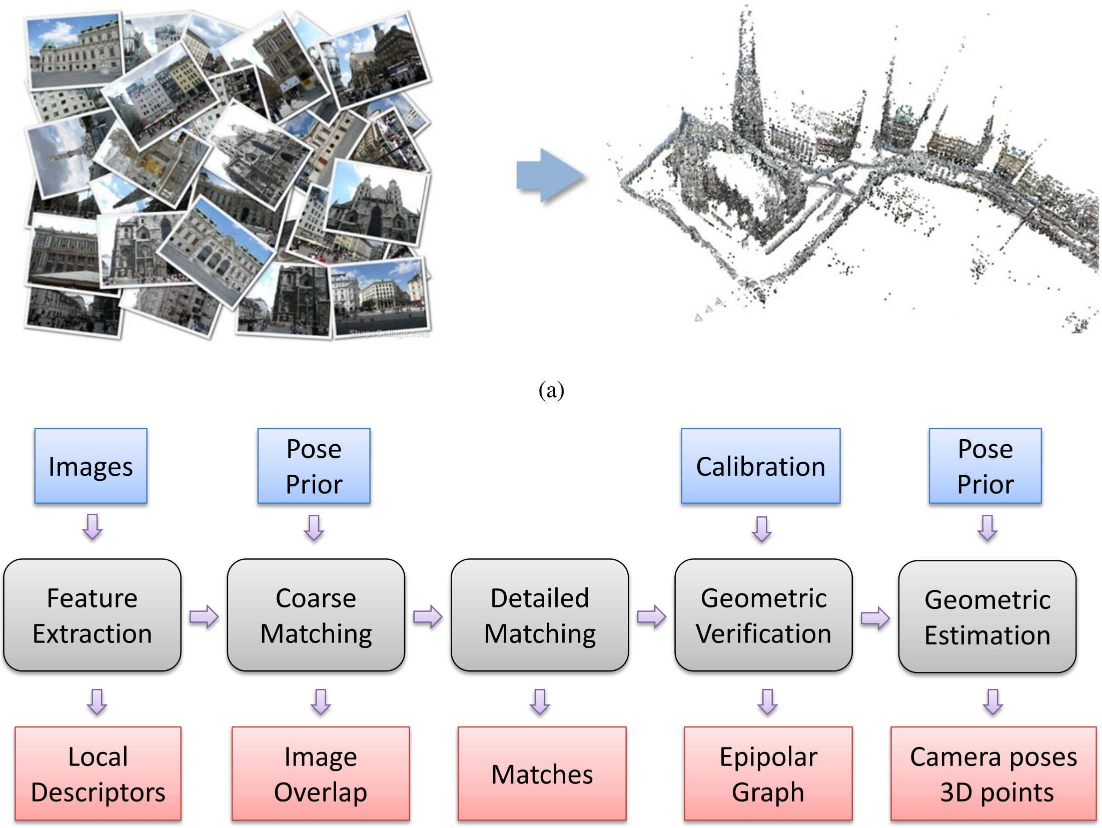
4. **Geometric Verifcation:** Feature matching delivers a set of potential correspondences that are used to compute the relative orientation between cameras. As no exact matching producer exist, geometric verification on the epipolar geometry effectively reduces outliers that arise from mismatches.

5. **Geometric Estimation:** Finally, pairwise correspondences are linked into point-tracks and structure from motion is solved by the geometric estimation module.

In Figure 2.4 an overview of a general structure from motion pipeline is depicted. The following sections give algorithmic details of our reconstruction system that combines and extends current state-of-the-art approaches and is designed to leverage the massive parallel processing power of current Graphics Processing Units.

2.4 Feature Extraction

One of the very first processing steps in a structure from motion pipeline is the determination of correspondences between (all) or a subset of images. This involves the detection of stable and invariant image locations (interest point extraction) and the local description of the appearance that surrounds the points. Over the past decades a variety of feature detectors and descriptors have been proposed. Early work on interest point detectors is based on corner detectors [Moravec, 1980, Förstner and Gülch, 1987, Harris and Stephens, 1988] and respective image patches that describe the local surrounding of the keypoints, denoted as descriptors. These detectors are based on the local autocorrelation matrix [Shi and Tomasi, 1994] around each pixel and show strong invariance to rotation and illumination changes. Scale invariance can be achieved by performing the same operations at multiple resolutions in a pyramid. However, it turns out that it is more efficient to extract features that are stable in both location and scale [Mikolajczyk and Schmid, 2004]. Extensive evaluation on interest point detectors and local descriptors [Mikolajczyk and Schmid, 2005] has shown that the Scale Invariant Feature Transform (SIFT) [Lowe, 2004] and the Speeded Up Robust Features (SURF) operator [Bay et al., 2008] are among the top performing features in terms of accuracy and repeatability. These features are invariant to scale and rotation



(a)

(b)

Figure 2.4: (a) Unordered set of input images and corresponding structure from motion result of the Graben street (Vienna) using correspondence information only. (b) A flowchart of individual processing steps of an automated structure from motion pipeline.

and robust against illumination changes and geometric distortion, hence they are well suited for wide baseline matching. Furthermore, these interest points can be efficiently determined and fast GPU implementations [Wu, 2007] are available. Moreover, these detectors are sub-pixel accurate and feature extraction normally delivers enough repeatable and matchable keypoints for structure from motion computation. Figure 2.5 depicts matching results between challenging image pairs such as wide baseline, drastic illumination and very large scale changes.

While SIFT and SURF are very repeatable and stable features, the computational complexity is often too high for real time applications. This is especially true for mobile devices with limited computational resources. The ever growing resolution of images and the increasing number of vision applications requires computationally efficient algorithms to handle the large amount of data. This is especially true for new applications like hand-held augmented reality, targeted to run on mobile devices with limited computational resources. Examples are the FAST [Rosten and Drummond, 2006] and AGAST [Mair et al., 2010] detectors which are many times faster than other existing corner detectors while providing high levels of repeatability under large aspect changes at the same time. FAST interest points combined with the recently proposed and very efficient and robust binary descriptor BRIEF [Calonder et al., 2010] is an efficient toolbox for key-point detection and matching. Furthermore, BRIEF is highly discriminative even when using relatively few bits and can be computed using simple intensity difference tests. Furthermore, the descriptor similarity can be evaluated using the Hamming distance, which is very fast to compute using the latest SSE4 instruction set. Recent state-of-the-art includes BRISK [Leutenegger et al., 2011] and ORB [Rublee et al., 2011], these feature achieve matching performance comparable to SIFT and SURF but are about two orders of magnitude faster to compute.

In our pipeline, any one of the previously mentioned scale invariant feature detectors and descriptors can be selected to establish correspondences. We primary focus on SiftGPU [Wu, 2007], because this approach currently offers the best trade-off between matching performance and computation time on GPU supported desktop PCs.

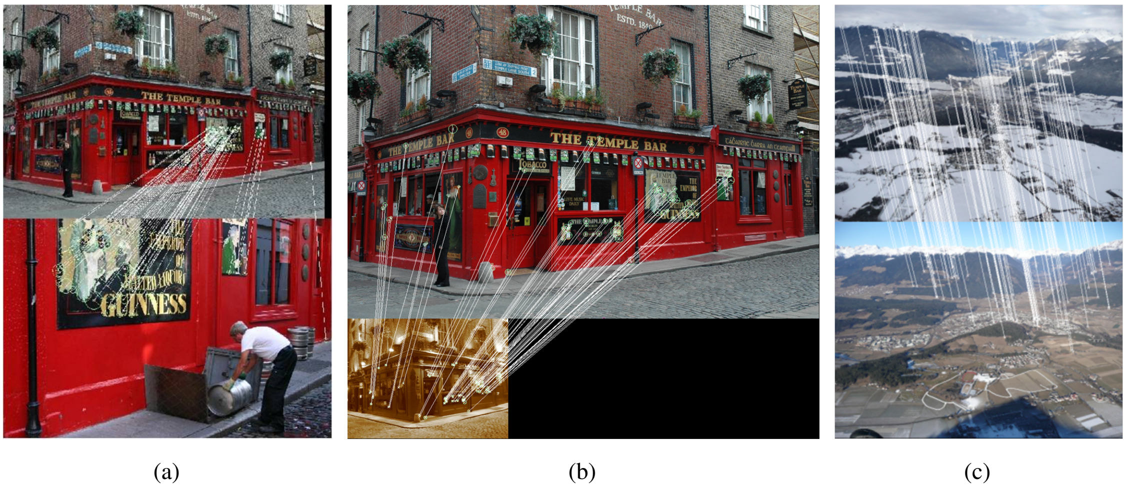

(a) (b) (c)

Figure 2.5: Natural feature matching results based on the Scale Invariant Feature Transform (SIFT) for challenging image pairs, (a) large scale and viewpoint changes, (b) illumination changes (e.g. day, night) and (c) large scene variations (e.g. summer-winter scene). Lines indicate corresponding points that satisfy the epipolar geometry. Even though the distortion of the images is significant a large number of correct correspondences is obtained. This would not be possible using conventional feature detectors like Harris corners and local intensity patches.

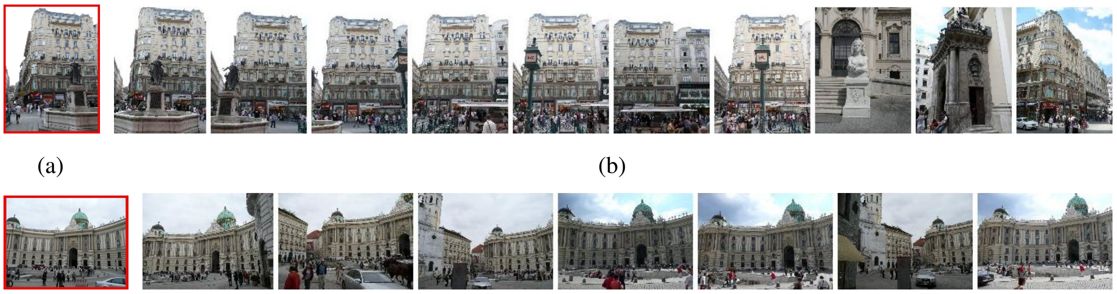
2.5 Coarse Matching

Unlike feature point tracking in video sequences, where correspondence search can be restricted to local regions, matching of unordered still images essentially requires exhaustive search between all image pairs and all features seen therein. Hence, the matching costs are quadratic in the total number of extracted features from the image database. Note the number of SIFT features from a medium sized image (e.g. 4000×3000 pixel) normally exceeds a value of 10000. For a small image database consisting of 1000 images, more than 10 million SIFT keys are detected. This translates into 100 billion descriptor comparisons that are necessary for exhaustive nearest neighbor search. This is a considerable amount of computation, which turns out to be a prohibitively expensive operation executed on a single CPU. Dependent on the scene structure, matching all $\binom{n}{2} \approx O(n^2)$ pairs of images is actually not necessary since many pairs do not overlap. To make the correspondence search more tractable, we divide the matching procedure into two submodules. We build upon work on efficient image retrieval [Nistér and Stewenius, 2006] and use a vocabulary tree to determine an image-to-image similarity score. Recent work on efficient image similarity computation includes [Chum et al., 2009,Chum et al., 2011]. Such an approach assumes that each image is represented as a bag of words and the occurrence of similar words between images determines a similarity score. This concept is borrowed from state-of-the-art text retrieval and document search [Brin and Page, 1998]. Consequently, only images with a sufficiently high similarity score are considered for detailed pair-wise image matching. Our proposed approach described in [Irschara et al., 2007] is nowadays a standard component in large scale reconstruction pipelines [Agarwal et al., 2009]. This strategy allows to substantially reduce the matching effort since for large image databases an image usually only matches with a small fraction the database due to missing overlap and occlusions.

2.5.1 Vocabulary Tree based Image Similarity

The bag-of-words image representation [Sivic and Zisserman, 2003] based on SIFT features [Lowe, 2004] are at the core of state-of-the-art large scale image retrieval systems. This representation describes an image based on a histogram of quantized feature occurrences with respect to a codebook of pre-defined visual words. The corpus is hence organized as an inverted file structure that compactly represents the whole image database. An efficient approach for approximated nearest neighbor search on the codebook can be done using hierarchical quantization of descriptor vectors, also denoted as a vocabulary tree [Nistér and Stewenius, 2006]. The tree is determined up to some maximum number of levels L and each node is divided into K children. Each SIFT-key is then propagated down the tree by comparing the descriptor vector to the K children and choosing the closest cluster center.

The vocabulary tree concept relies on the following basic assumption: if the similarity between two features $sim(f_i, f_j)$ is high, then there is a relatively high probability that the two features are assigned to the same visual word $w(f_i) \equiv w(f_j)$, i.e. the features reach the same leaf node in the vocabulary tree. Based on the quantized features from a query image $\mathcal{Q}$ and each database image $\mathcal{D}$ a scoring of relevance is derived. Typical scoring functions are based on a vector model, as for instance the *tf-idf* (term frequency, inverse document frequency), which delivers a relative document ranking according to the degree of similarity between query and database images. Figure 2.6 shows query images and corresponding top-ranked database images according to a vocabulary tree based image retrieval system.



(a) (b)

(c) (d)

Figure 2.6: (a),(c) Sample query images from the Vienna dataset (consisting of 2640 street side images) and (b),(d) corresponding top ranked images from a bag of words vocabulary search using $td - idf$ scoring.

2.5.2 Inverted Files

Each image is represented as a set of visual words (VW). For each visual word an inverted file is attached that stores the respective identifier of each image and the frequency of the visual word (i.e. number of occurrences). Image query is performed using a weighted voting scheme on the global inverted file table, a concept borrowed from text retrieval [Brin and Page, 1998]. The memory footprint to represent the occurrence of a visual word in an inverted file is as little as 6 byte, assuming that the image ID is stored as an integer and the term frequency as short integer. Hence, SIFT keys requiring 128 byte can be compressed into 6 $byte$ corresponding to 4.6% of the raw keypoint size. If multiple instances of the same visual word in an image occur, the compression is even larger and leads to $0.046/n$, where n is the number of similar visual words. Since all documents are indexed by the terms they contain, the process of generating and storing document representations is called indexing [Singhal, 2001]. The inverted file data structure allows to efficiently store and index a global image database with low memory requirements. For instance the raw 12 Million extracted SIFT keys from the ukbench[1] database (12000 images) require about $1.5GB$ of memory. Instead, the respective inverted file structure of a $L = 3, K = 50$ vocabulary tree quantization requires $15MB$ to store the visual words (hierarchical vocabulary) and $68MB$ for the inverted file table, only.

2.5.3 Building the Visual Vocabulary

Creating a visual vocabulary from a large set of feature descriptors is a challenging task. In order to build a vocabulary tree of M leave nodes, $N >> M$ data points are required. We choose typically $N > 10M$, which means that on average ten data samples are associated to each cluster center. The large amount of required data (e.g. 10 Million SIFT key for a vocabulary with 1 Million leave nodes) makes the usage of complex and memory intensive clustering algorithms like mean-

[1] http://www.vis.uky.edu/ stewe/ukbench/

shift [Comaniciu and Meer, 2002], spectral and agglomerative clustering virtually impossible. However, k-means clustering is feasible since it only requires linear memory $O(k + N)$ in the number of cluster k and datapoints N.

GPU k-means Clustering

The k-means problem can be defined as follows: given a set of data points $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, N$, where each observation is a d-dimensional real vector, k-means aims to partition the N observations into k clusters $k \in \mathbb{N}^+ < n$. The goal is to find an assignment of data points to clusters as well as the centroids $\mathbf{c}_j \in \mathbb{R}^d$ such that the sum of squares of the distances of each data point to its closest vector $\mathbf{c}_j$, is a minimum. Thus, the optimal set $\mathcal{C}$ of k centroids can be found by minimizing,

$$\phi = \sum_{i=1}^{N} \sum_{j=1}^{k} r_{ij} ||\mathbf{x}_i - \mathbf{c}_j||^2 \tag{2.18}$$

where $r_{ij} \in \{0, 1\}$ is a set of binary variables describing which of the k clusters the data point $\mathbf{x}_i$ is assigned to. This problem has been proven to be NP-hard [Drineas et al., 2004] but approximate (non optimal) solutions exist, e.g. the well known Lloyd [Lloyd, 1982] algorithm.

For high-dimensional data such as the SIFT-descriptors $\mathbf{x} = (x_1, \ldots, x_{128})$ the vectors direction is more important than the magnitude. Hence a unit vector representation ($||\mathbf{x}|| = 1$) that additionally accounts for gradient variations is used. The unit vector representation makes the descriptor more robust to illumination changes, hence normalized cluster centers $||\mathbf{c}|| = 1$ are required. The additional constraint is accounted by a spherical k-means algorithm (i.e. k-means on a unit hypersphere) that aims to maximize the average cosine similarity objective. The main difference to the standard k-means algorithm is that the re-estimated mean vectors are normalized to unit-length, which implies that the underlying probabilistic models are not Gaussian any more. However, given the fact that cluster centers are compact, only a small error is induced by this approximation and convergence is guaranteed. Since $\mathbf{x}$ and $\mathbf{y}$ are both unit vectors, the cosine similarity is equivalent to the Euclidean distance,

$$||\mathbf{x} - \mathbf{y}||^2 = ||\mathbf{x}||^2 + ||\mathbf{y}||^2 - 2\mathbf{x}^\top \mathbf{y} = 2 - 2\mathbf{x}^\top \mathbf{y} \tag{2.19}$$

which can be efficiently computed by matrix multiplication. With increased number of descriptors, even the k-means clustering algorithm slows down for clustering large descriptor sets. Most of the computational time is spend in calculating the exact nearest neighbors between data points and cluster centers. Cluster center assignment can be sped up using an approximated k-means algorithm as proposed in [Chum et al., 2007], thus reducing the computational complexity for each iteration from $\mathcal{O}(NK)$ to $\mathcal{O}(N \log(K))$. While approximated k-means is computational efficient, the approximation error in high-dimensional spaces (like the SIFT descriptors lives) can be arbitrary large. Therefore, we rely on linear search but take advantage of the computational power of modern graphic processing units to speedup cluster assignment. In particular, we employ a CUDA based [Zechner and Granitzer, 2009] k-means implementation. To this end descriptors are partitioned into blocks of threads and the data point to cluster distances are computed in parallel (see Equation 2.19). The induced speed up of the GPU for hierarchical clustering 4×10^6 SIFT descriptors in a tree with branch factor $K = 512$ and $L = 2$ levels is more than one order of magnitude.

2.5.4 Vocabulary Tree Traversal

A hierarchical vocabulary tree structure enables an efficient quantization of feature descriptors. In practice, a high-dimensional descriptor (e.g. SIFT) that requires 128 bytes can be compressed into a unique integer comprising 4 bytes which gives a large reduction in memory. Furthermore, the hierarchical tree structure allows an extremely fast quantization through a Best Bin First (BBF) [Beis and Lowe, 1997] search strategy. For instance, feature quantization for a vocabulary tree with branch factor K and L levels requires $O(KL)$ dot products, only. Figure 2.7(b) shows the computational efficiency of different vocabulary trees with respect to the number of required dot products. The speedup is considerable. While exhaustive nearest neighbor search of $10K$ descriptors on a $260K$ vocabulary requires $114s$, hierarchical quantization on a $K = 4$, $L = 9$ vocabulary can be done in 40ms (Intel Pentium D 3.2Ghz).

Table 2.1 shows an evaluation of the vocabulary tree quantization performance with respect to different branch factors and tree levels. A training set of $4M$ SIFT descriptors extracted from images of the uk-benchmark[1] dataset was used to build vocabulary trees of different shapes (i.e. different K,L), but with (approximate) constant number of leaf nodes. Next, a training set of $100K$ SIFT-keys was used to asses the quantization performance induced by best bin first search with respect to the different vocabularies. We write $w_{vt}(f)$ to denote the visual world corresponding to feature f determined trough a vocabulary tree traversal and $w_{bf}(f)$ are respective visual words according to exhaustive nearest neighbor search on respective leaf nodes. Performance p is defined as the percentage of visual words that satisfy $w_{vt}(f_i) == w_{bf}(f_i)$,

$$p = \frac{\sum |w_{vt}(f_i) == w_{bf}(f_i)|}{|\mathcal{F}|}. \tag{2.20}$$

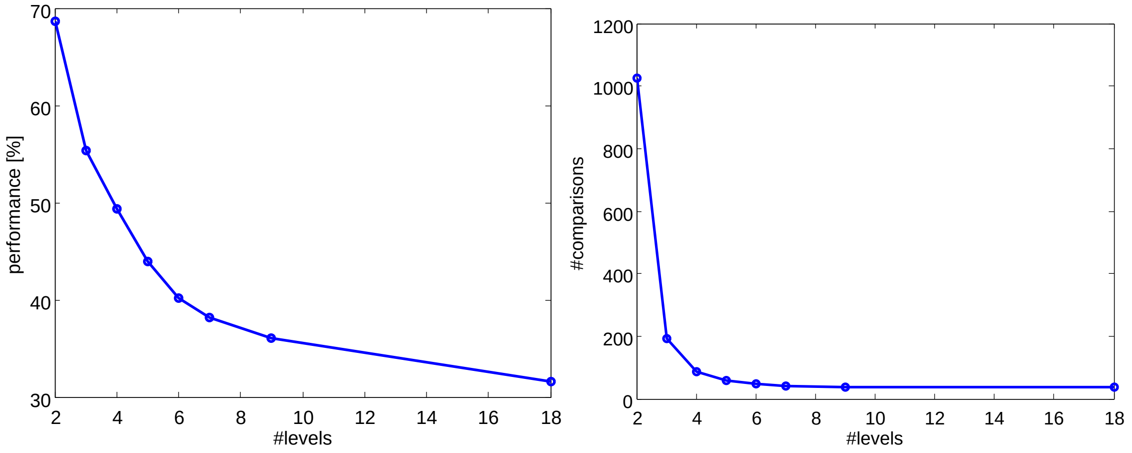
with $f_i \in \mathcal{F}$. We observe that a broader tree yields superior quantization performance since more descriptors are considered. This is in accordance to the gained quantization performance of a greedy N-Best paths search as presented in [Schindler et al., 2007a]. Figure 2.7(a) shows the number of descriptor comparisons for different trees with (approximate) equal number of leave nodes. Table 2.1 gives timings for clustering and tree traversal on the CPU (single threaded) and the parallelized GPU implementation. The speedup induced by the GPU is about ten two twenty. Furthermore, Figure 2.7(a) shows the number of descriptor comparisons for different trees with equal number of leave nodes.

A qualitative evaluation of the quantization performance of a vocabulary tree is shown in Figure 2.8. Image patches that are associated to respective SIFT-keys end up in the same leave node during vocabulary tree training. Note that the patches are visually very similar which confirms the repeatability of SIFT and the effectiveness of the vocabulary tree quantization.

[1] http://www.vis.uky.edu/ stewe/ukbench/

K	L	#leaves	#nodes	T_{kC} [h]	T_{kG} [h]	T_{EC} [s]	T_{EG} [s]	$\frac{\lvert w_{vt}(f_i)==w_{bf}(f_i)\rvert}{\lvert\mathcal{F}\rvert}$
512	2	262144	262656	15.7	1.4	0.63	0.035	0.687
64	3	262144	266304	3.7	0.62	0.12	0.0093	0.554
22	4	234265	245410	1.8	0.51	0.058	0.0053	0.494
12	5	248832	271452	1.36	0.55	0.041	0.0046	0.439
8	6	262144	299592	1.07	0.56	0.034	0.0042	0.402
6	7	279936	335922	0.88	0.58	0.031	0.0041	0.382
4	9	262144	349524	0.71	0.57	0.027	0.0039	0.361
2	18	262144	524286	0.61	0.69	0.027	0.0027	0.316

Table 2.1: Vocabulary tree quantization error dependent on branch factor K and number of vocabulary tree levels L, $w_{bf}(f)$ denotes the exhaustive nearest neighbor assignment and $w_{vt}(f)$ vocabulary tree based assignment to the respective leave node. T_{kC} and T_{EC} time for clustering $4M$ SIFT SIFT keys on the CPU (Intel Pentium D 3.2Ghz) and GPU (Nvidia GeForce GTX280), respectively. T_{EC} and T_{EG}, vocabulary tree traversal time for 5000 SIFT features on the CPU and GPU.



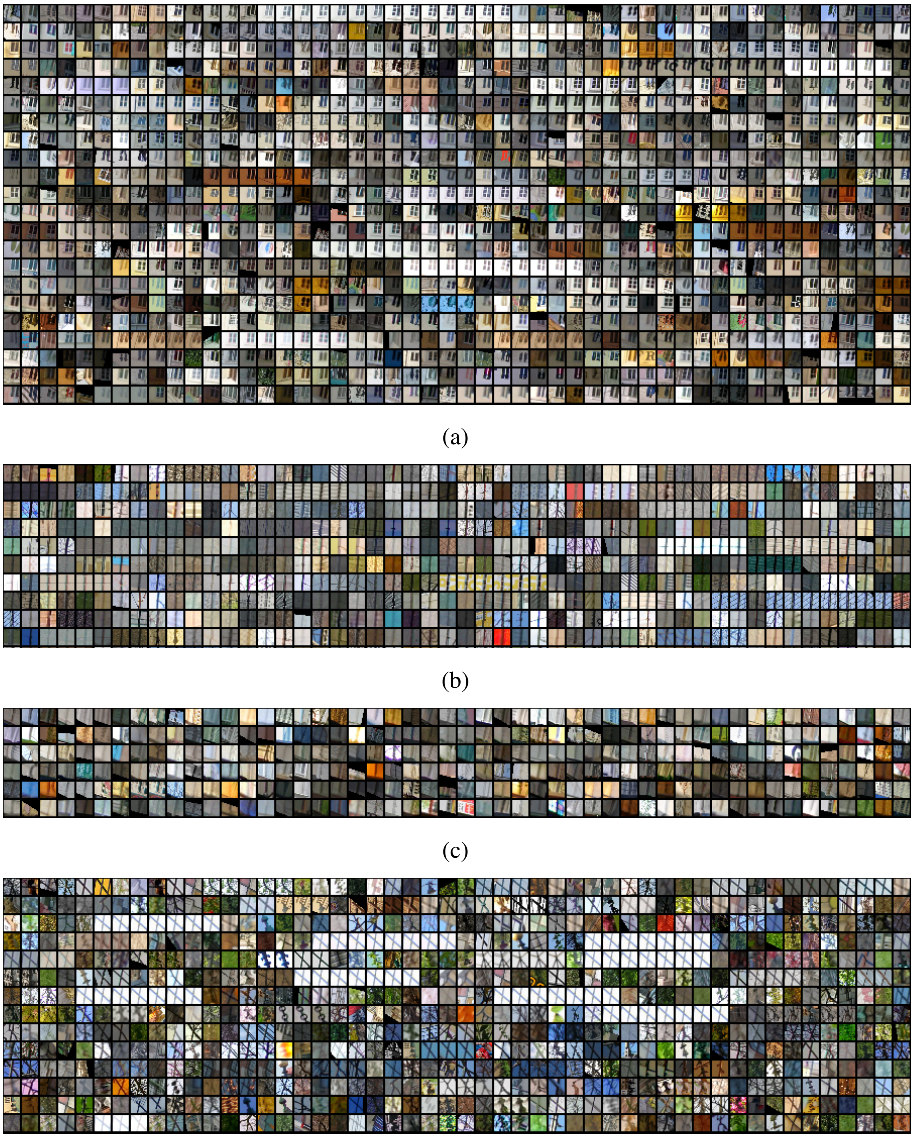
(a)

(b)

Figure 2.7: (a) True positive quantization performance measure of features that are assigned to the nearest neighbor leaf node by the Best Bin First search. (b) Number of comparisons (i.e. $K \cdot L$ dot products) during vocabulary tree traversal using BBF reflecting the amount of quantization computation for different trees.

Implementation Details

In our implementation the vocabulary tree is stored as linear float array of length $128 \times \sum_{i=0}^{L} K^i$ corresponding to a breath first traversal of the tree. This allows fast pointer arithmetic to access



(a)

(b)

(c)

(d)

Figure 2.8: (a)-(d) four different vocabulary tree leave nodes with respective image patches that are assigned to the corresponding cluster center.

individual quantized features. A vocabulary feature at depth L and position k is indexed by,

$$n = k + \sum_{i=0}^{L-1} K^i \tag{2.21}$$

Figure 2.9 shows a binary vocabulary tree of branch factor $K = 2$ and $L = 4$ levels. Since query features are handled independently, the tree traversal can be performed in parallel for each descriptor. We employ a CUDA-based approach executed on the GPU for faster determination of the respective visual words. The speed-up induced by the GPU is about twenty (Nvidia GeForce GTX280 vs. Intel Pentium D 3.2Ghz) and allows to incorporate more descriptor comparisons, i.e. a deeper tree with a smaller branching factor can be replaced by a shallower tree with a significantly higher number of branches. The intuition is that a broader tree yields a more uniform (hence representative) sampling of the high-dimensional descriptor space [Schindler et al., 2007a].

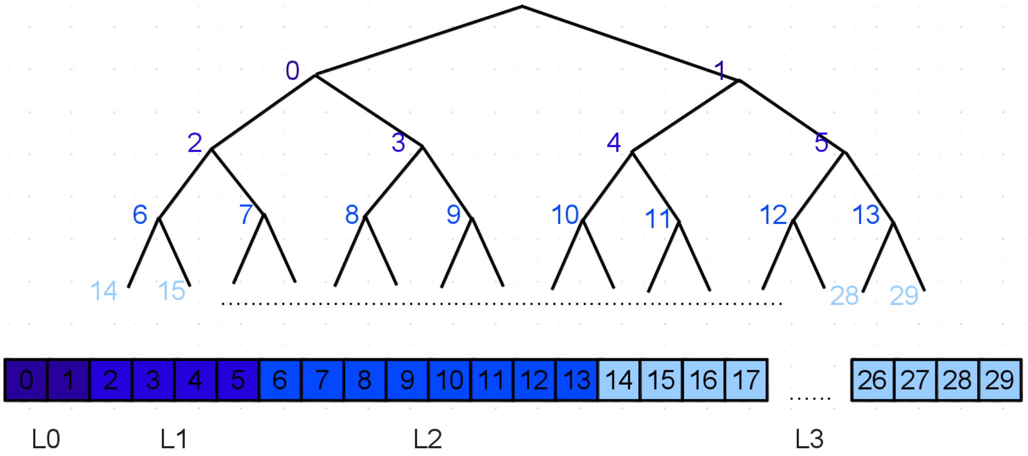


Figure 2.9: Vocabulary tree with branch factor $K = 2$ and $L = 4$ levels.

2.5.5 Scoring Functions

Efficient image search based on vocabulary trees is inspired by large scale text retrieval approaches used for web search engines [Brin and Page, 1998]. Images are first parsed into visual words $w(f)$ according to a precomputed vocabulary. The quantization can be done linear [Sivic and Zisserman, 2003] or hierarchical through a vocabulary tree structure, which has been shown to be more efficient [Nistér and Stewenius, 2006]. Due to the quantization, the query and all the documents in the image database is represented as a sparse vector of visual word occurrences. The similarity of the query and document vector can be computed using different metrics, denoted as scoring functions.

Jaccard Scoring

The Jaccard similiarity coefficient is a statistic for comparing the similarity and diversity of sample sets. In particular this metric is defined as the size of the intersection divided by the size of the union of sample sets,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.22}$$

where A and B are binary occurrences of visual words in image i and j, respectively.

TD-IDF Scoring

A common scoring function is based on the *tf-idf* (term frequency-inverse document frequency) [Sivic and Zisserman, 2003, Nistér and Stewenius, 2006] weighting and computed as follows. Let $\mathcal{V}$ be a vocabulary of visual words then each document is represented by a vector,

$$\mathbf{v}_d = (t_1, \ldots, t_i, \ldots, t_{|\mathcal{V}|}) \tag{2.23}$$

of *tf-idf* weighted word frequencies with components,

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \tag{2.24}$$

where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in the document d, n_i is the number of documents containing term i and N the number of documents in the whole database. Given two *tf-idf* vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the cosine similarity is used to compare documents,

$$\cos(\phi) = \frac{\mathbf{v}_1 \mathbf{v}_2}{||\mathbf{v}_1||\,||\mathbf{v}_2||} \,. \tag{2.25}$$

At the retrieval stage the query vector $\mathbf{v}_q$ is compared with the document vectors $\mathbf{v}_d$ in the databse and the documents are ranked according to the similiarity measure. A similarity yielding a value of 1 means that the documents are exactly the same, 0 independent and -1 exactly opposite.

Probabilistic scoring

Given a query image $\mathcal{Q}$, let R be the set of relevant images (i.e. images that have an overlap to the query) and $\bar{R}$ be the set of non-relevant images (i.e. images that do not share similar visual content). Furthermore, let $P(R|\mathcal{D}_j)$ be the probability that the image $\mathcal{D}_j$ is relevant to the query, and $P(\bar{R}|\mathcal{D}_j)$ be the probability that $\mathcal{D}_j$ is non-relevant to $\mathcal{Q}$. The similarity between a document image $\mathcal{D}_j$ and a query image $\mathcal{Q}$ is then defined as the ratio,

$$sim(\mathcal{D}_j, \mathcal{Q}) = \frac{P(R|\mathcal{D}_j)}{P(\bar{R}|\mathcal{D}_j)} \tag{2.26}$$

We determine the posterior probability by Bayes' rule, i.e.

$$sim(\mathcal{D}_j, \mathcal{Q}) = \frac{P(\mathcal{D}_j|R)P(R)}{P(\mathcal{D}_j|\bar{R})P(\bar{R})} \tag{2.27}$$

where $P(\mathcal{D}_j|R)$ stands for the probability of randomly selecting the image $\mathcal{D}_j$ from the set R of relevant images. $P(R)$ stands for the probability that an image randomly selected from the entire collection is relevant. Since $P(R)$ and $P(\bar{R})$ are the same for all images, eq. (2.27) simplifies to,

$$sim(\mathcal{D}_j, \mathcal{Q}) \sim \frac{P(\mathcal{D}_j|R)}{P(\mathcal{D}_j|\bar{R})} \tag{2.28}$$

Under the assumption of independence among features f_k and respective visual words $v_i = w(f_k)$ we can write,

$$sim(\mathcal{D}_j, \mathcal{Q}) \sim \frac{(\prod_{g_i(\mathcal{D}_j)=1} P(v_i|R)) \times (\prod_{g_i(\mathcal{D}_j)=0} P(\bar{v}_i|R))}{(\prod_{g_i(\mathcal{D}_j)=1} P(v_i|\bar{R})) \times (\prod_{g_i(\mathcal{D}_j)=0} P(\bar{v}_i|\bar{R}))} \tag{2.29}$$

where $P(v_i|R)$ is the probability that the visual word v_i is present in an image randomly selected from the set R of relevant images. On the other hand, $P(\bar{v}_i|R)$ stands for the probability that the visual word v_i is not present in an image with a potential overlap to the query. The probability $P(v_i|R)$ depends on the vocabulary quantization error, the image content overlap, occlusions and the feature repeatability. We simply accumulate these effects into one universal value p_1,

$$p_1 = p_{quant.} \times p_{overlap} \times p_{rep.} \tag{2.30}$$

where $p_{quant.}$ accounts for the probability of exact quantization $P(w(f_i) \equiv w(f_j)|sim(f_i, f_j) > \theta)$, i.e. the probability that two similar features in descriptor space get quantized into the same visual word, $p_{overlap}$ determines the average degree of expected image overlap and occlusion, and $p_{rep.}$ accounts for the repeatability of feature point detection.

Assuming that features vote for unrelated images uniformly (i.e. by pure coincidence), the probability $P(v_i|\bar{R})$ can be estimated as

$$P(v_i|\bar{R}) := \frac{\#\mathcal{D}_j}{\#leaves} = \frac{\#\mathcal{D}_j}{B^L}, \tag{2.31}$$

where $\#\mathcal{D}_j$ denotes the number of leaves in which database image $\mathcal{D}_j$ is appearing in the respective inverted files.

2.6 Detailed Matching

Detailed feature matching deals with establishing putative correspondences between two images. Again, we rely on SIFT as interest point operator and descriptor. A variety of approaches have been proposed to speed up nearest neighbor matching in high-dimensional spaces like the 128-dimensional SIFT descriptor space. Among the most promising methods are randomized kd-trees [Anan and Hartley, 2008] with priority search, and hierarchical k-means trees [Fukunaga and Narendra, 1975]. These algorithms are in general designed to run on a single CPU and are known to provide speedups of about one or two orders of magnitude over linear search, but the speedup comes with the cost of a potential loss in accuracy [Muja and Lowe, 2009]. Given feature vectors of unit length, the Euclidean distance between feature vectors can be derived from the cosine similarity (see Equation 2.19). Let $\mathbf{f}$ and $\hat{\mathbf{f}}$ be two feature vectors of unit length, the Euclidean distance between them writes as,

$$||\mathbf{f} - \hat{\mathbf{f}}||_2^2 = 2 - 2\sum_{i=1}^{K} \mathbf{f}_i \cdot \hat{\mathbf{f}}_i \tag{2.32}$$

Hence, matching can be implemented as a dense matrix multiplication,

$$\begin{array}{c} \mathbf{f}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{f}_N \end{array} \begin{pmatrix} d_{1,1} & .. & d_{1,K} \\ .. & .. & .. \\ .. & .. & .. \\ .. & .. & .. \\ .. & .. & .. \\ .. & .. & .. \\ .. & .. & .. \\ d_{N,1} & .. & d_{1,K} \end{pmatrix} \begin{array}{c} \begin{array}{ccccc} \hat{\mathbf{f}}_1 & \cdots & \cdots & \cdots & \hat{\mathbf{f}}_M \end{array} \\ \begin{pmatrix} d_{1,1} & .. & .. & .. & d_{1,M} \\ .. & .. & .. & .. & .. \\ d_{K,1} & .. & .. & .. & d_{K,M} \end{pmatrix} \end{array} = \begin{pmatrix} c_{1,1} & .. & .. & .. & c_{1,M} \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ c_{N,1} & .. & .. & .. & c_{N,M} \end{pmatrix} \tag{2.33}$$

Efficient linear algebra implementation on the CPU (BLAS) and GPU (CUBLAS) exist. Dense matching implemented as a dense matrix multiplication on the GPU currently achieves timings that can be performed faster or on a par with approximate nearest neighbor search, but delivers the exact solution. Table 2.2 gives detailed timings for different matching algorithms depending on the number of features.

#features (K=128)	GPU (CUBLAS) Nivdia GTX 280	BLAS (ATLAS) Pentium 3.2 GHz	Kd-tree bbf Pentium 3.2 GHz
1000 × 1000	0.0082	0.0486	0.041
5000 × 5000	0.044	1.1	0.22
10000 × 10000	0.23	4.5	0.48

Table 2.2: Timing comparisons of different feature matching algorithms depending on the number of features to match.

One approach to determine correspondences between a feature $\mathbf{f} \in \mathcal{F}$ of the first image and the features $\hat{\mathbf{f}} \in \hat{\mathcal{F}}$ of the second image is to determine the nearest neighbor in descriptor space,

$$\mathbf{f}_{nn} = \underset{\hat{\mathbf{f}} \in \hat{\mathcal{F}}}{\operatorname{argmin}} ||\mathbf{f} - \hat{\mathbf{f}}||_2^2 \tag{2.34}$$

Using the nearest neighbor assignment in general leads to a large number of incorrect correspondences since in general only a fraction of features in each image can be successfully re-detected and matched between images. As reported in [Turcot and Lowe, 2009], on average only about $5 - 20\%$ of extracted features between two visually related images are repeatable. Note, this value varies with baseline, occlusion and depends on the scene structure and texture. As pointed out in [Lowe, 2004], using a global threshold on the Euclidean distance between features does not perform well, as some descriptors are more discriminative than others. A more efficient measure is obtained by considering the distance ratio of the closest neighbor $d(f, f_{1^{st}})$ to the second closest neighbor $d(f, f_{2^{nd}})$,

$$\frac{d(f, f_{1^{st}})}{d(f, f_{2^{nd}})} < \tau_r \tag{2.35}$$

where τ_r is a fixed threshold. A reasonable value for τ_r is in the range of $[0.6 \ldots 0.9]$, Lowe suggest to use $\tau = 0.8$. This measure ensures that the closest neighbor is significantly closer than all other matching candidates, hence the probability of wrong matches is substantially decreased, especially due to repetitions.

2.7 Geometric Verification

In a structure from motion system, image correspondences are usually determined by automatic feature matching approaches, as described in Section 2.6. Such methods rely on local, repeatable interest points f and associated descriptors that are determined from a local image patch. The intuition is that descriptors of homologous feature points are likely to be nearest neighbors in descriptor space under some norm (e.g. Euclidean distance). While this condition is often true for a large fraction of true correspondences (inliers), mismatches (outliers) still occur since no exact matching procedure exists [Horn and Schunck, 1981]. Fortunately, spurious matches (outliers) can be determined and eliminated using the epipolar constraint $\mathbf{x}'^{\top} F\mathbf{x} = 0$ (see Section 2.2) and robust estimation methods. We employ a RANSAC [Fischler and Bolles, 1981] based approach for robust geometric verification using the Five-Point algorithm (see Section 2.2.1). Since the epipolar constraint is only valid for perfect measurements, in practice the following cost function is used,

$$d_{\perp}^2 = \frac{(\mathbf{x}'^{\top} F\mathbf{x})^2}{(F\mathbf{x})_1^2 + (F\mathbf{x})_2^2 + (F^{\top}\mathbf{x}')_1^2 + (F^{\top}\mathbf{x}')_2^2} \quad \left[\text{pixel}^2\right] \tag{2.36}$$

where $(F\mathbf{x})_j^2$ is the square of the j-th entry of vector $(F\mathbf{x})$. This simplified cost function, also denoted as Sampson Distance, is the first order approximation of the reprojection error [Hartley and Zisserman, 2000]. The minimization of the Sampson Distance only involves the optimization of the seven parameters of the fundamental matrix, without reconstructing the 3-space points. Although the Sampson approximation is not optimal, in general it gives extremely good results and is therefore a widely used objective function [Ma et al., 2003].

2.7.1 Random Sample Consensus

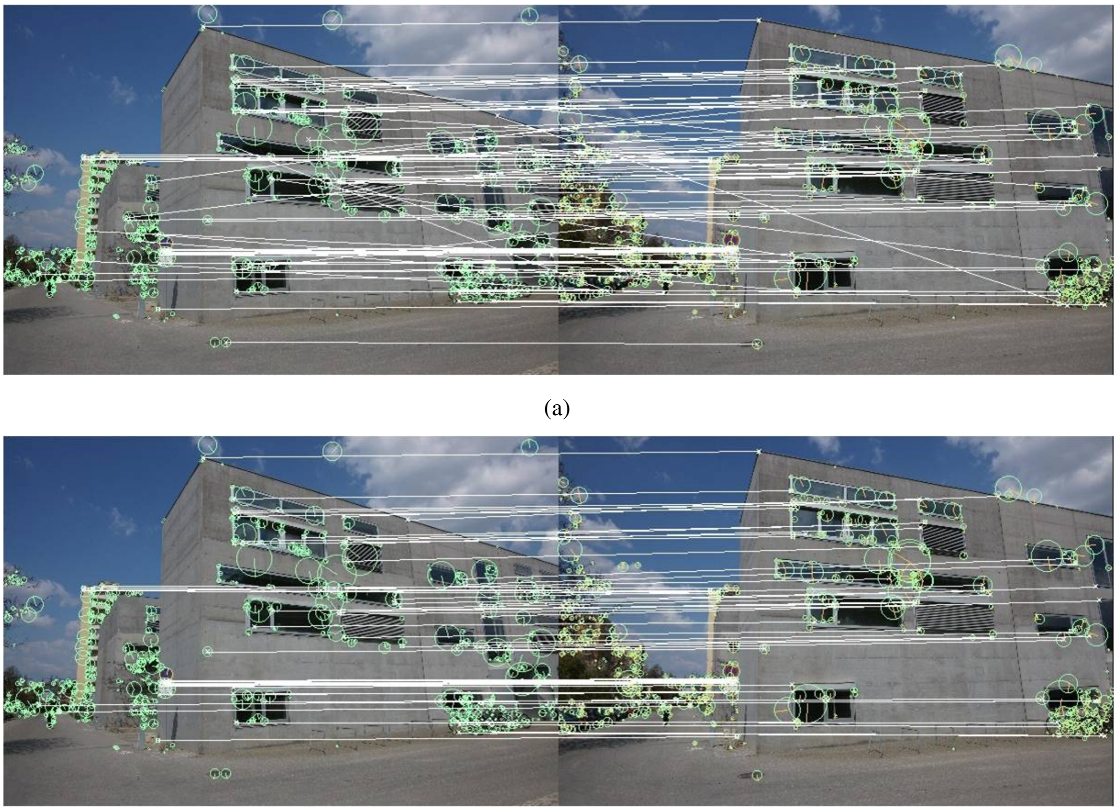
Random Sample Consensus (RANSAC) [Fischler and Bolles, 1981] offers and effective method for robust model fitting to noisy data. RANSAC acts as a hypothesize and verification framework. The model (hypothesis) is computed from a (minimal) number of data points and the support of all matches in agreement is scored. Hence, RANSAC is able to automatically determine the model and the corresponding set of inliers S_i and outliers S_o from the total set of observations $S = S_i \cup S_o$. Figure 2.10 depicts a geometric verification example produced by RANSAC and the Five-Point algorithm. The RANSAC scheme is summarized in Algorithm 1.

RANSAC Timings

The run-time of RANSAC can be written as as a sum of the time to generate a single hypothesis t_h and t_v the time to verify a hypothesis against the full data set. More precisely, the RANSAC

Algorithm 1: Random Sample Consensus (RANSAC)

```
1. Randomly select a minimal subset of $s$ data points from the whole set $S$ of observations and
   compute the model from the subset only.
2. Determine the set of data points $S_i \subseteq S$ which are within a distance threshold $t$ of the model. The
   consensus set $S_i$ defines the inliers of $S$.
3. If $S_i$ contains more inliers than some threshold $T$, re-estimate the model using all the points in $S_i$
   and terminate.
4. If the size of $S_i$ is less than $T$, select a new subset and repeat steps 1,2,3.
5. After a maximal number of $N$ trials with $N = f(|S_i|)$, the largest consensus set $S_i$ is selected and
   the model is re-estimated using all the points in the subset $S_i$.
```



(a)

(b)

Figure 2.10: (a) Image pair showing correspondences from SIFT-key matching. (b) Inlier set determined by RANSAC ($N = 500$, $t = 2$) that satisfies the epipolar constraint.

run-time can be formulated as,

$$t_{RANSAC} = \sum_{i=1}^{N} (t_h + t_v) \tag{2.37}$$

where N is the number of iterations the hypothesize-and-verify step is performed. The number of required RANSAC iterations N is computed by,

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \tag{2.38}$$

where, p is the confidence that at least one sample has no outliers for a given size of s samples and an outlier proportion of ϵ. Figure 2.11 shows that the complexity is exponentially in the number of parameters s. In general the outlier proportion is not known a priori, but can be estimated simultaneously during the RANSAC iteration. The adaptive RANSAC concept allows an early termination. There exist a large body of work targeting at the improvement of RANSAC run-time. A thorough survey of recent RANSAC algorithms can be found in [Raguram et al., 2008]. Methods like [Chum et al., 2003] optimize the hypothesis sampling strategy to minimize the number of iterations N, others aim to optimize the process of model evaluation [Nistér, 2005, Raguram et al., 2009]. While the time for hypothesis generation is constant in the number of observations, model evaluation linearly depends on the number of measurements. For instance, evaluation of the Sampson Distance requires $M \times 13ns$, where M is the number of data points, while the optimized Five-Point algorithm requires $40\mu s$ on average (Intel Pentium D 3.2Ghz). With a growing number of data points ($M > 3000$), the RANSAC runtime is governed by the verification process. This is especially true if multi-matches are used for geometric verification as they significantly increase the number of correspondences to be verified. However, model evaluation can be easily parallelized for the execution on graphic processing units. We employ a CUDA based Sampson Distance computation which achieves a speedup of $20\times$ on a current GPU (Nvidia GeForce GTX280) compared to a single core CPU implementation (Intel Pentium D 3.2Ghz). Since the number of matching candidates for high resolution images dominates the hybrid CPU/GPU implementation considerably speeds up geometric verification. For instance, the single core CPU implementation requires $0.15s$ for $N = 500$ and $M = 20000$ data-points, while the hybrid approach runs in less than $0.03s$.

2.7.2 Confidence of Epipolar Geometry

RANSAC delivers a set of correspondences that are geometrically consistent (inliers) and the relative orientation between image pairs. Furthermore, the inlier fraction $w = \frac{|S_i|}{|S_i \cup S_o|}$ and the number of used RANSAC iterations determines a confidence level p,

$$p = 1 - (1 - w^s)^N \tag{2.39}$$

that at least one sample has no outliers for a given size of samples s. Normally an epipolar geometry is accepted for $p = 0.99$. However, as pointed out in [Tordoff and Murray, 2002] the RANSAC stopping criterion is often optimistic and for image pairs with many correspondences the probability of determining a sufficiently large inlier set just by coincidence is high. This is especially true for image pairs where repeating structures occur. Therefore, rather than relying on the raw number of inliers $m = |S_i|$ between view i and j we determine an effective number of

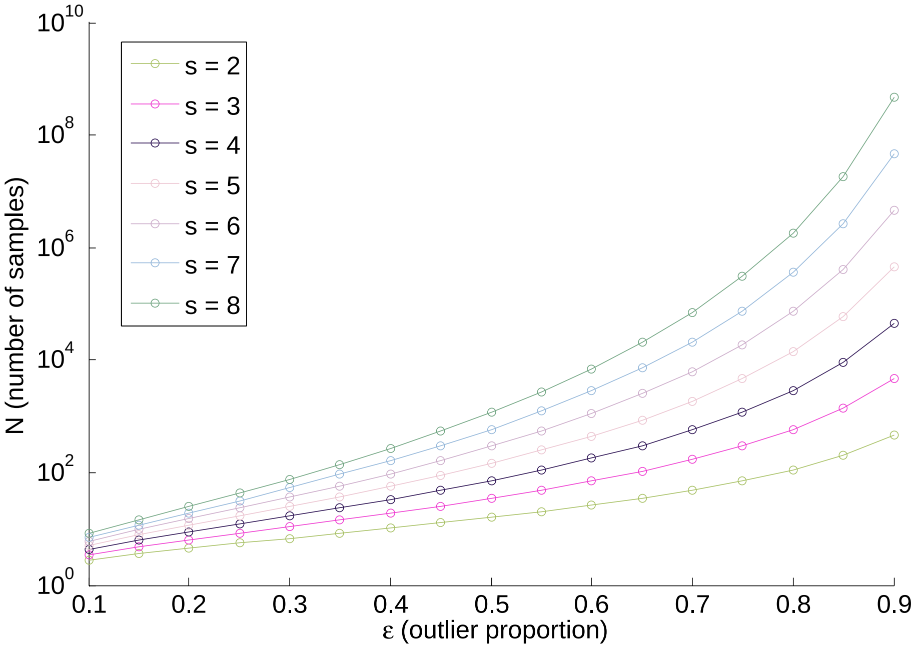


Figure 2.11: The number of N samples required to ensure with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers, ϵ.

inliers,

$$m^* = m \min(c_i, c_j), \tag{2.40}$$

where c_i and c_j is a measure of feature coverage,

$$c_*(S_i) = \begin{cases} 0 & |S_i| < \alpha \\ \frac{A(S_i, r)}{A_\boxdot} & \text{otherwise} \end{cases} \tag{2.41}$$

where α is a minimal number of required inliers (e.g. $\alpha = 10$ in our experiments), $A_\boxdot$ denotes the total image area and $A(\mathcal{F}_{ij}, r)$ is the resulting area that the feature points $\mathcal{F}_{ij}$ cover after applying a dilation operation with a circular structuring element of radius $r = \sqrt{\frac{A_\boxdot}{|F_{ij}|}}$. In addition to the raw number of inliers that determines the confidence of the relative orientation result, the coverage criterion further takes the spatial distribution of correspondences into account. As a consequence, convergent views that have well distributed correspondences produce a higher score than epipolar pairs with the same number of correspondences but with random point distribution. The idea of point coverage is depicted in Figure 2.12. While the number of features is equal in (a) and (b), the uniform spatial distribution of point features in (a) can be regarded as more reliable than the one shown in (b). Hence the effective inlier fraction writes as,

$$w^* = \frac{m^*}{m^* + |S_o|} \tag{2.42}$$

with $w^* \leq w$. Taken this measure in RANSAC normally results in more iterations but largely reduces the fraction of wrong epipolar relations.

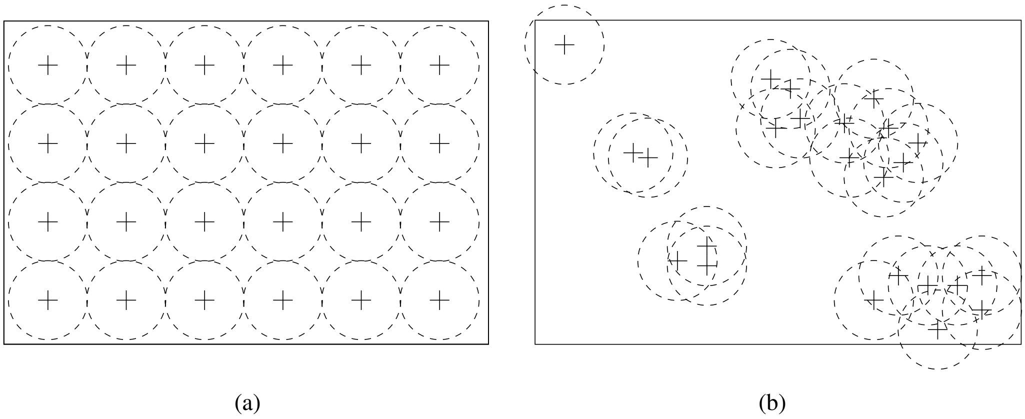


Figure 2.12: Coverage of (a) uniformly and (b) non-uniformly distributed image measurements.

2.7.3 Epipolar Graph

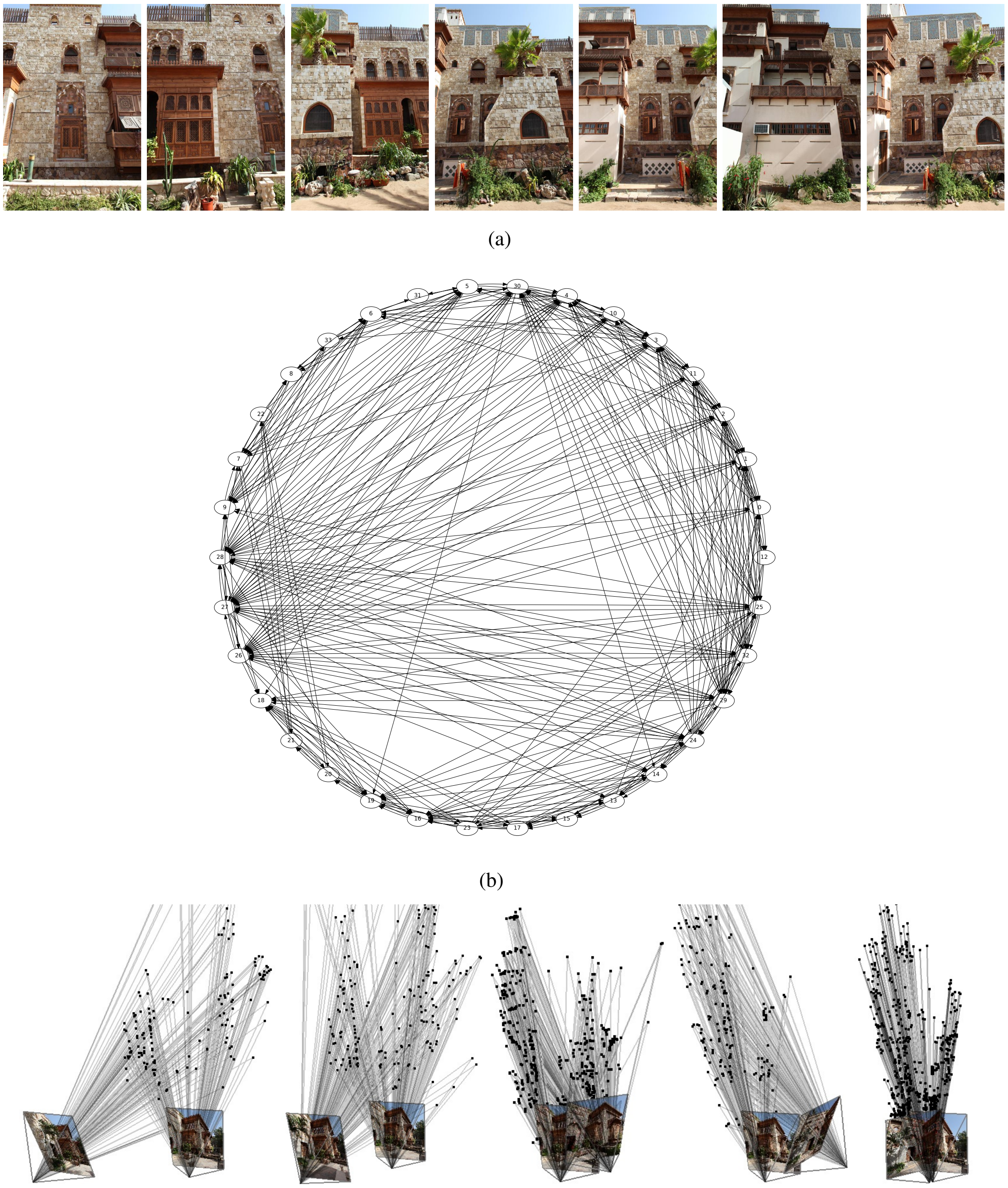
The output of the automatic matching procedure is a graph structure denoted as epipolar graph $\mathcal{G}$, that consists of the set of vertices $\mathcal{V} = \{V_1 \dots V_N\}$ corresponding to the images and a set of edges $\mathcal{E} = \{e_{ij} | i, j \in \mathcal{V}\}$ that are pairwise reconstructions, i.e. relative orientations between view i and j, $e_{ij} =< P_i, P_j >$,

$$P_i = K_i[I, 0] \text{ and } P_j = K_j[R, t] \tag{2.43}$$

and a set of triangulated points with respective image measurements. Next a linear triangulation method [Hartley and Zisserman, 2000] is used estimate the 3D point location. This procedure is followed by a pruning step that discards points at infinity and points that do not satisfy the cheirality criterion. Figure 2.13 shows a typical epipolar graph and samples of pairwise epipolar geometries. Based on the epipolar graph, connected components are extracted and point tracks over multiple views are generated. The tracks are used later for structure initialization.

2.7.4 Track Generation

The epipolar graph $\mathcal{G}$ stores a set of relative orientations and feature correspondences between view pairs $< V_i, V_j >$. Every image V_i is matched to a number of neighboring images and the matching information is stored locally in every node. Note, $\mathcal{G}$ is a directed graph, a match $V_i \rightarrow V_j$ does not necessarily imply $V_j \leftarrow V_i$. Next, for each image node V_i of the graph, point measurements are aggregated to tracks $m = (< x_1^i, y_1^i >, < x_2^j, y_2^j > \dots, < x_n^k, y_n^k >)$, where $f =< x^i, y^i >$ represent feature locations of image I_i. Since point tracks are generated for each image and stored locally, at first instance, the set of point tracks $m \in \tilde{\mathcal{M}}$ is redundant, i.e. a feature point f from image V_i can take part in different tracks. The point tracks are later used for global optimization in bundle adjustment. From a practical viewpoint, redundant measurements are not desired since it involves more parameters in the optimization framework, hence we are interested in a minimal representation. To this end we determine a subset of tracks $\mathcal{M} \subseteq \tilde{\mathcal{M}}$ that covers every matched feature correspondence of the epipolar graph only once. This is an instance



(c)

Figure 2.13: (a) Seven sample images of an outdoor scene and (b) corresponding epipolar graph by matching all 30 images. Nodes represent images, vertices are valid epipolar geometries. (c) Relative orientations results (i.e. epipolar geometries) with respect to image 20.

of the set cover problem [Karp, 1972a], one of the earliest problems known to be NP-complete. We us a greedy approach [Johnson, 1974] to efficiently determine a minimal set of tracks that are subsequently used to initialize the sparse 3D structure.

2.8 Geometric Estimation

The epipolar graph $\mathcal{G}$ encodes relative orientations and pairwise reconstructions. Chaining all relative orientations together should result in a global consistent 3D structure. In [Nister et al., 2007] the proof is given that such a problem is in general NP-hard when missing data is allowed. Here, missing data refers to that 3D points are not always observed in all views which is inherent in large scale multiple view geometry due to occlusion and the limited repeatability of feature detectors. Furthermore, relative rotations are sometimes prone to errors due to mismatches and local errors accumulate and cause drift. This makes global optimization of camera orientations and 3D structure a challenging problem. Current state of the art reconstruction methods normally follow a greedy, incremental reconstruction approach [Nistér, 2000, Pollefeys et al., 2004, Snavely et al., 2006, Agarwal et al., 2009, Frahm et al., 2010, Gherardi et al., 2010]. These methods run iteratively, starting with a small set of views and repeatably add images and refine 3D points and camera poses. Structure and camera pose refinement is done using nonlinear optimization, also known as bundle adjustment. Such an algorithm is highly sensitive to initialization, time consuming and loop closure is hard to handle.

Closed form batch reconstruction can be done for orthographic cameras by using factorization methods [Tomasi and Kanade, 1992]. However, such methods are difficult to apply to perspective cameras with significant outliers and missing data. Other methods rely on the max-norm cost function [Kahl, 2005, Hartley and Schaffalitzky, 2004] that allows global optimization. Geometric estimation is often solved using a two step approach [Sinha et al., 2010, Dalalyan and Keriven, 2009, Zach and Pollefeys, 2010]. First global camera rotations are estimated considering all pairs [Govindu, 2004, Martinec and Pajdla, 2007]. Second, translations are determined. Global camera rotations can be estimated linearly in a least squares in a manner alike [Govindu, 2001]. Given the known camera rotations, the camera translations and the 3D points can be recovered in a globally optimal manner using quasi convex optimization [Kahl, 2005]. However, this method may completely fail due to a single mismatch while minimizing the maximum reprojection error. To overcome this problem in [Martinec and Pajdla, 2007] a robust approach based on Second order Cone programming is presented. In contrast to that, [Dalalyan and Keriven, 2009] address the problem of translation estimation using a Bayesian framework. The fidelity of the data is measured by the L_∞-norm while the regularization is done by the L_1-norm. Outlier removal is done by solving a linear program (LP) where the number and proportion of outliers is automatically determined. Recently [Crandall et al., 2011] presented a hybrid discrete-continuous optimization method for coarse global image alignment. The method is able to naturally incorporate prior pose evidence from geo-tags and vanishing points.

2.8.1 Global Rotation Registration

Given the epipolar graph $\mathcal{G}$, the initial camera positions and orientations remains to be determined. First, relative rotations R_{ij} between view pairs i and j are upgraded into a consistent set of rotations R_i by solving the (overdetermined) system of equations,

$$R_{ij}R_i = R_j \tag{2.44}$$

subject to the constraint that R_i are orthonormal. As described in [Martinec and Pajdla, 2007], the solution can be obtained by solving the system initially for approximate rotation matrices $\hat{R}_i$ (without satisfying the orthonormality constraint) and subsequently projecting the approximate rotation $\hat{R}_i$ to the closest rotation in the Frobenius norm. This is done by using the singular value decomposition (SVD). Equation (2.44) is normally overdetermined since the epipolar graph consists of a redundant set of relative orientations that contribute to the global structure. Not all epipolar geometries are equally important. In general, relative rotations that are determined by many correspondences can be rated as more confident than orientations that are only supported by a small number of measurements. As suggested in [Martinec and Pajdla, 2007], we consider the number of inliers and reweight each row of equation (2.44) according to a quality criterion that determines the accuracy of an epipolar geometry e_{ij}. Rather than using the raw number of inliers $\mathcal{F}_{ij}$ as suggested in [Martinec and Pajdla, 2007], we compute the weights ω_{ij} as follows,

$$\omega_{ij} = \sqrt{N}\min(c_i, c_j) \tag{2.45}$$

where $N = |\mathcal{F}_{ij}|$ is the number of inliers between view i and j and c_i, c_j is a measure of feature coverage (see Equation (2.41) Section 2.7.2). In addition to the raw number of inliers that determines the confidence of the relative orientation result, the coverage criterion further takes the spatial distribution of correspondences into account. As a consequence, convergent views that have well distributed correspondences produce a higher score than epipolar pairs with the same number of correspondences but with random point distribution. The re-weighted system (2.44) then reads,

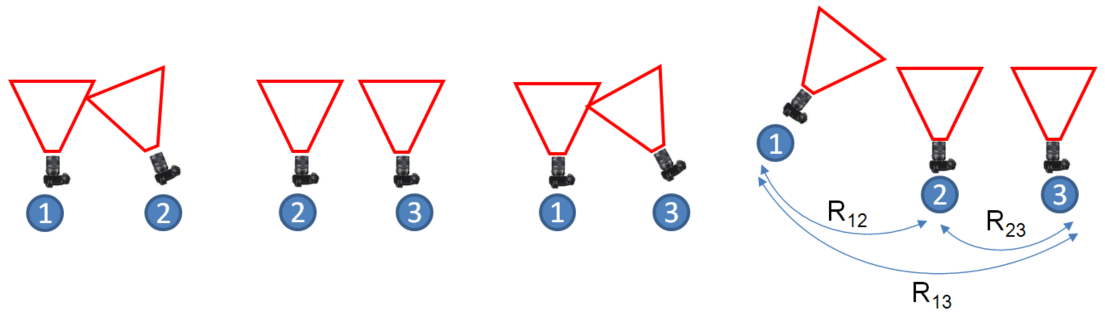
$$\omega_{ij}(R_{ij}\mathbf{r}_i^k - \mathbf{r}_j^k) = \mathbf{0}_{3\times 1} \tag{2.46}$$

for $k = 1, 2, 3$, where $\mathbf{r}_i^k$ are columns of R_i. The system can be efficiently solved by a sparse least squares solver (e.g. using the ARPACK library). The concept is depicted in Figure 2.14.

Given the known rotations, camera centers can be determined by external sensors such as Global Positioning Systems (GPS) [Irschara et al., 2011] or in a globally optimal manner using quasi convex optimization [Kahl, 2005].

2.8.2 Greedy Incremental SfM

In this section we describe a 3D reconstruction approach that incrementally builds and updates structure and cameras from a suitable start configuration. The method is closely related to [Snavely et al., 2006] with some modifications. To reconstruct a consistent 3D model, a robust and reliable start configuration is required. When the initial structure is prone to errors, a subsequent iterative

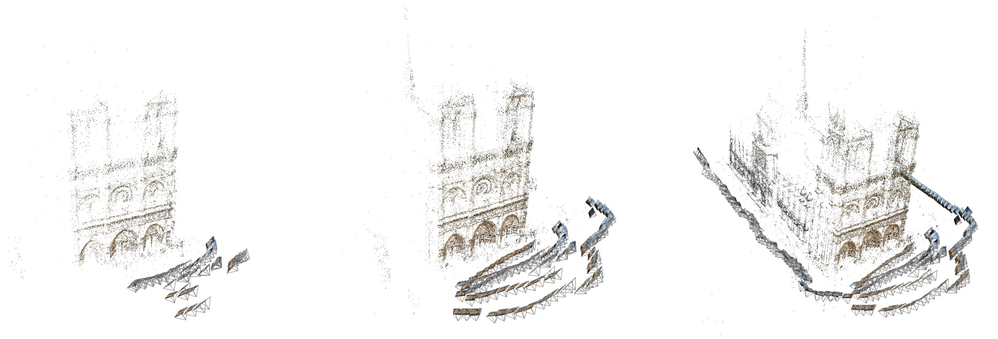


(a) (b)

Figure 2.14: (a) Pairwise relative orientations and (b) globally consistent rotation registration result.

optimization procedure will eventually end up in a wrong local minimum, hence good initialization is critical. As proposed in [Klopschitz et al., 2010] we initialize the geometry in the most connected parts of the graph, therefore the view V^* with highest degree, i.e. the node having the largest number of edges, is determined. Next, all point-tracks corresponding to view V^* are used to compute a global scale factor of the initial structure $R^* =< \mathcal{P}^*, \mathcal{X}^* >$ with $\mathcal{P}^* \subset \mathcal{P}$, $\mathcal{X}^* \subset \mathcal{X}$. Then, bundle adjustment [Triggs et al., 2000, Lourakis and Argyros, 2004] is used to optimize camera orientations P_i and 3D points $\mathbf{X}_j$ by minimizing the reprojection error. The implementation details are described in Section 2.9. Given the initial, optimized structure $\mathcal{R}^*$, each 3D point is back-projected and searched for in every image. We utilize a 2D kd-tree for efficient search and restrict the search radius to a constant factor r_t. Again, given the new measurements, bundle adjustment is used to optimize 3D points and camera parameters. This method ensures strong connections within the current reconstruction.

Next, for every image V that is not reconstructed ($V \notin \mathcal{R}^*$) and has a potential overlap to the current 3D scene (estimated from the epiploar graph $\mathcal{G}$), 2D–to–3D correspondences are established. A three-point pose algorithm [Haralick et al., 1991] inside a RANSAC loop (see Section 2.7.1) is used to insert the position of a new camera $P^\dagger$ with respect to $\mathcal{R}^*$. When a pose can be determined (i.e. a sufficient inlier confidence is achieved), $\mathcal{R}^*$ is updated with $P^\dagger$ and all measurements visible therein. A subsequent procedure expands the current 3D structure by triangulation of new correspondences that are visible in $P^\dagger$. We follow the approach of Snavely et al. [Snavely et al., 2006] and use a priority queue $\mathcal{Q}$ to guide the insertion order. Our insertion order is based on a saliency measure that favors early insertion of images that have a strong overlap with the given 3D structure. Rather than using the raw number of potential 2D–to–3D matches, we compute an effective matching score, as described in Section 2.7.2, that further takes the spatial match distribution of correspondences into account. Whenever a number of N images is added (we use $N = 10$), bundle adjustment is used to simultaneously optimize structure and camera parameters. The iterative view insertion procedure is repeated until $\mathcal{Q}$ is empty. Figure 2.15 depicts intermediate reconstruction results of an incremental reconstruction of the Notre Dame scene after adding 84, 177 and 277 view, respectively.



(a) 84 images (b) 177 images (c) 277 images

Figure 2.15: Incremental 3D reconstruction result of the Notre-Dame Cathedral after adding (a) 84,(b) 177 and (c) 277 images, respectively.

2.9 Bundle Adjustment

Given initial parameters for camera poses and the 3D structure, the goal of bundle adjustment [Triggs et al., 2000, Hartley and Zisserman, 2000, Lourakis and Argyros, 2009] is to find jointly 3D point positions and camera parameters that minimize the error between observed feature locations (correspondences) and projections (predicted image measurements of 3D points). Given a set of measured image feature locations and correspondences, bundle adjustment optimizes camera orientations and structure by minimizing the reprojection error,

$$\mathcal{C}(P_i, \mathbf{X}_j) = \sum_i \sum_j v_{ij} d(P_i \mathbf{X}_j, \mathbf{x}_{ij})^2 \tag{2.47}$$

where the 2D point measurements $\mathbf{x}_{ij}$ are the observations of unknown 3D points $\mathbf{X}_j$ in the unknown cameras P_i and v_{ij} is a binary variable that is 1 if the point $\mathbf{X}_j$ is visible in image P_i and 0 otherwise. Thus, bundle adjustment involves adjusting the bundle of rays between each 3D point and the set of camera centers by minimizing the reprojection error, which is usually expressed as the sum of squares of a large number of nonlinear, real-valued constraints. The minimization of bundle adjustment can be achieved using nonlinear least-squares algorithms. Bundle adjustment is tolerant to missing data (i.e. not every 3D point must be visible in each camera) and is a large sparse geometric parameter estimation problem. Since each camera has six degrees of freedom and each 3-space point three degrees of freedom, a reconstruction involving n point and m cameras requires minimization of $3n + 6m$ parameters. Under the assumption that measurements are independent and Gaussian measurement noise, bundle adjustment provides a Maximum Likelihood parameter estimate. Let $f(\epsilon)$ be the probability distribution of an error ϵ in the measurements, then the probability of a set of measurement with error ϵ_i is given by $p(\epsilon_1, \dots, \epsilon_n) = \prod_i^n f(\epsilon_i)$. In log-space the function reads $-\log(p(\epsilon_1, \dots, \epsilon_n)) = -\sum_{i=1}^n f(\epsilon_i)$ and is suitable for a cost function. However, since there exists no closed-form solution to a non-linear least squares problem, bundle adjustment is a local minimizer and it requires a good initialization to be provided.

For small optimization problems, dense nonlinear least-squares is sufficient but is computationally very demanding when employed to minimize functions depending on a large number of parameters. Hence, an efficient implementation is necessary to handle large problems. The following section gives implementation details for large scale efficient bundle adjustment [Lourakis and Argyros, 2009, Engels et al., 2006] based on the Levenberg Marquardt algorithm.

2.9.1 Levenberg Marquardt

The Levenberg Marquardt (LM) optimization is among the most efficient optimization schemes for non-linear least squares problems. Normally, LM significantly outperforms gradient descent and conjugate gradient methods for medium sized problems [Madsen et al., 2004].

LM tries to minimize a cost function $c(x)$ iteratively by approximating the cost function locally around the current position x with a quadric Taylor expansion

$$c(x + \delta x) \approx c(x) + \nabla c(x)^\top \delta x + \frac{1}{2}\delta x H_c(x)\delta x \tag{2.48}$$

where $\nabla c(x)$ is the gradient,

$$\nabla c(x) = \begin{bmatrix} \frac{\delta c}{\delta x_1}(x) & \cdots & \frac{\delta c}{\delta x_M}(x) \end{bmatrix} \tag{2.49}$$

of c at x and $H_c(x)$ is the Hessian,

$$H_c(x) = \begin{bmatrix} \frac{\delta^2 c}{\delta x_1 \delta x_1}(x) & \cdots & \frac{\delta^2 c}{\delta x_1 \delta x_M}(x) \\ \vdots & \ddots & \vdots \\ \frac{\delta^2 c}{\delta x_N \delta x_1}(x) & \cdots & \frac{\delta^2 c}{\delta x_N \delta x_M}(x) \end{bmatrix} \tag{2.50}$$

of c at x. If we solve for the minimum x by setting the left hand side of Equation 2.48 to zero and take the derivative, we obtain

$$\nabla c(x)^\top + H_c(x)\delta x = 0 \tag{2.51}$$

$$H_c(x)\delta x = -\nabla c(x), \tag{2.52}$$

which is a linear equation in the update vector δx. Finding the solution is known as Newton's method. One can show that quadratic convergence is achieved if the Hessian at the solution is positive definite [Frandsen et al., 1999]. Hence, Newton's method is normally very good in the final stage for the iterations, where x is close to the local minimum x^*. However, there is no guarantee that the quadratic approximation will lead to an update δx that improves the cost function. One method to avoid this behavior is the hybrid LM algorithm based on Newton's method and steepest descent. When improvement in Newton's method fails, steepest descent takes over which guarantees convergence. In the LM algorithm this is achieved by adding some scalar λ to all the diagonal elements of $H_c(x)$. When improvement succeeds, λ is decreased towards zero, taking full advantage of the quadratic approximation which leads to fast convergence near the minimum.

On the other hand, when Newton's method fails, λ is increased, which makes the update tend towards,

$$\delta x = -\frac{1}{\lambda} \nabla c(x), \tag{2.53}$$

which guarantees that improvement will be found for a sufficiently large λ. Under the assumption of Gaussian noise, the squared sum is the optimal metric of all the dimensions of the (N-dimensional) error vector function $f(x)$,

$$c(x) = f(x)^\top f(x). \tag{2.54}$$

However, the squared error cost function is normally not suitable for structure from motion since outliers in the measurement vector are ineluctable. Bundle adjustment allows the replacement of the error vector $f(x)$ by a robust cost function that accounts for outliers as described later in Section 2.9.2. One problem of Newton's method is the implementation of $H_c(x)$, which is complicated. Instead it is very common to use the Gauss-Newton approximation of the Hessian $H_c^*(x) = 2J_f(x)^\top J_f(x)$ where $J_f(x)$ is the Jacobian. This modification is known as the Quasi-Newton method. Here the vector function $f(x)$ is approximated around x with the first order Taylor expansion,

$$f(x + \delta x) \approx f(x) + J_f(x)\delta x, \tag{2.55}$$

with,

$$J_f(x) = \begin{bmatrix} \frac{\delta f_1}{\delta x_1}(x) & \cdots & \frac{\delta f_1}{\delta x_M}(x) \\ \vdots & \ddots & \vdots \\ \frac{\delta f_N}{\delta x_1}(x) & \cdots & \frac{\delta f_N}{\delta x_M}(x) \end{bmatrix}. \tag{2.56}$$

Inserting Equation 2.55 into 2.54, we get

$$c(x + \delta x) \approx f(x)^\top f(x) + 2f(x)^\top J_f(x)\delta x + \delta x^\top J_f(x)^\top J_f(x)\delta x. \tag{2.57}$$

Solving the minimum leads to the following update equation, also know as normal equation [Golub and van Loan, 1996]:

$$J_f(x)^\top J_f(x)\delta x = -J_f(x)^\top f(x). \tag{2.58}$$

Since $2J_f(x)^\top f(x)$ is the gradient of $c(x) = f(x)^\top f(x)$ it follows from Equation (2.51),

$$H_c^*(x) = 2J_f(x)^\top J_f(x). \tag{2.59}$$

Using the Quasi-Newton Method has two advantages over Newton's method. First, computation of the second derivative is not necessary, which is often hard to perform for complex problems. Second, $H_c^*(x)$ and its inverse is normally positive definite,

$$\delta x^\top J_f(x)^\top J_f(x)\delta x > 0 \qquad \forall \delta x \neq 0. \tag{2.60}$$

This allows to multiply the diagonal of $J_f(x)^\top J_f(x)$ by the scalar $(1 + \lambda)$, which leads to the Levenberg-Marquardt algorithm.

Primary Structure

The parameters involved in bundle adjustment (BA) consist of cameras and 3D scene points. More formally, we assume that each camera j is parameterized by a vector $\mathbf{a}_j$ and each 3D point i by a vector $\mathbf{b}_i$. The core feature of bundle adjustment is to take advantage of the primary sparsity. This pattern arises because the parameters of scene points $\mathbf{b}_i$ and cameras $\mathbf{a}_j$ combine to predicted measurements $\mathbf{x}_{ij}$ in the images, while 3D point parameters do not combine directly and camera parameters do not combine directly. Therefore the Jacobian has the structure,

$$J_f = \begin{bmatrix} J_b & J_a \end{bmatrix} \tag{2.61}$$

where J_b is the Jacobian of the error vector f with respect to the 3D point parameters and J_a is the Jacobian of the error vector f with respect to the camera parameters. Therefore, the approximated Hessian reads as,

$$H_c^*(x) = \begin{bmatrix} J_b^\top J_b & J_b^\top J_a \\ J_a^\top J_b & J_a^\top J_a \end{bmatrix} . \tag{2.62}$$

Inserting it into the linear equation system 2.51 leads to,

$$\begin{bmatrix} H_{bb} & H_{ba} \\ H_{ba}^\top & H_{aa} \end{bmatrix} \begin{bmatrix} \delta b \\ \delta a \end{bmatrix} = \begin{bmatrix} c_b \\ c_a \end{bmatrix} , \tag{2.63}$$

where H_{xx} are abbreviations for the elements of $H_c^*(x)$ and $c_b = -J_b^\top f$ and $c_a = -J_a^\top f$. H_{bb} and H_{aa} are block diagonal, where the blocks correspond to points and cameras, respectively. One can use block-wise Gaussian elimination by multiplying Equation 2.63 from the left with the block lower triangular matrix,

$$\begin{bmatrix} H_{bb}^{-1} & 0 \\ 0 & I \end{bmatrix} \tag{2.64}$$

which results in,

$$\begin{bmatrix} I & H_{bb}^{-1} H_{ba} \\ H_{ba}^\top & H_{aa} \end{bmatrix} \begin{bmatrix} \delta b \\ \delta a \end{bmatrix} = \begin{bmatrix} H_{bb}^{-1} c_b \\ c_a \end{bmatrix} . \tag{2.65}$$

The lower left block can be eliminated by subtracting $H_{ba}^\top$ times the first row from the second row. This can be done by multiplying the matrix,

$$\begin{bmatrix} I & 0 \\ -H_{ba}^\top & I \end{bmatrix} \tag{2.66}$$

from the left on both sides, hence one obtains a smaller equation system,

$$(H_{aa} - H_{ba}^\top H_{bb}^{-1} H_{ba}) \delta a = c_a - H_{ba}^\top H_{bb}^{-1} c_b \tag{2.67}$$

$$H_r^* \delta a = c_a^* \tag{2.68}$$

for the camera parameter update δa. This Gaussian elimination step is known as the Schur complement method. The point parameter update c_b can be computed by back-substitution,

$$\delta b = H_{bb}^{-1} c_b - H_{bb}^{-1} H_{ba} \delta a. \tag{2.69}$$

Second Order Sparsity

Note that for very large systems, H_r^* is still sparse due to the fact that not all scene features appear in all sensor views. This is especially true for large scale reconstructions with occlusions and loops. An efficient method to solve Equation (2.68) is thus to use the Cholesky factorization with some appropriate on-the-fly variable ordering or preconditioned conjugate gradient. This can be achieved by the CHOLMOD package [Davis, 2006], which provides a highly-optimized Cholesky decomposition solver for sparse linear systems. In general the complexity of the decomposition of Equation (2.68) will be $O(n^3)$ in the number of variables. However, for sparse matrices, the density will only depend on the density of the Cholesky factor, which depends on the structure of H_r^*. As pointed out in [Mahon et al., 2008], the factor density can range from $O(n)$ to $O(n^3)$. Table 2.3 depicts examples of dense and sparse second oder camera configurations. While for small and strongly connected cameras networks, H_r^* offers a dense structure, for large camera configuration such as aerial networks, H_r^* is sparse and the number of non-zero elements dominates. Note, for the large aerial camera network M3 consisting of 2962 images, the sparse implementation brings a memory saving of more than two orders of magnitude.

2.9.2 Cost function

An important decision to make in nonlinear least squares is the precise form of cost function. If we assume Gaussian measurement noise, minimizing the the least-squares cost function is equal a Maximum Likelihood estimate. However, when outliers are present in the data least-squares is normally not appropriate since this cost function is not robust (i.e. a single wrong measurement can distort the model by the quadratic influence of the error). In the structure from motion problem, outliers often occur due to errors in feature extraction and matching. Outliers are often handled by robust estimation like the RANSAC algorithm [Fischler and Bolles, 1981] based on the epipolar geometry or the 2D-3D absolute pose problem. A very large fraction of outliers can be handled by those methods, but outliers often still occur and thus the least squares cost function in bundle adjustment is not appropriate. Bundle adjustment allows to easily incorporate robust cost functions that are able to handle outliers. In its basic implementation Levenberg-Marquardt minimizes a squared vector norm,

$$c(x) = f(x)^\top f(x) = \sum_i ||\epsilon||^2 \tag{2.70}$$

with $\epsilon = ||\mathbf{x}_i - \hat{\mathbf{x}}_i||$. A robust cost function can be implemented by re-weighting the error vector $\epsilon_i' = w_i \epsilon_i$ such that,

$$||\epsilon_i'||^2 = w_i^2 ||\epsilon_i|| = C(||\epsilon_i||). \tag{2.71}$$

Therefore it follows $\sum_i C(||\epsilon_i||) = \sum_i ||\epsilon_i||^2$ as desired where,

$$w_i = \frac{\sqrt{C(||\epsilon_i||)}}{||\epsilon_i||}. \tag{2.72}$$

The weighting w_i is often called attenuation factor since it seeks to attenuate the cost of the outliers. Typical cost functions for bundle adjustment include,

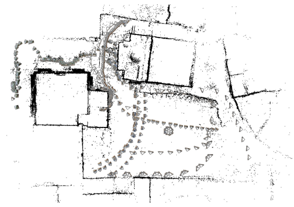
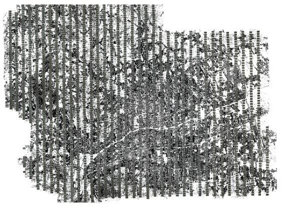
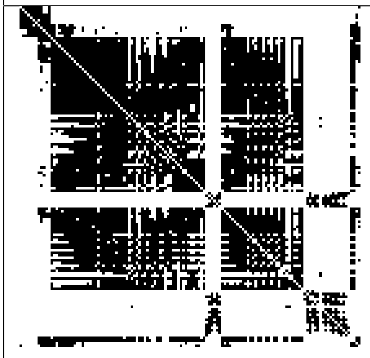
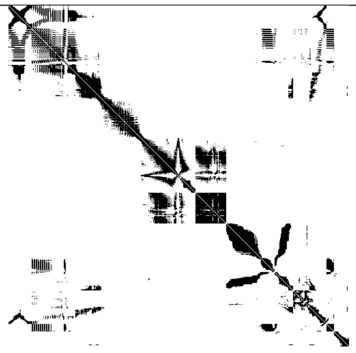
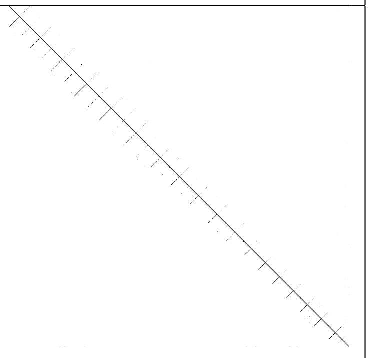
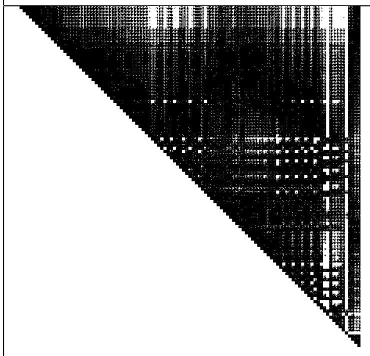
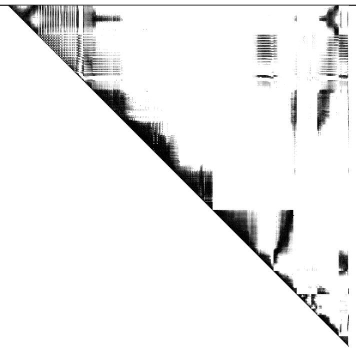
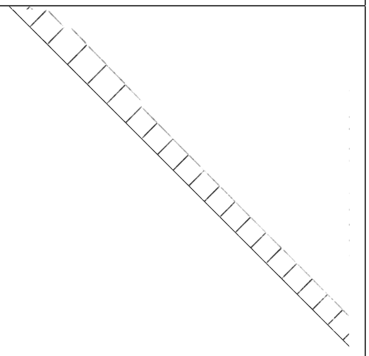
M1	M2	M3
 #images: 109 #points: 6933 #measurements: 227594	 #images: 397 #points: 128743 #measurements: 827859	 #images: 2962 #points: 3775128 #measurements: 12690926
 connectivity: 44%	 connectivity: 8.5%	 connectivity: 0.35%
 H_r^* size: 654×654 #non zeros: 402084 H_r^* density: 94%	 H_r^* size: 2382×2382 #non zeros: 1315980 H_r^* density: 23%	 H_r^* size: 17772×17772 #non zeros: 2242902 H_r^* density: 0.71%

Table 2.3: M1-M3: 3D reconstruction of different scenes. (M1) small scale 3D reconstruction of one building facade, (M2) medium size terrestrial building reconstruction, (M3) large aerial camera network. While the camera network M1 leads to a dense reduced normal equation, H_r^* is sparse for $M2$ and $M3$ due to the fact that not all scene features appear in all cameras.

- **Squared error**

$$C(\epsilon) = \epsilon^2 \tag{2.73}$$

- **Huber**

$$C(\epsilon) = \begin{cases} \epsilon^2 \text{ for } |\epsilon| < b \\ 2b|\epsilon| - b^2 \text{ otherwise} \end{cases} \tag{2.74}$$

- **Blake-Zisserman**

$$C(\epsilon) = \begin{cases} \epsilon^2 \text{ for } |\epsilon| < b \\ b^2 \text{ otherwise} \end{cases} \tag{2.75}$$

- **Sigma**

$$C(\epsilon) = \begin{cases} \epsilon^2 \text{ for } |\epsilon| < b \\ 2b|\epsilon| - b^2 \text{ for } b < |\epsilon| < \sigma b \\ b^2(2\sigma - 1) \text{ otherwise} \end{cases} \tag{2.76}$$

- **Cauchy**

$$C(\epsilon) = b^2 \log(1 + \epsilon^2/b^2) \tag{2.77}$$

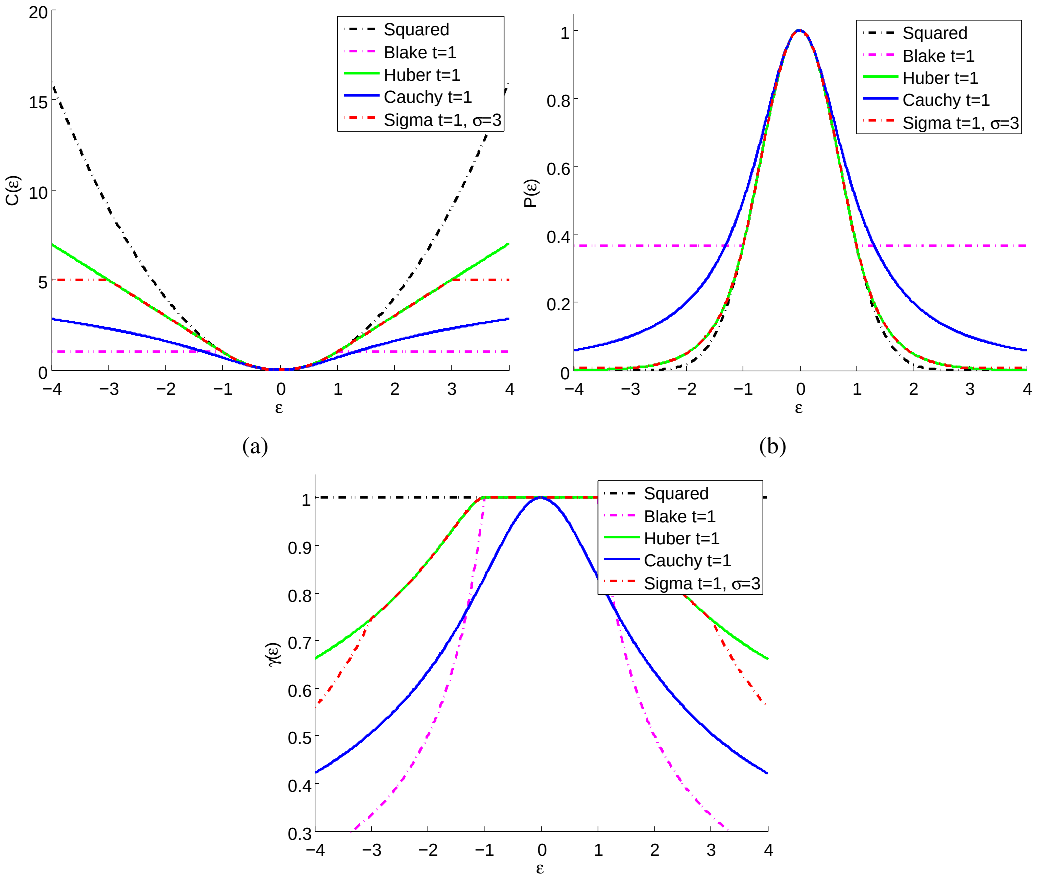
Graphs corresponding to the individual cost functions are depicted in Figure 2.16(c). All but the Squared error cost function seek to deemphasize the cost of outliers once the error exceeds a certain threshold. Note, Blake-Zisserman, Sigma and Cauchy are non convex and hence many local minima may exist. One important role has the Huber cost function that takes the form of a quadric for small values of the error and is linear for values of ϵ beyond a given threshold. This cost function has the very desirable property of being convex while retaining the outlier stability of the $L1$ cost function.

2.9.3 Evaluation of Cost Functions

In this section we asses the quality of different cost functions for bundle adjustment. The experiments investigate the influence of outliers and Gaussian noise for different ground truth camera networks and respective 3D points from typical SfM datasets. Furthermore we perform real world experiments on data acquired with a micro aerial vehicle where GPS/INS pose information is used to initialize camera parameters for bundle adjustment.

Synthetic Experiments

Camera network configurations and sparse points are taken from existing structure from motion models corresponding to realistic scenes listed in Table 2.4. The provided image measurements are corrected to reflect a noise free ground truth, i.e. each measurement $\hat{x}$ is replaced by its projection $x = P\mathbf{X}$. First, image measurements are perturbed by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with zero mean and standard deviation σ. Second, outliers are added, to a fraction of measurements. Outliers are



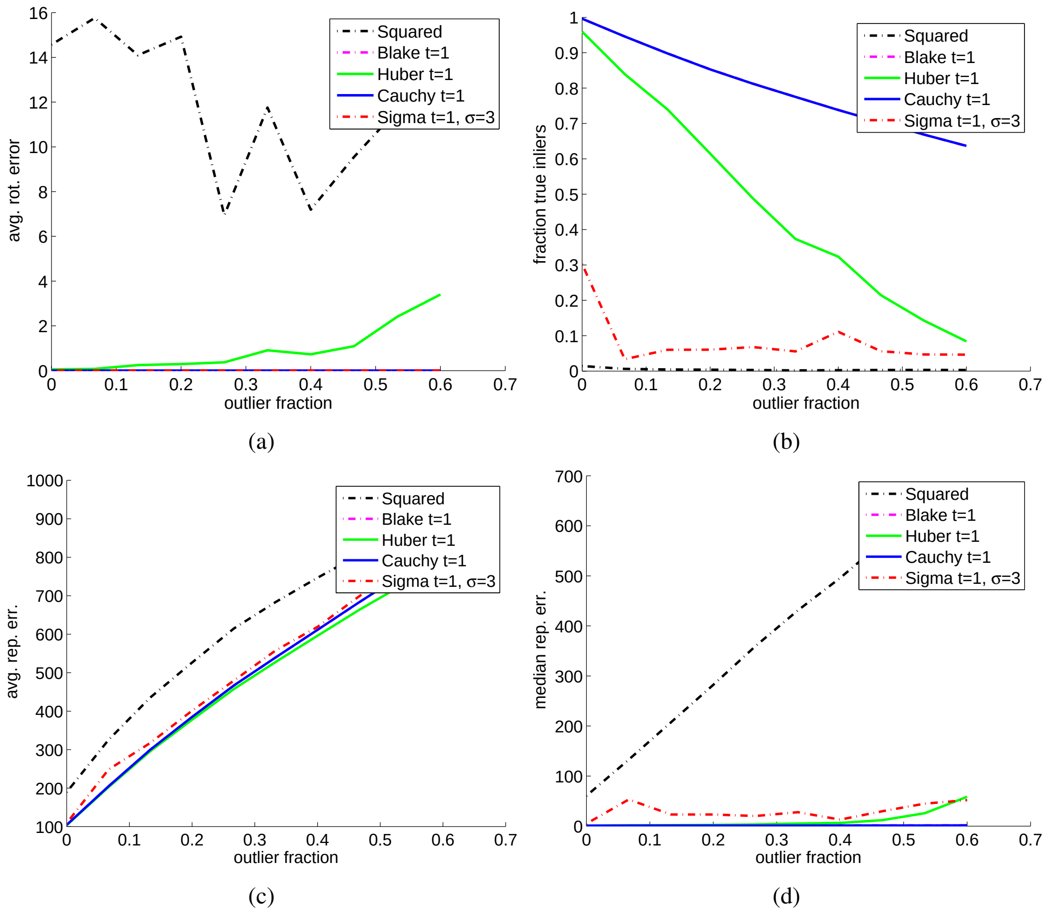
(a)

(b)

(c)

Figure 2.16: Comparison of different cost functions $C(\epsilon)$ dependent on the measurement error ϵ. (a) cost functions $C(\epsilon)$, and corresponding PDFs (b) and (c) attenuation factor.

assumed to follow a uniform distribution across the image plane $x_{outlier} = \langle \mathcal{U}(0,w), \mathcal{U}(0,h) \rangle$. Furthermore, the exterior camera parameters $R = (\alpha, \beta, \gamma), t = (t_x, t_y, t_z)$ are perturbed by additive Gaussian Noise. For the rotation components $\sigma = 5°$ is used and the translational components are corrupted by $\sigma = 0.05 d_i$ where d_i is the mean depth of all scene points visible in camera i. Next, linear triangulation is used to get an initial estimate of 3D point locations. Table 2.5 shows the reprojection error and the deviation of the ground truth after bundle adjustment for the different cost functions. It turns out that Cauchy and Blake-Zisserman perform best. The squared cost function constantly fails to improve the reprojection error for model $M1 - M5$ and gives only a slight improvement for model $M6$. In a second experiment cost functions are compared and evaluated with respect to increasing outlier fractions. The results are summarized in Figure 2.17.



(a) (b)

(c) (d)

Figure 2.17: Comparison of different cost functions with respect to the fraction of uniformly distributed outliers for test reconstruction T6. Camera rotations and translations are perturbed by Gaussian Noise with standard deviation $\sigma_r = 3°$ and $\sigma_t = 0.05d_i$, respectively. Furthermore a fraction of $f_p = 0.05$ points is triangulated using perturbed cameras which leads to erroneous 3D point locations.

Real World Experiment

We perform real world experiments to test and evaluate different cost functions in bundle adjustment. Bundle adjustment requires an initial estimate of the camera parameter and triangulated points. We use a Micro Aerial Vehicle (MAV) equipped with a custom GPS/INS sensor to acquire geo-referenced images in a 3D World Geodetic System (WGS84). The matching graph is created where relative rotation R_{ij} between view pairs i and j are upgraded into a consistent set of rotations R_i as described in Section 2.8.1. Next, pairwise features are linked into feature tracks as described (see Section 2.7.4). The camera centers are initialized with the rough GPS datum and bundle adjustment is executed to minimize the reprojection error. Different cost functions are evaluated with respect to the final reprojection error. The results are summarized in Table 2.6. While

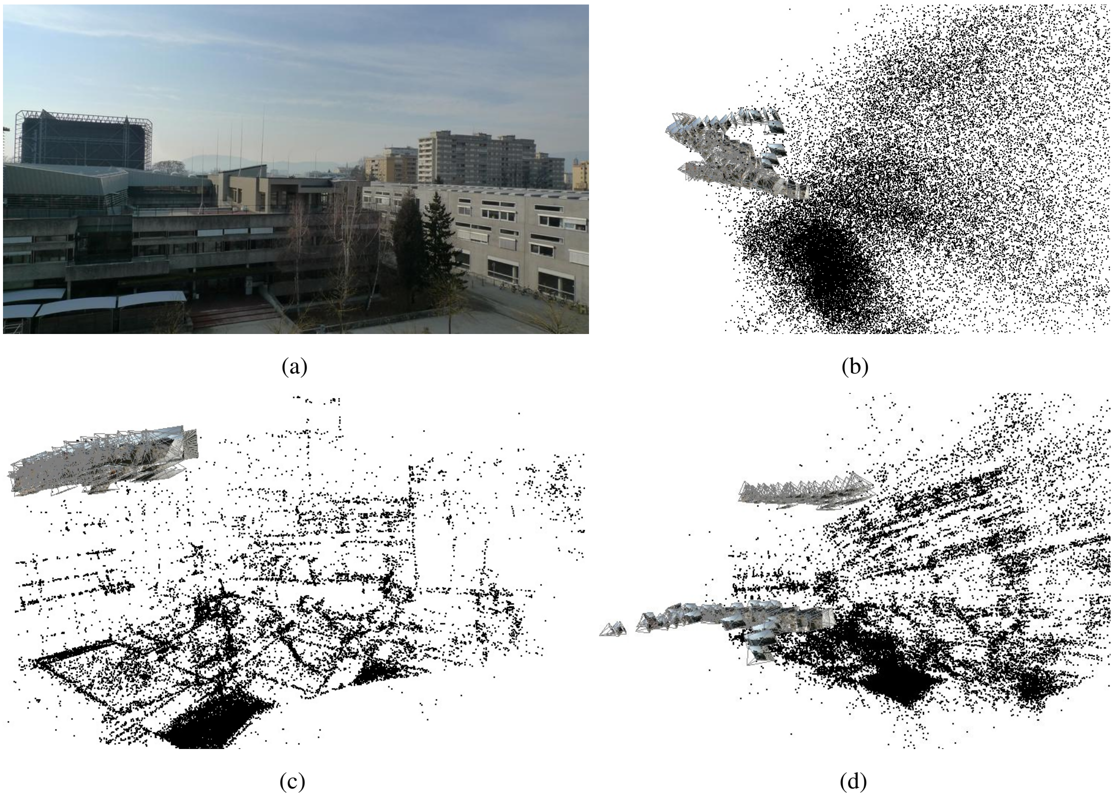


Figure 2.18: (a) Sample image of the scene and (b) initial camera orientation and 3D structure before bundle adjustment. Reconstruction result after bundle adjustment using the robust Cauchy (c) and non-robust Squared Error (d) cost function. Note, while in (c) a true geometric configuration is found, the Squared Error cost function leads to a wrong geometric configuration (d).

robust cost functions (e.g. Huber, Blake-Zisserman, Sigma, Cauchy) achieve comparable performance in terms of average and median reprojection error, the Squared Error cost function leads to a wrong reconstruction, i.e. bundle adjustment does not converge to a reasonable geometric solution. This can be seen from Figure 2.18(d).

2.10 Conclusion and Discussion

In this chapter the core components of current state-of-the-art 3D reconstruction pipelines for unordered images were described and discussed in detail. Determining structure from motion from unordered image collections is computationally demanding and requires fast and scalable algorithms. We provide efficient and robust algorithms for matching and geometric estimation. In particular we introduce the concept of effective inliers that considers the distribution of measurements. Several components of our proposed reconstruction pipeline utilize the massive processing power of current graphic processing units. The induced speed up of the GPU is about $10 - 20\times$ compared to single core CPU implementations. The employed bundle adjustment takes advantage of the second order sparsity which saves memory and considerably speeds up processing time. This allows global bundle adjustment for large aerial camera networks which otherwise would be impossible to achieve due to memory restrictions. We empirically compared the performance of different cost functions for bundle adjustment. From our study we conclude that bundle adjustment is still able to converge from weakly initialized camera orientations and 3D points and robust cost functions are able to handle a quite large amount of outliers.

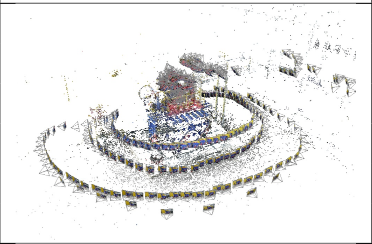
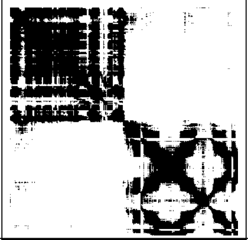
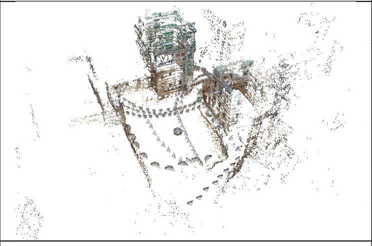
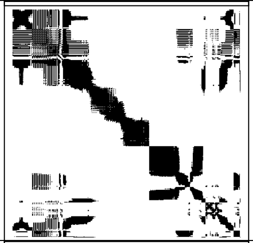
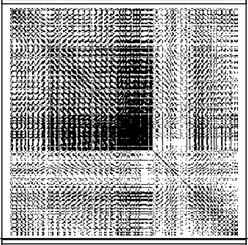
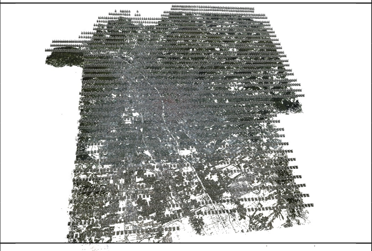
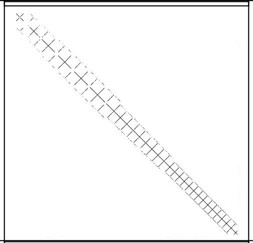
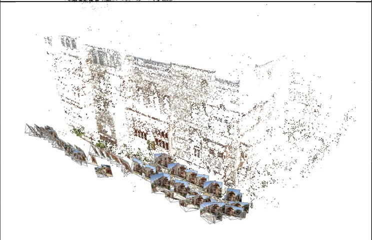
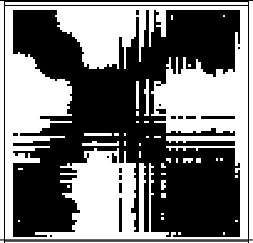
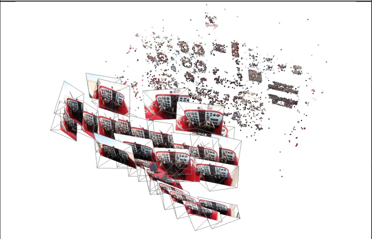
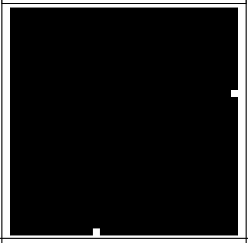
ID	Model	Connectivity	Statistics	
M1			# cams	386
			# points	114609
			# measurements	591596
			avg. # points/img	1532
			avg. # rays/point	5.16
			connectivity	0.33
M2			# cams	397
			# points	128743
			# measurements	827859
			avg. # points/img	2085
			avg. # rays/point	6.4
			connectivity	0.19
M3			# cams	310
			# points	71330
			# measurements	338582
			avg. # points/img	1092
			avg. # rays/point	4.74
			connectivity	0.33
M4			# cams	2962
			# points	3775128
			# measurements	12690926
			avg. # points/img	4284
			avg. # rays/point	3.36
			connectivity	0.008
M5			# cams	92
			# points	25776
			# measurements	177147
			avg. # points/img	1925
			avg. # rays/point	6.87
			connectivity	0.24
M6			# cams	33
			# points	8167
			# measurements	45223
			avg. # points/img	1370
			avg. # rays/point	5.54
			connectivity	0.99

Table 2.4: Sparse reconstructions of different scenes and corresponding properties. *Connectivity* represents the adjacency matrix of the epipolar graph, dark pixels represent valid two view relations.

Model	Cost function	ϵ_{rot} $[^\circ]$	ϵ_{trans} $[\%]$	ϵ_{rep} $[pixel]$	ϵ_{inlier} $[pixel]$	true inlier $[\%]$
M1	Before BA	3.98176	1.61899	448.215	340.553	0.000261
	Squared	**fail**	**fail**	**fail**	**fail**	**fail**
	Blake-Ziss.	0.22731	0.113114	141.573	14.0155	0.9594
	Huber	1.02767	0.520316	142.919	28.5056	0.475454
	Sigma	0.309049	0.200757	138.349	10.3251	0.967733
	Cauchy	0.341035	0.209335	137.365	9.34907	0.968433
M2	Before BA	3.97133	1.75533	627.391	439.453	0.0001195
	Squared	**fail**	**fail**	**fail**	**fail**	**fail**
	Blake-Ziss.	0.358019	0.726515	235.025	14.9726	0.924114
	Huber	**fail**	**fail**	**fail**	**fail**	**fail**
	Sigma	0.398808	0.470354	228.939	9.27123	0.94713
	Cauchy	0.576728	0.794316	228.715	9.99543	0.930977
M3	Before BA	3.98675	1.66345	565.324	375.56	0.000265
	Squared	**fail**	**fail**	**fail**	**fail**	**fail**
	Blake-Ziss.	0.0307046	0.322131	217.68	9.17144	0.988474
	Huber	0.432659	1.06562	215.559	26.3731	0.533253
	Sigma	0.0368807	0.512585	218.083	9.69955	0.987504
	Cauchy	0.0121001	0.492083	216.688	8.41596	0.989199
M4	Before BA	3.98408	1.69158	439.354	346.248	0.000357
	Squared	**fail**	**fail**	**fail**	**fail**	**fail**
	Blake-Ziss.	3.98408	0.169158	439.354	346.248	0.000357
	Huber	**fail**	**fail**	**fail**	**fail**	**fail**
	Sigma	**fail**	**fail**	**fail**	**fail**	**fail**
	Cauchy	0.833278	0.477546	118.458	6.83417	0.871176
M5	Before BA	3.92908	2.22885	637.331	442.443	0.000131
	Squared	**fail**	**fail**	**fail**	**fail**	**fail**
	Blake-Ziss.	0.0010081	0.135687	232.413	6.21016	0.992054
	Huber	0.900162	0.460379	237.036	23.7131	0.46551
	Sigma	0.00119874	0.347321	232.77	6.45154	0.992299
	Cauchy	0.00139413	0.231444	232.094	5.95546	0.991521
M6	Before BA	3.88078	1.54324	591.781	418.601	0.000147
	Squared	3.46416	1.99431	313.891	183.68	0.008461
	Blake-Ziss.	0.00123388	0.0654017	202.734	3.55311	0.993329
	Huber	0.227291	0.206478	199.784	15.3892	0.739754
	Sigma	0.00112362	0.0407556	202.621	3.50726	0.993476
	Cauchy	0.00110252	0.0771208	202.57	3.49085	0.992421

Table 2.5: Evaluation of different cost functions for bundle adjustment. Camera rotations are perturbed by Gaussian Noise with standard deviation $\sigma_r = 5^\circ$ and the relative translations with a $\sigma_t = 0.05d_i$ where d_i the mean depth of all scene points visible in camera i. Furthermore, a fraction $f_m = 0.1$ of image measurements is replaced by uniformly distributed outliers and a fraction $f_p = 0.1$ points is re-triangulated using the perturbed cameras.

Cost function	$\epsilon_{avg.}$	ϵ_{median}	inliers [%]
Before bundle	403.56	312.67	0.13
Squared error	11.46	4.72	71.89
Huber	4.015	0.724	98.24
Blake-Zisserman	4.56	0.66	98.20
Sigma	4.51	0.677	98.52
Cauchy	4.592	0.662	98.46

Table 2.6: Evaluation of bundle adjustment with respect to different cost functions for a fixed number of 150 iterations. While robust cost functions (Huber, Blake-Zisserman, Sigma, Cauchy) achieve comparable results in terms of average ($\epsilon_{avg.}$) and median (ϵ_{median}) reprojection errors and detect a comparable fraction of inliers (we denote a measurement as inlier if the reprojection error is below 3 pixel), the squared error cost function does not properly converge and the minimization fails as shown in Figure 2.18(d).

Chapter 3

3D Reconstruction Applications and Evaluation

Image based 3D reconstruction has a large range of application. In this chapter we present different applications that utilize structure from motion for model reconstruction from terrestrial and aerial data and images acquired using Micro Aerial Vehicles (MAVs). Furthermore, our proposed workflow is evaluated on a large range of datasets and a detailed evaluation of the employed methods is given.

3.1 Dense City Reconstruction from User-Contributed Photos

In this section we focus on the uncoordinated generation of digital copies of urban habitats from community supplied terrestrial images. Our proposed approach is designed to work on unorganized but pre-calibrated image datasets. Taking advantage of recent progress in image matching and structure from motion (SfM) we present an end-to-end workflow for image based scene reconstruction. Our idea is to apply the famous and effective Wiki-principle, well known from textual knowledge databases (e.g. Wikipedia), to the objective of creating photorealistic 3D city models. As input we rely on images from low cost digital consumer cameras taken by multiple users. Being integrated in most of todays mobile phones, digital cameras are nowadays available at any time and everywhere. Furthermore, photogrammetric evaluations [Gruen and Akca, 2007] have also shown that mobile phone cameras provide a sufficiently high accuracy for many photogrammetric tasks. We expect that these kind of devices can be used for detailed and accurate city modeling. Recent advances in wide baseline image matching and structure from motion made it possible to even reconstruct a scene from diverse and uncontrolled photo collections taken by different people under varying weather and illumination conditions. Such a system was presented in [Snavely et al., 2006], demonstrating fully automatic 3D reconstruction from community photo collections downloaded from photo-sharing websites (e.g. www.flickr.com). Goesele et al. [Goesele et al., 2007] further demonstrated the applicability of Multi-view stereo (MVS) techniques on such inhomogeneous and diverse datasets. Community photo collections normally comprise millions of

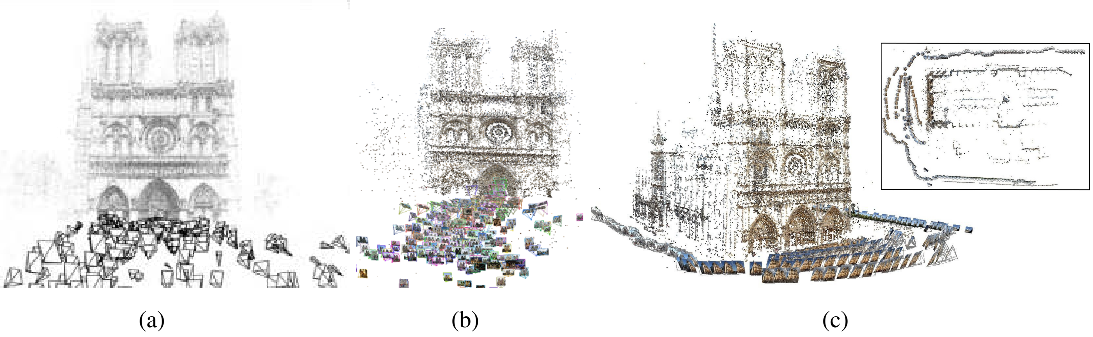


Figure 3.1: Notre Dame reconstruction result from Internet photo collections (a) Snavely et al. [Snavely et al., 2006] and (b) Li et al. [Li et al., 2008]. (c) 3D model obtained by a structured, Wiki-based image acquisition strategy.

images of famous and important landmarks. However, it turns out that humans have a tendency towards capturing a landmark from just a few prominent viewpoints. These locations comprise a huge image density, whereas photos from ordinary streets or even whole cities might be entirely missing. Therefore the resulting models are incomplete, as only popular viewpoints of landmarks are well-represented. In contrast, a Wiki-based reconstruction approach implies a more structured image acquisition strategy, since photos are intentionally captured for the purpose of 3D modeling. Therefore, larger and more complete 3D models are obtained. Figure 3.1(a) and 3.1(b) shows reconstructions of Notre Dame Cathedral computed from community photo collections using the methods of [Snavely et al., 2006] and [Li et al., 2008], respectively. Even though there exists thousands of images from Notre Dame on the web, from those images only the front facade of the landmark can be reconstructed [Raguram et al., 2011] (due to the lack of images from other perspectives). In contrast, a Wiki-based, structured image acquisition strategy leads to more complete reconstruction results as depicted in Figure 3.1(c).

We expect that a user contributed system will result in a rapid creation of virtual copies of urban environments. In the first instance, we aim on textured dense models in quality similar to the results presented in [Pollefeys et al., 2004]. These raw models need to be post-processed in subsequent steps to allow an efficient visualization. By adding more and more images, the reconstructed models can be incrementally maintained and refined gradually. The final city models can then be used for many applications ranging from tourism and cultural heritage over city planning to emergency support.

Reconstruction systems based on still images (e.g. [Brown and Lowe, 2005, Kamberov et al., 2006, Martinec and Pajdla, 2007]) are generally designed to operate in batch mode. Photo Tourism [Snavely et al., 2008a] and the related PhotoSynth Web-interface[1]) is probably the most-well known application for automatic structure from motion computation from a large set of unordered images. A collection of supplied images is analyzed and correspondences are established, from which a relevant subset of views and the respective 3D structure are determined. Photo Tourism

[1] *http://labs.live.com/photosynth*

does not explicitly incorporate calibrated cameras, but relies partially on the focal length specification found in the image meta-data to obtain the initial metric structure. Images with incorrect or missing meta-data can be registered by pose estimation.

The majority of 3D modeling approaches is intended for a decentralized use on personal computers. [Vergauwen and Van Gool, 2006] presented the first Web-based interface[1] to their 3D modeling engine, working with uncalibrated cameras [Pollefeys et al., 2004]. Autodesk provide the recently announced 123D Catch web-service[2] that used cloud computing and allows to automatically create dense 3D models of a scene from unordered images. Registered users can upload their images and subsequently receive the resulting textured mesh models. Their proposed system is targeted at reconstructing individual sites from a limited number of images, but is not aimed on building and maintaining a global image and 3D model database like in our approach.

One of the main difference which distinguishes our reconstruction pipeline from current state-of-the-art systems is the incremental reconstruction approach. In our system, the 3D models can evolve over time and are all stored in a global repository. Instead of using publicly available photo collections as done in [Snavely et al., 2006, Li et al., 2008] and [Goesele et al., 2007, Frahm et al., 2010], we rely on calibrated images submitted by interested users. In general it is not necessary to use calibrated images to get a metric (up to scale) reconstruction, since self-calibration methods (e.g. [Triggs, 1998]) exist. However, our own experience with these techniques indicates that the accuracy and stability of structure-from-motion computation is higher in a pre-calibrated setup.

To ease the calibration effort for the end-user, we employ a procedure aiming for the accuracy of target calibration techniques without the need for a precise calibration pattern. The approach is based on simple printed markers imaged in several views. The use of specific markers enables to establish robust and correct correspondences between the views.

3.1.1 Processing Pipeline

Our structure from motion pipeline follows an incremental 3D reconstruction approach as described in Section 2.8.2. Each input image is resampled according to the lens distortion obtained from the calibration procedure. Since the internal camera parameters are known, an Euclidean SfM algorithm is employed to compute the camera orientations and the sparse point reconstruction of individual scenes. Overall, our SfM algorithm consists of three major processing steps. Firstly, salient features are extracted in each frame. We rely on SIFT features [Lowe, 2004] because of their success reported in the vision community. Secondly, we compute feature correspondences for all images. Since exhaustive pair-wise image matching on large databases is prohibitively expensive, we use a vocabulary tree data structure and inverted files for coarse matching. Each image is taken as query and for the first fraction of reported candidate images a more discriminative pair-wise image matching is performed. In addition, a geometric consistency check is used to remove mismatches. Finally, an incremental reconstruction algorithm computes the position of each camera and the 3D points associated to the extracted feature points. After camera orientations

[1] *http://www.arc3d.be/*

[2] *http://www.123dapp.com/catch*

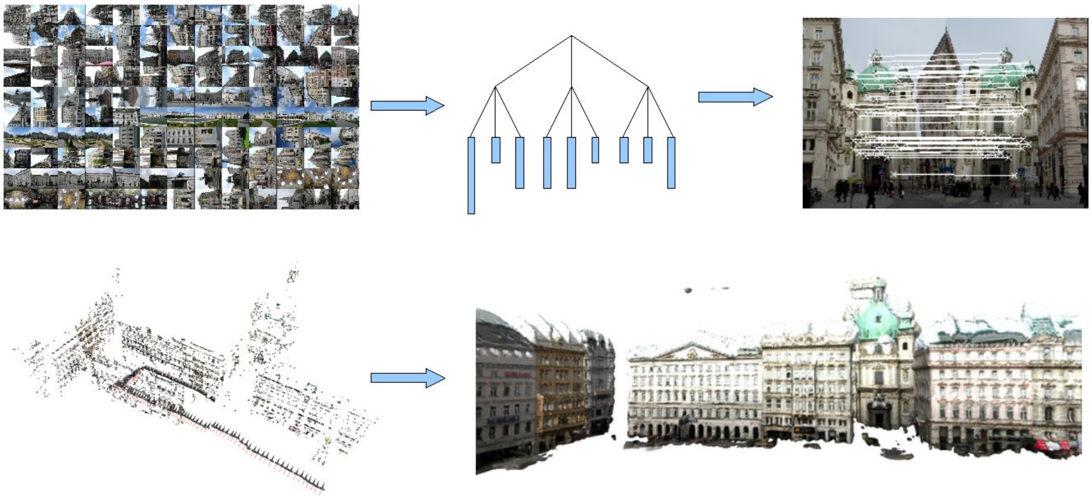


Figure 3.2: Overview of our proposed reconstruction system. From left to right and top to bottom: Unordered image collection, image retrieval by a generic vocabulary tree, discriminative image matching with geometric verification, sparse reconstruction obtained by structure and motion computation and final dense reconstructed scene.

and sparse 3D structure is recovered, batch-processing is used for multi-view dense matching that delivers photo-consistent 3D city models. An overview of our proposed reconstruction pipeline is depicted in Figure 3.2, individual processing steps are summarized in Algorithm 2.

3.1.2 Fast and Flexible Camera Calibration

A variety of approaches exist for accurate camera calibration that are either based on 3D targets [Tsai, 1986, Heikkilä, 2000] or on planar patterns [Triggs, 1998]. In [Zhang, 2000] a robust and flexible camera calibration technique that only requires the camera to observe a planar pattern shown at a few (at least two) different orientations is proposed. Either the camera or the planar pattern can be freely moved, whereas the motion need not to be known. This method has been proven to be very flexible to apply, robust and accurate. One prerequisite of Zahng's approach is that the entire calibration target is visible in each image and the target is in focus. These conditions are often hard to achieve, especially since 3D reconstruction of outdoor environments generally requires an infinite focus due to the usually large distances to the pictured object. The calibration pattern thus needs to be sufficiently large.

We propose a new calibration technique that overcomes these problems and further eases the calibration effort for users. Our calibration method is based on simple printed markers that are pictured from several views. The use of specific markers enables to establish robust and correct correspondences between the views. Marker patterns are printed on several sheets of paper and are typically arranged on the floor (see Figure 3.3). These print-outs can be laid out arbitrarily, hence the well-known calibration method of Zhang [Zhang, 2000] is therefore not applicable. Note, in contrast to calibration methods that rely on traditional planar checkerboard patterns, our approach does not necessarily require visibility of markers in every captured image.

Marker Detection

The first step in the calibration procedure is the detection of the circular markers in the images and the extraction of the unique marker ID (see Figure 3.3(a)). Ellipses are extracted via the Canny edge detection and a subsequent grouping comprises the set of putative markers. The ellipses are further checked for the occurrence of a valid binary circular pattern (after rectification of the local image patch using the ellipse parameters). The centers of the extracted ellipses are only approximations of the true marker centers, hence we utilize an additional central checkerboard pattern for accurate marker localization. The 2D marker position is refined using a nonlinear optimization procedure to align a synthetic checkerboard pattern to the rectified image patch. Matching feature points across multiple views is trivial, since unique and easily extractable IDs are available. Of course, the uniqueness of extracted markers in every image needs to be checked to avoid incorrect detections in case of blurred or otherwise low-quality images.

Solving Camera Calibration

Since the marker images are laid out on a planar surface, corresponding feature points are related by a homography. Hence, the first estimation of lens distortion parameters attempts to minimize the reprojection error between extracted feature points with free homography and lens distortion parameters [Pajdla et al., 1997]. More formally, if x_i^k denotes the position of marker k in the i-th image, the initial distortion estimation determines

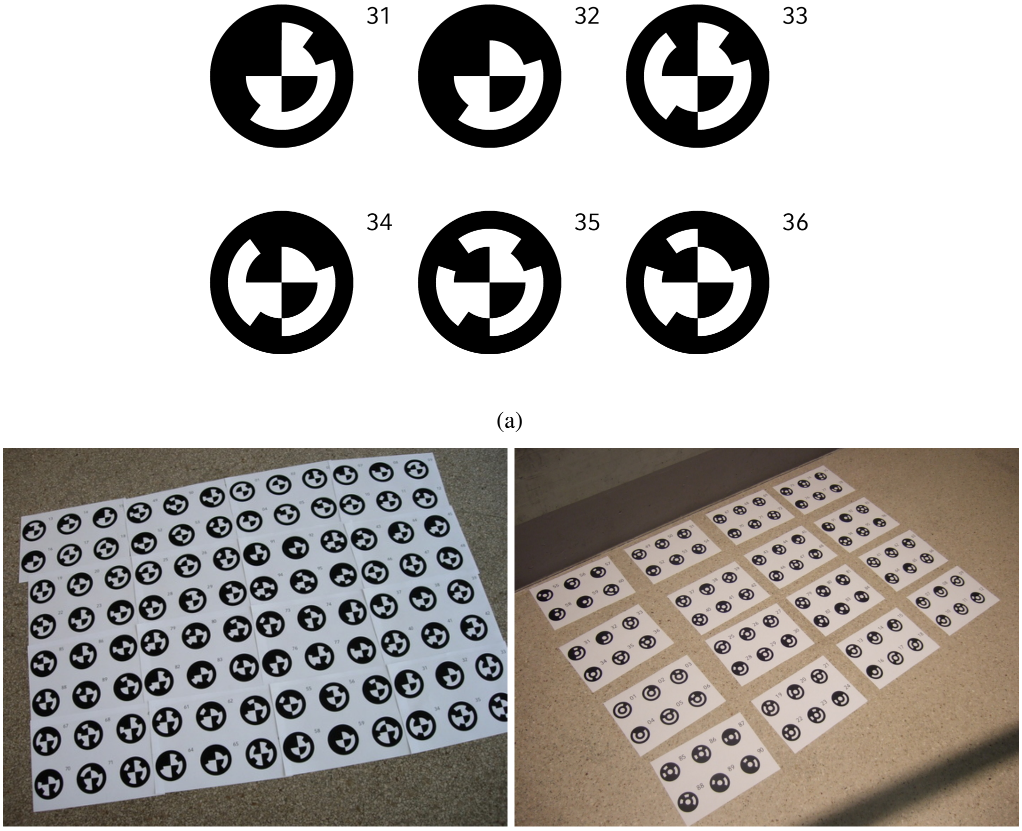
$$\arg\min_{H_{ij},\theta} \sum_{i,j} |\tilde{D}(x_j^k, \theta) - \tilde{D}(H_{ij} x_i^k, \theta)|^2, \tag{3.1}$$

where H_{ij} denotes the image homography from view i to j and $\tilde{D}(x, \theta)$ is the *inverse* distortion function with coefficients θ. The distortion model is

$$\tilde{D}(x, \theta) = (x - (\tilde{u}_0, \tilde{v}_0)^T) \cdot (1 + k_1 r^2 + k_2 r^4), \tag{3.2}$$

with $r = \|x - (\tilde{u}_0, \tilde{v}_0)^\top\|$. θ is the vector $(\tilde{u}_0, \tilde{v}_0, k_1, k_2)$ consisting of the distortion center $(\tilde{u}_0, \tilde{v}_0)$ and the coefficients k_1 and k_2, as described in Section 2.1.1.

The center of radial distortion $(\tilde{u}_0, \tilde{v}_0)^\top$ is independent from the optical principal point (u_0, v_0), thus essentially removing the need for decentering distortion parameters [Tsai, 1987]. The initial homographies are set to the gold standard results [Hartley and Zisserman, 2000] and the distortion parameters are initialized with the image center, and 0 for the coefficients k_1 and k_2, respectively. The non-linear minimization is performed with a (sparse) Levenberg-Marquardt method. Implementation details are described in Section 2.9.1. Note that the homographies are not independent: a consistent set of inter-image homographies should satisfy $H_{ij} = H_{lj}\, H_{il}$ for all l. In our implementation this is enforced by using a minimal parameterization solely based on homographies between adjacent views, $H_{i,i+1}$, and representing $H_{ij} = \prod_{j>l\geq i} H_{l,l+1}$.



(a)

(b)

Figure 3.3: (a) Six calibration markers with central checkboard pattern. Each marker encodes an unique ID. (b) Two typical calibration images showing 96 calibration markers arbitrarily arranged in a 4×4 grid on the floor.

After determining the initial estimate for the lens distortion, the focal length of the camera is estimated from the set of homographies. Both [Triggs, 1998] and [Malis and Cipolla, 2002] employ a non-linear minimization technique for the intrinsic parameter estimation and an initial estimate is required. We utilize a much simpler search technique to quickly determine the camera intrinsics: First, we assume that the principal point is at the image center and that the aspect ratio and skew are one and zero, respectively. Hence, we search for a constant, but unknown focal length f determining the calibration matrix K. If the correct intrinsic matrix K is known, the image-based homographies H_{ij} can be upgraded to homographies between metric image planes, $\tilde{H}_{ij} = K^{-1} H_{ij} K$. For a particular view i assumed with canonical pose, $\tilde{H}_{ij}$ can be decomposed as $\tilde{H}_{ij} = R_{ij} - t_{ij} n_i^\top / d_i$ (via singular value decomposition [Faugeras and Lustman, 1988,Zhang and Hanson, 1996]), where (R_{ij}, t_{ij}) depicts the relative pose and n_i and d_i denote the plane normal and distance (according to the coordinate frame of view i), respectively. Note that each $\tilde{H}_{ij}$ provides its own estimate of $n_i = n_i(\tilde{H}_{ij})$.

For the true calibration matrix K, the extracted normals $n_i(\tilde{H}_{ij})$ should coincide into one common estimate of the plane normal. Hence, a low variance of the set $\{n_i\}$ indicates approximately correct calibration parameters. A slight complication is induced by the fact that decomposing $\tilde{H}_{ij}$ results in two possible relative poses and plane parameters (denoted with n_i^+ and n_i^-). Let (n_0^+, n_0^-) be the most separated pair of normals from all pairs $(n_i^+(\tilde{H}_{ij}), n_i^-(\tilde{H}_{ij}))$. We use n_0^+ and n_0^- as the estimates for the mean of the set $\{n_i\}$. Now, the score for K is the minimum of

$$\sum_{i,j} \min\left(\angle(n_i^+(\tilde{H}_{ij}), n_0^+), \angle(n_i^-(\tilde{H}_{ij}), n_0^+)\right) \tag{3.3}$$

and

$$\sum_{i,j} \min\left(\angle(n_i^+(\tilde{H}_{ij}), n_0^-), \angle(n_i^-(\tilde{H}_{ij}), n_0^-)\right). \tag{3.4}$$

This score is evaluated for potential choices of f, e.g. $f \in [0.3, 3]$ in terms of normalized pixel coordinates. Figure 3.4 depicts the exhaustive evaluated error function (i.e. Equation (3.3),(3.4)) over a range of focal lengths for four different cameras. The objective function comprises many local minima, however a robust global minimum can be found. The value of f with the lowest score is used as initial estimate for the focal length. This procedure is both simple and very fast, and yields sufficiently accurate focal lengths in our experiments.

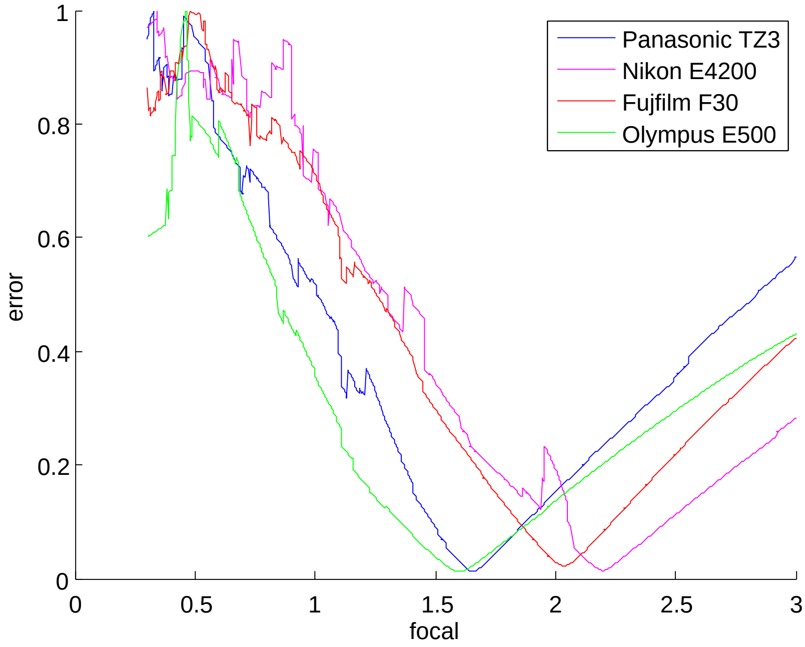


Figure 3.4: Error function with respect to focal length evaluated over the range of $f = [0.3 : 0.0025 : 3]$ for the different cameras. Note, the error function is peaked but smooth near the global minimum.

With the (approximate) knowledge of the focal length, an initial metric reconstruction based on two appropriate views is generated. The remaining views are added by estimating their absolute poses [Haralick et al., 1991]. A final bundle adjustment [Triggs et al., 2000] procedure optimizes the parameters of the forward distortion function, hence the inverse of the originally obtained distortion parameters is required. Since the employed polynomial distortion model is not closed under function inversion, the initial forward distortion parameters are determined by a least squares approach. The final bundle adjustment procedure is applied to refine the camera intrinsics and distortion parameters and to improve the only approximately planar 3D structure and camera poses.

3.1.3 Upgrading the View Network

Our reconstruction system is designed to run online and incrementally. Each time a new image is added to the view network, correspondences to related database images are computed and the structure from motion algorithm is triggered. Since images may be taken from different locations, we do not expect to obtain a single coherent reconstruction, but a forest of multiple reconstructions. We require that a reconstruction consists of at least three images (view triple) and twenty common triangulated points. In general, four different cases can occur if a new image is processed:

1. **Pose Resectioning:** The view can be robustly registered with exactly one given reconstruction. The position of the current view can be computed immediately by robust absolute pose estimation [Fischler and Bolles, 1981, Haralick et al., 1991], since 2D to 3D point correspondences are known. Thereafter, the camera parameters are optimized by iterative refinement and new correspondences are triangulated.

2. **Model Merging:** The image can be aligned with multiple reconstructions. If current image takes part of two or more different reconstructions, the reconstructions are progressively merged. We use RANSAC (Section 2.7.1) to compute a robust 3D to 3D similarity transform for the registration of the corresponding 3D points. According to the computed transformation we insert the smaller reconstruction into the coordinate system of the larger one. The registration is then followed by an Euclidean bundle adjustment as described in Section 2.9. Performing bundle adjustment frequently helps to remove the dependency of the output model on the exact order of the provided images. Of course, since the processing pipeline is incremental, complete order-independency can not be achieved.

3. **Triple Initialization:** The current view cannot be (robustly) aligned with an existing reconstruction, but forms a good view triple with two other views. In this scenario, the neighbors of the current image are estimated from the epipolar graph (Section 2.7.3) and a new reconstruction is initialized from a well-conditioned view triple. The view triple should provide a good triangulation angle and at the same time have many correspondences. Therefore, in the first step we identify the view pair which minimizes,

$$\frac{1}{N}\sum_{i}^{N}\frac{1}{\sin^2(\alpha_i)}\ . \tag{3.5}$$

   Here, N is the number of triangulated points and α_i the angle between the two camera rays which intersect at the 3D point X_i. Thereafter, a third view which minimizes the value in Equation 3.5 with respect to this first configuration is estimated. Note, that $1/\sin(\alpha_i)$ approximates the uncertainty (deviation) of X_i in the depth direction. In [Beder and Steffen, 2006] the view pair with maximal mean roundness (essentially the same as $1/N \sum \sin(\alpha_i)$) is taken. Such an approach does not consider the number of correspondences between two views. In this work, we assume that the precision of further (least-squares) computations depending on the initial structure scales with $1/\sqrt{N}$. Consequently, Equation 3.5 estimates the mean variance of the initial structure for a given view pair.

The relative pose between the first two views is computed by the Five-Point algorithm and the third camera is inserted by the Three-Point [Haralick et al., 1991] algorithm with respect to the triangulated 3D points. Thereafter, bundle adjustment is used to globally optimize the exterior camera orientations and the initial structure. A view triplet is considered as a valid reconstruction if at least twenty 3D points are shared between all views that have a triangulation angle larger than $5°$. This criterion is sufficient to determine robust view triplets for structure initialization.

4. **Postpone Registration:** The geometric relation of this image with any of the known views cannot be established, and structure from motion determination is postponed until a new suitable view is inserted.

Whenever a number of M (15 in our case) views is added to a reconstruction, all cameras and 3D points are optimized by bundle adjustment. Thereafter, for each image measurement the reprojection error is computed, 3D points with an average reprojection error larger than 1.3 pixel and a triangulation angle less than $2°$ are removed. Our experiments suggest that this strategy improves both, accuracy and robustness of the reconstruction algorithm.

3.1.4 Sparse Reconstruction Results and Evaluation

To test our reconstruction approach at large scale, we acquired several thousands photographs from urban environments over a period of three month. Some scenes are fully connected, others are widely separated and cannot be visually linked. In particular, we have a database *Vienna* consisting of 2640 images, and a larger database of 7181 street-side images from *Graz*. In total, four different compact digital cameras were used to generate the databases. The images were captured at different days and under varying illumination conditions. Since the image acquisition was done for the purpose of 3D reconstruction, most of the images have a relative high overlap of about 80%. Some images are sequentially ordered, but the ordering is not considered in the reconstruction pipeline at any time. Overall, the size of the source images varies from two to seven Megapixels. To remove potential compression artifacts, the supplied images are Gaussian filtered and resampled to half resolution for further processing.

Cameras are calibrated with the method described in Section 3.1.2, the obtained intrinsic parameters are summarized in Table 3.1. The calibration precision (i.e. the final mean reprojection error) ranges from 1/20 to 1/5 pixel. To test the stability of the camera intrinsics over time, we repeated the calibration procedure after several month for a camera of type Panasonic TZ3 and

Algorithm 2: Incremental 3D Reconstruction

```
Input: Expanding set of images $I \in \mathcal{I}$ and associate calibration information $K \in \mathcal{K}$
       generic vocabulary tree $\mathcal{V}$ with empty inverted files $\mathcal{F} = \emptyset$
Output: Set of 3D Reconstructions $\mathcal{R}$
        Epiploar Graph $\mathcal{G}$
foreach $I \in \mathcal{I}$ do
    1. Remove Radial Distortion (see Section 3.1.2)
    2. Extract SIFT features (see Section 2.5)
    3. Update inverted files and global vocabulary tree structure $\mathcal{F} = I \cup \mathcal{F}$
    4. Determine potential matching candidates $D \in \mathcal{D}$ from vocabulary tree scoring
    5. foreach $D \in \mathcal{D}$ do
        (a) Detailed Feature Matching (see Section 2.6)
        (b) RANSAC Five-Point (see Section 2.7)
        (c) Determine inlier set and effective number of inliers $m^*$ (see Section 2.7.2)
            if $m^* > t$ then
                $\mathcal{G} = \mathcal{G} \cup e_{ij}$
                insert neighbors of $j$ to $\mathcal{D}$
            end
       end
    6. Upgrade View Network $\mathcal{R}$ (see Section 3.1.3)
end
```

Nikon E4200. The cameras were extensively used during that period, but the calibration parameters remain almost constant. For both cameras the deviation of the focal length and the principal point is within 2% (see Table 3.1 Panasonic TZ3 (1) and (2) and Nikon E4200 (1) and (2)). The variations are small in comparison to the uncertainty of the SfM algorithm and can therefore be neglected in practice.

The image retrieval performance for different database sizes (fractions of the *Graz* database) are depicted in Figure 3.5. On average an image in the *Graz* database has an overlap with about eight other images. We observe that even for the full database size, the first ranked image by the vocabulary tree satisfies the epipolar geometry with a confidence of more than 90%. The results indicate that the vocabulary tree approach generalizes for much larger databases.

In Figure 3.7 the seven largest reconstructions incrementally computed from the *Vienna* database are shown. The images were taken by two users on different days with a Panasonic TZ3 and a

Camera type	# images	resolution	f	(u_0, v_0)	k_1	k_2	err
Panasonic TZ3 (1)	20	3072 × 2304	2564.81	$(1547.2, 1163.7)$	-6.13e-09	1.07e-15	0.124
Panasonic TZ3 (2)	19	3072 × 2304	2564.45	$(1539.4, 1178.8)$	-5.32e-09	7.75e-16	0.148
Nikon E4200 (1)	16	2272 × 1704	2515.98	$(1111.8,\ 838.2)$	-3.05e-08	1.22e-15	0.083
Nikon E4200 (2)	19	2272 × 1704	2525.49	$(1112.3,\ 836.6)$	-3.06e-08	1.34e-15	0.136
Fujifilm F30	25	2848 × 2136	3113.2	$(1385, 7\ 926.8)$	-6.06e-09	3.27e-16	0.23
Olympus E-500	12	3200 × 2400	2593.48	$(1595.7, 1171.3)$	-1.99e-08	1.98e-15	0.05

Table 3.1: Typical calibration parameters of four different cameras. # images denotes the number of used calibration images, f the focal length, (u_0, v_0) the principal point and k_1, k_2 the first and second radial distortion parameters. The last column shows the calibration precision in terms of the final mean reprojection error (err). All values are given in pixels.

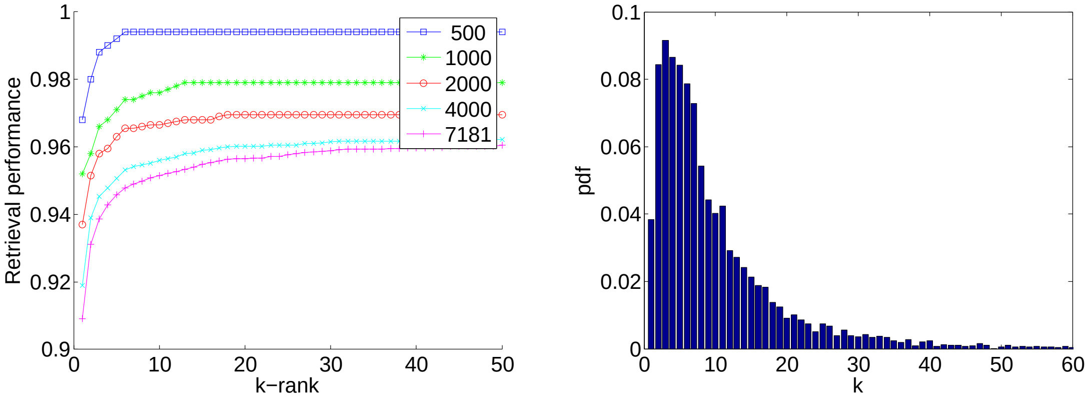
Nikon E4200 camera. Each reconstruction comprises more than 100 images. The largest reconstruction shows the *Graben* street with the St. Stephen's Cathedral with 1330 registered images (see Figure 3.7(h)).

Adding an image to the view network (of size 1000) takes on average $21s$ on the CPU and about $3.5s$ on the GPU. Most of the time is spend on pair-wise image matching and bundle adjustment. Table 3.2 gives typical processing times of the modules involved in our system and compares timings of a single CPU execution with timings achieved with GPU support. Regarding feature extraction and matching, the speedup induced by the GPU is about one order of magnitude.

	CPU [s]	GPU [s]
SIFT (4000 × 3000 pixel)	10	0.4
Coarse Matching	0.5	0.05
Matching (5000 × 5000)	$k \times 1.1$	$k \times 0.044$
RANSAC-H (5-pt, N=2000)	$k \times 0.1$	-
RANSAC-V ($\|C\|$=5000, N=2000)	$k \times 0.12$	$k \times 0.02$
Structure from Motion [h]	1	-
Total Time [h] (615 views, $k = 84$)	21	3.5

Table 3.2: Comparison of processing timings between execution on a single core CPU (Intel Pentium D 3.2Ghz) vs. a GPU accelerated implementation (Nvidia GeForce GTX280). RANSAC-H stands for the hypotheses generation step based on the Five Point algorithm, RANSAC-V for the evaluation module. N is the maximal number of hypothesis, $|C|$ the number of putative correspondences used for evaluation, and k reflects the number of considered images for detailed feature matching and geometric verification.

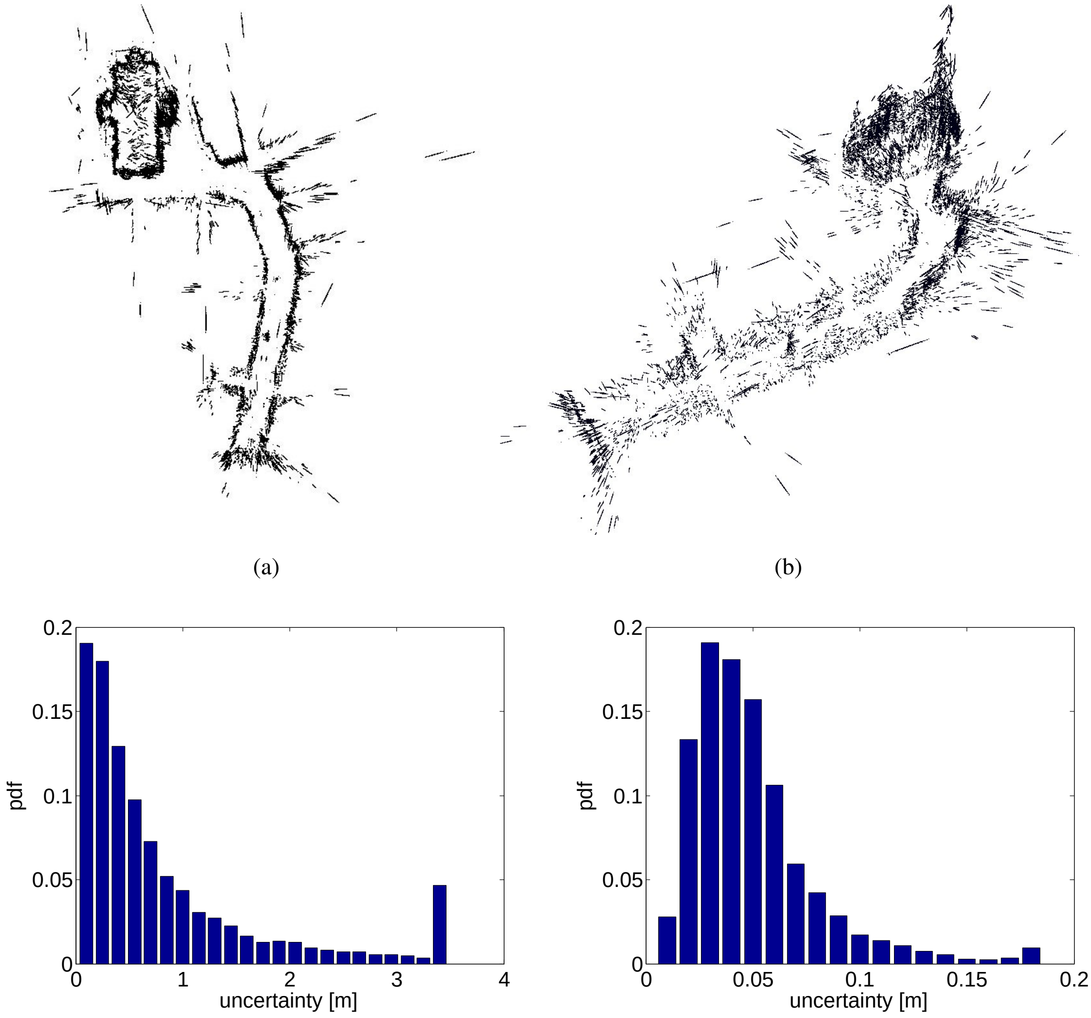
The mean reprojection error of the sparse reconstructions is typically between 0.3 and 0.5 pixels. For the *Graben* reconstruction we estimated the covariances of the 3D points analytically [Beder and Steffen, 2006] and by a Monte Carlo simulation for the purpose of visualization (see Figure 3.6(a) and (b)).



(a) (b)

Figure 3.5: (a) Vocabulary tree performance for image retrieval depending on the database size (fractions of the *Graz* database). The y-axis shows the probability to find an epipolar neighbor in the first k-ranked images reported by the vocabulary tree scoring. (b) Probability density function of the number of verified epipolar neighbors for a query image in the *Graz* database consisting of 7181 street-side images. On average a query images has an overlap with about eight images in the database.

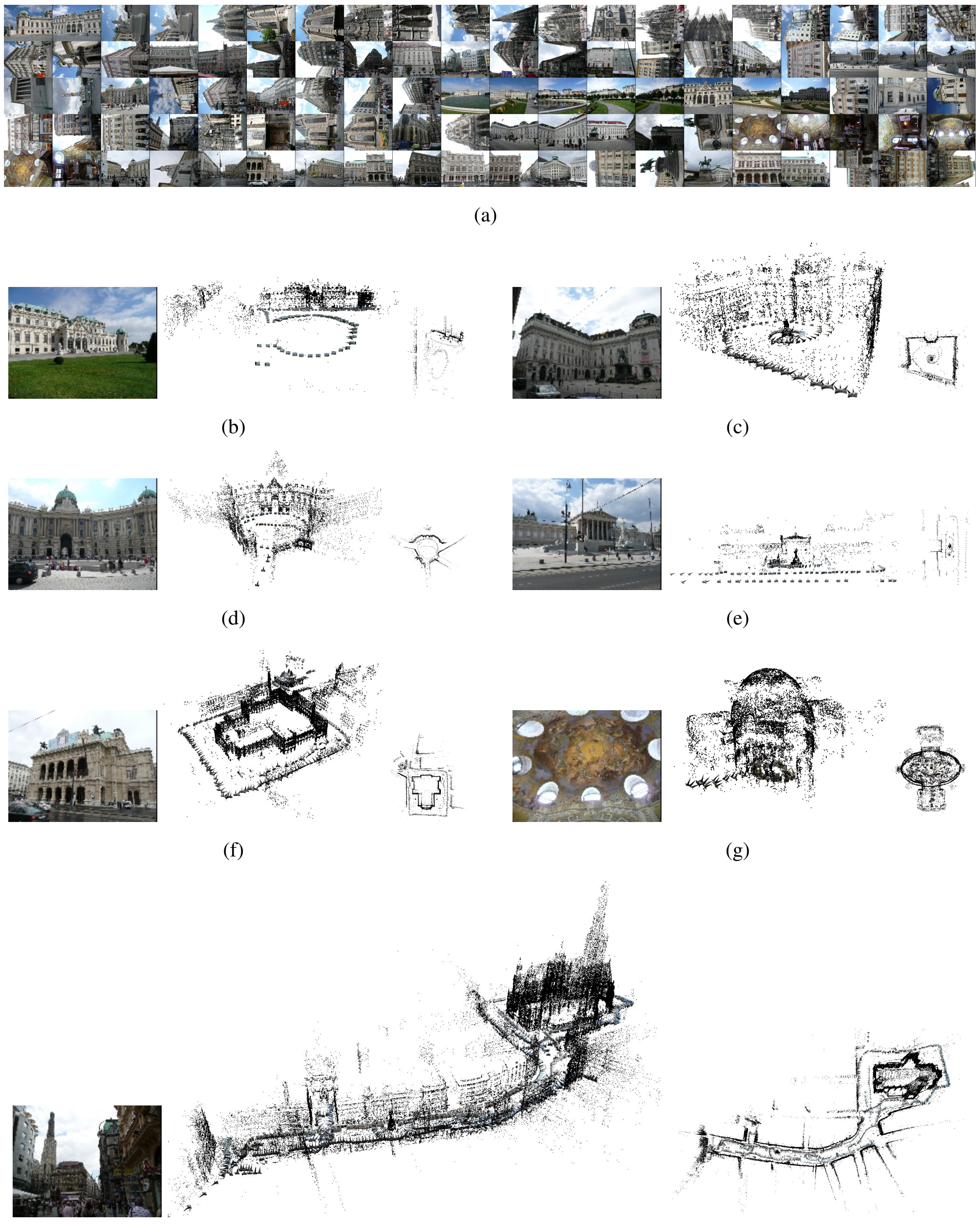
We assume that image measurements are perturbed by Gaussian noise with zero mean and standard deviation one. The uncertainty of the camera projection matrices is neglected. In order to determine a metric scale, the distance between two widely separated image positions ($400m$) enables us to estimate the absolute scale of the reconstruction. In Figure 3.6(c),(d), the probability density function (pdf) of the first two principal covariance components of the reconstructed 3D points is depicted. The first component can be interpreted as the uncertainty in depth, the second and third as the uncertainty for measurements along the image plane. Note, the uncertainty in depth is within $25cm$ for half of the points and within $4cm$ for in-plane measurements (both with confidence interval 95.4%).



(a) (b)

(c) (d)

Figure 3.6: Metric uncertainty analysis of the sparse *Graben* reconstruction (see Figure 3.7(h)). Top view (a) and side view (b) of 3D point covariances obtained from a Monte Carlo simulation. We assume that image measurements are affected by Gaussian noise with zero mean and standard deviation one. (c) Distribution of the 3D point uncertainty in depth (saturated by $3.5m$) and (d) for in-plane measurements (both with confidence interval 95.4%).



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.7: (a) 100 sample images out of 2640, taken in the first district of Vienna. (b)-(h) The seven largest reconstructions obtained by the incremental structure-from-motion algorithm, comprising more than 90% of the database images. (h) Sparse reconstruction of the *Graben* street with the famous St. Stephen's Cathedral consisting of 1330 registered images and 138410 triangulated 3D points.

3.1.5 Dense Multi-View Stereo

Obtaining the initial 3D structure and camera locations is an essential aspect of image-based modeling, but dense geometry is required for a faithful virtual representation of the captured scene. Since we face a large number of images with potentially associated depth maps, we focus on simple but fast dense depth estimation procedures.

Plane-sweep approaches to dense stereo [Yang and Pollefeys, 2003, Cornelis and Van Gool, 2005] enable an efficient, GPU-accelerated procedure to create the depth maps. To obtain reasonable depth values in homogeneous regions, we employ GPU-based scanline optimization in our framework [Zach et al., 2006b]. Note that dense geometry generation is currently performed in an offline fashion for individual large and connected view networks.

For general view networks, a suitable selection of the two matching views for each key view used for depth estimation is necessary. We use the following simple, but effective heuristic to select appropriate matching images, which is based on an estimate for image overlap and viewing directions: for a particular key view and a potential matching view, we determine the correspondences between these views and compute the convex hull of the respective 2D measurements in the key view. The area of this convex hull A_H in relation to the total key image area A gives an estimate of the relevant image overlap. Note that A does not necessarily denote the whole key image size, since insignificant image portions like sky regions can be excluded. Matching view candidates with an overlap A_H/A smaller than a given threshold (typically set to 0.3) are discarded. The image overlap is only a partial guideline for view selection. The angle between corresponding camera rays is another suitable criterion. Very small angles give rise to large depth uncertainties, whereas larger angles are susceptible to image distortion and occlusions. From a practical point of view, triangulation angles of about $\alpha_0 = 6^\circ$ are favorable for dense stereo [Okutomi and Kanade, 1993] (meaning a distance to baseline ratio of about 10:1). Consequently, view pairs with a large overlap and appropriate triangulation angles are preferable. Hence, we rank the views with sufficient overlap according to the following score:

$$\frac{A_H}{A}\, median(\psi(\alpha_i)), \tag{3.6}$$

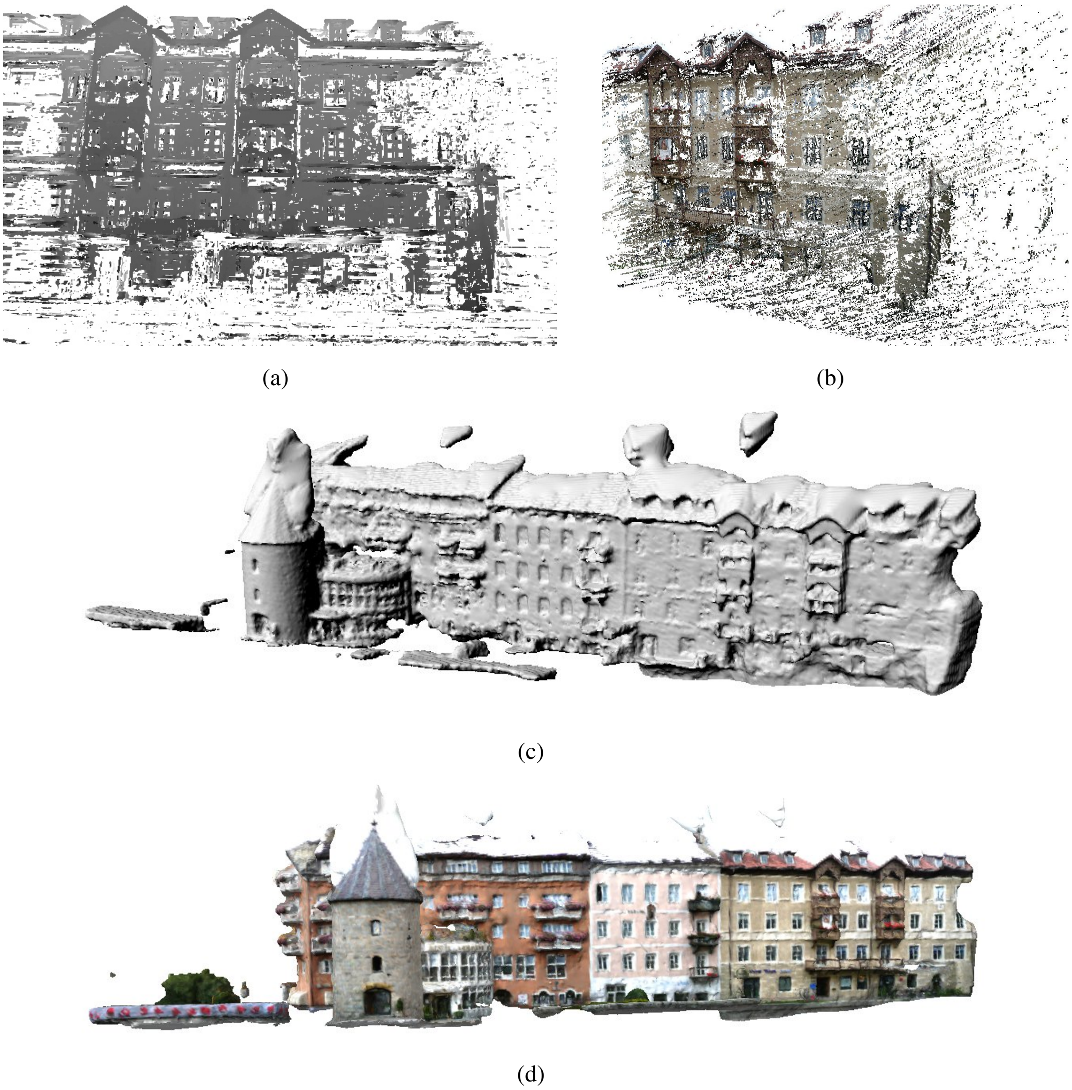
where α_i is the angle between corresponding rays and $\psi(\cdot)$ is a unimodal weighting function with its maximum at α_0. We choose ψ as

$$\psi(\alpha) = \alpha^2\, e^{-2\alpha/\alpha_0}. \tag{3.7}$$

The two views with highest scores are taken as matching views.

Dense depth estimation depends heavily on the quality of the provided epipolar geometry. In order to increase the performance, multi-view stereo is applied on downsampled images (512×384 pixels for $4:3$ format digital images). Matching cost computation and scanline optimization for view triples (one key view and two matching views) takes less than $1s$ on a GeForce 8800GTX. The set of depth maps provides 2.5D geometry for each view. These depth maps are subsequently fused into a common 3D model using a robust depth image integration approach, which is based on

[Zach et al., 2007]. Fusion of range images using volumetric approaches naturally handles surfaces with arbitrary genus, and typically generates watertight meshes [Curless and Levoy, 1996, Wheeler et al., 1998, Kazhdan et al., 2006, Hornung and Kobbelt, 2006]. The input of these volumetric methods consists of (optionally signed) approximations of 3D distance functions induced by the range images, either generated explicitly (e.g. [Curless and Levoy, 1996]) or implicitly using point splatting [Kazhdan et al., 2006, Hornung and Kobbelt, 2006].



(a)

(b)

(c)

(d)

Figure 3.8: (a) Depth map computed using a plane-sweep approach on image triplets and GPU-based scanline optimization [Zach et al., 2006b] and respective colorized point cloud (b). (c) Untextured 3D model using the robust depth image integration approach [Zach et al., 2007] and (d) detailed reconstruction result with texture.

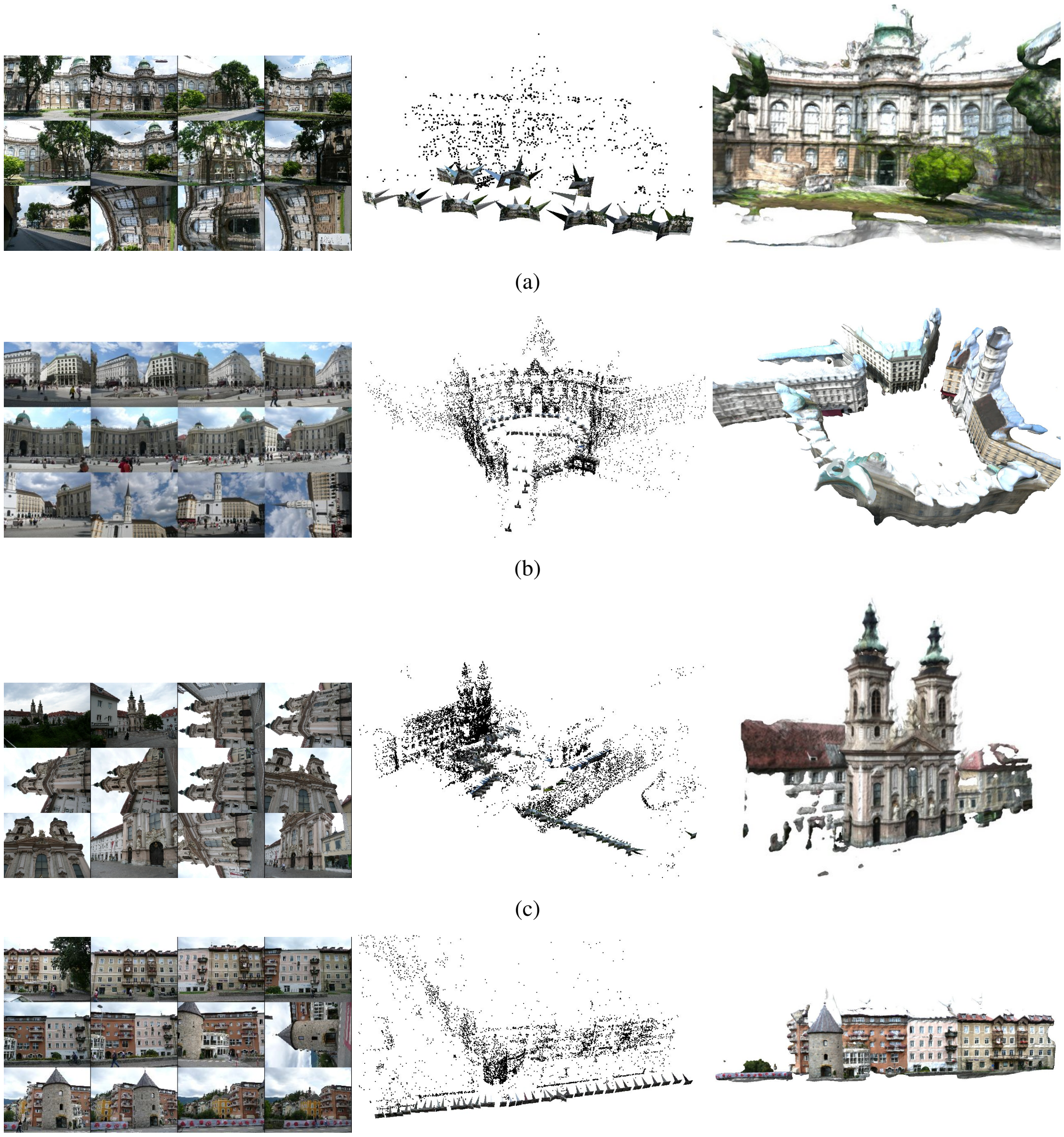
3.1.6 Dense Reconstruction Results

In Figure 3.9, several dense reconstruction results from different scenes are shown. The models are represented by a dense mesh augmented with per-vertex coloring. Post-processing steps are needed to generate textured models which are suitable for efficient visualization. Table 3.3 shows large scale structure from motion results and respective semi-dense models obtained by the Patch-based Multi-view Stereo (PMVS) software [Furukawa and Ponce, 2009]. PMVS is able to reconstruct a scene from camera projection matrices and images by patch based multi-view triangulation. Multi-view reconstruction is implemented as a match, expand and filter procedure that outputs a dense set of patches covering the surface of an object or a scene observed by multiple calibrated photographs. Sparse feature points are extracted and matched across multiple images and from an initial set of 3D patches a procedure expands the current surface. Global visibility constraints are used to filter away erroneous matches. Since our structure from motion pipeline is able to deliver sub-pixel accurate reconstruction results, PMVS can be run at full image resolution.

3.1.7 Conclusion and Discussion

We presented an incremental reconstruction approach for city modeling from user contributed data. We introduced a new and accurate self-calibration method based on coded markers that is both, fast and simple, and can therefore be directly employed by end users. Sub-pixel accurate intrinsic calibration is achieved, on average the reprojection error varies between $0.1 - 0.25$ pixel for standard consumer digital cameras. The reprojection error of the obtained structure from motion models is about 0.5 pixel. In a city environment the average uncertainty in depth is about $25cm$, whereas in-plane measurements have an uncertainty of about $4cm$, respectively. Unlike previous systems that rely on community photo collections downloaded from the web, we employ the Wiki-principle for city modeling. Since photos are intentionally captured for the purpose of 3D modeling, such image collections provide a more complete covering of scene surfaces. This is also observed in the PhotoCity project [Tuite et al., 2011], an online game that trains its players to become experts at taking photos at targeted locations and in great density, for the purposes of creating 3D building models. We conclude that a Wiki-based approach for city modeling has several advantages over 3D reconstruction from Internet photo collections. First of all, a regular and structured image acquisition strategy in general leads to a broader coverage of a landmark. Secondly, a Wiki-based system provides feedback to the user which can decide what images are needed for better scene coverage. Finally, a regular, Wiki-based image acquisition policy enables direct dense modeling techniques, thus view selection optimization [Goesele et al., 2007] for multi view stereo is not necessarily required. Our incremental structure from motion algorithm runs online, hence there is no requirement to see all images in advance. We demonstrate dense 3D reconstruction based on variational methods and using the PMVS software.

We assume that global positioning information will be required in order to generate coherent reconstructions at city scale. Loop detection and closing strategies as well as efficient large scale bundle adjustment are also necessary. Furthermore, the reconstruction of narrow alleys is challenging due to the limited field of view and probably requires special wide angle camera setups.



(a)

(b)

(c)

(d)

Figure 3.9: From left to right: some sample images, sparse reconstructions and final dense models of different scenes. (a) Alte Galerie at Landhausmuseum Joanneum (*Graz*) computed from 49 views. (b) Michaeler square (*Vienna*), 110 images. (c) Mariahilfer church (*Graz*), more than 400 registered photographs in the sparse model, dense reconstruction from a subset of 30 images. (d) Building facades from 54 processed images.

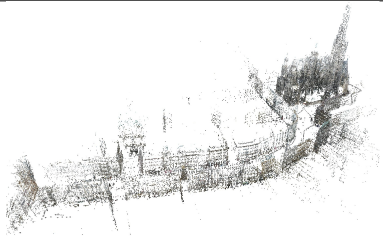
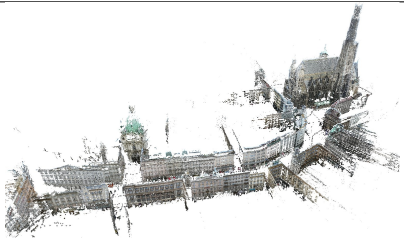
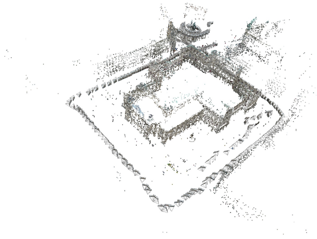
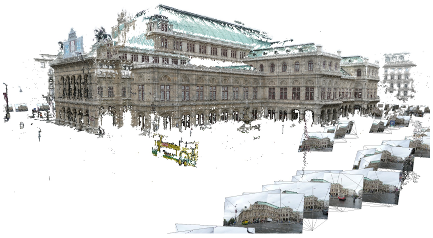
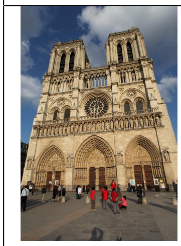
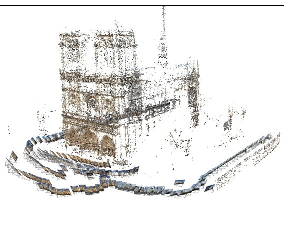
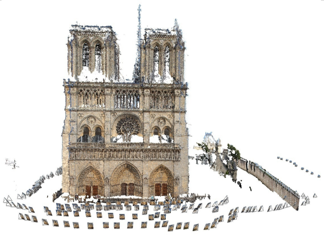
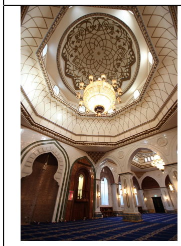
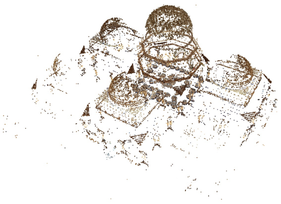
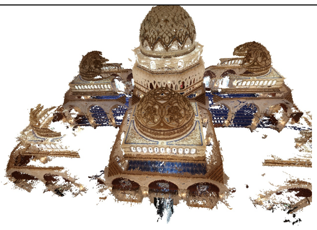
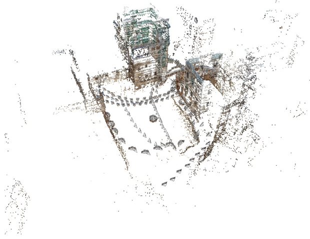
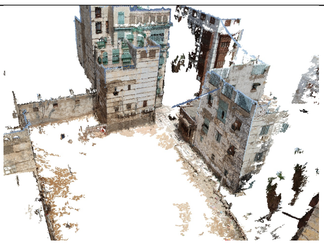
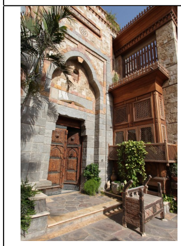
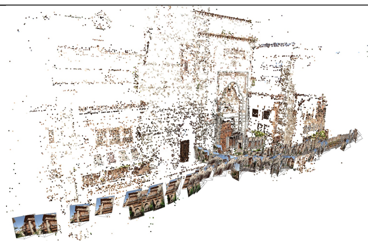
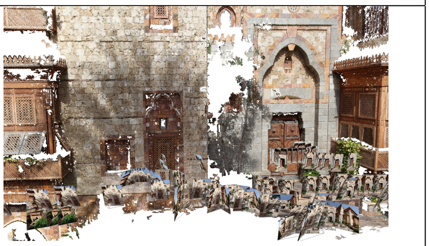
Scene	SfM	PMVS
		
		
		
		
		
		

Table 3.3: Sparse reconstruction results (SfM) of different scenes and semi-dense point cloud (PMVS) computed by the approach of [Furukawa and Ponce, 2009].

3.2 Aerial Images: Redundancy and Dense Reconstruction

Typical airborne photogrammetric surveys are flown using high resolution digital cameras acquiring images with at least 80% forward overlap and 60% side-lap. This image acquisition strategy provides a high degree of redundancy and allows to automatically create detailed maps of the environment [Zebedin, 2010]. Aerial images produced by state of the art large format camera systems such as the products from Vexcel Imaging[1] currently comprise up to 200 megapixels at a high radiometric resolution (e.g. Figure 3.10). The richness of image content information cannot be matched by any other data acquisition device. Today, passive photogrammetry outperforms current LiDAR systems [Baltsavias, 1999] by means of Ground Sampling Distance (GSD) and reduced flight costs [Leberl et al., 2010]. This fact directly leads to significant economic benefits. Aerial flight missions are normally performed using GPS-Aided Inertial Navigation that allows direct georeferencing [Hutton and Mostafa, 2005]. Photogrammetric workflows often use GPS/IMU for the avoidance of Aerial Triangulation (AT) and ground control measurements. However, GPS-Aided Inertial Navigation has several requirements that makes such systems costly and hard to apply in real world [Hutton and Mostafa, 2005]. First of all, the IMU must be rigidly attached to the camera and any misalignment of IMU/camera needs to be calibrated. Second, exact relative timing of image exposure and GPS/INS pose must be provided. Third, the camera interior geometry (focal length, principal point) must be well calibrated and stable. Even if the calibration is done accurately, total reliance on GPS/IMU does compromise the accuracy of resulting stereo matches [Leberl et al., 2010]. Hence, a sub-pixel accurate image alignment is essential for reliable dense matching which can be achieved by automatic extraction and matching of tie points across images and large scale bundle adjustment (see Section 2.9).

In this section we provide a fully automatic framework for aerial triangulation (AT), image overlap estimation and dense depth matching using global optimization techniques. Our algorithms are designed to run on current graphics processing units (GPUs) that makes large scale processing feasible at low cost. To handle large aerial images we introduce is Section 3.2.2 a memory efficient and parallelized SIFT implementation which is a key processing step for fully automated aerial triangulation. Furthermore, in Section 3.2.3 we investigate on the benefits of multi-view image matching compared to pairwise stereo for aerial image networks. Finally, in Section 3.2.4 we present an algorithm for multi view dense matching that is able to produce high-quality depth extraction for phogogrammetric end products like digital surface models DSM and orthophotos. While most of current photogrammetric systems require a manual or semi-automatic selection of point measurements in overlapping images to determine the unknown parameters of the camera, our structure from motion pipeline is fully automated and requires no user intervention at all.

[1] http://www.vexcel.com/

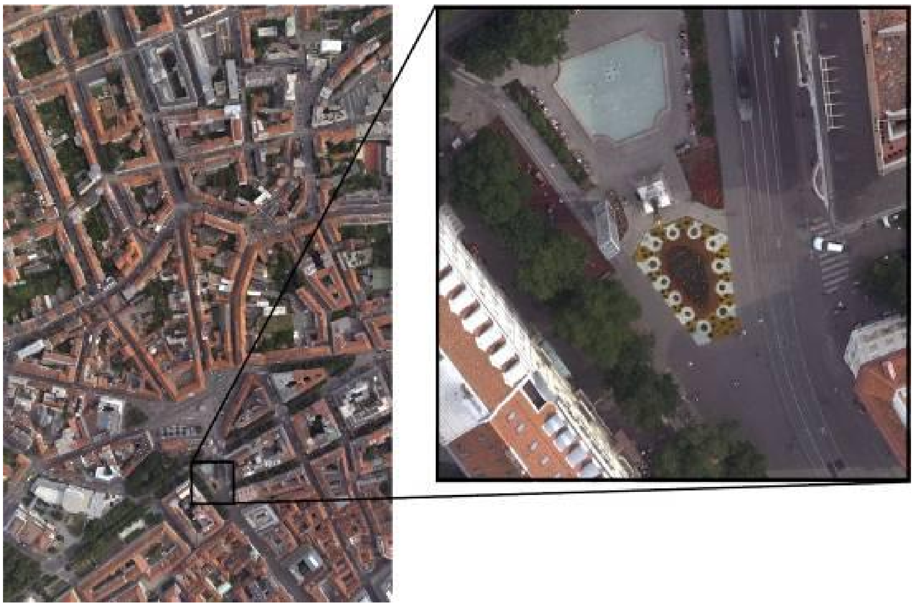


Figure 3.10: High resolution aerial image comprising 11500×7500 pixels at a ground sampling distance (GSD) of 8cm/pixel.

3.2.1 Aerial Triangulation

We employ a fully automated processing pipeline that computes the scene structure and camera orientations from aerial input images, only. First, several thousand Points of Interest (POIs) are extracted from each image using the Scale Invariant Feature Transform (SIFT) [Lowe, 2004]. Next, features between pairs of adjacent images along the flight path are matched. Given the image sequence $\mathcal{I}$ with n images, $\mathcal{I} = \{I_t | t = 1, \ldots, n\}$ the features of each view I_t are matched with a number of adjacent views I_{t+i} with $i = \{-r, \ldots, +r\}$ and $i \neq 0$ where r determines the matching interval. We use $r = 5$ to match aerial images with a forward overlap of 80%. This method achieves tracks along the flight paths but might miss correspondences between flight lines. To establish correspondences between flight lines, an image retrieval approach based on a vocabulary tree search (Section 2.5) is performed. Such an approach assumes that each image is represented as a bag of words [Sivic and Zisserman, 2003] and the employed method efficiently determines a similarity score of all image pairs. In general, overlapping images achieve a higher score than unrelated images, hence this approach is able to detected potential matching candidates across flight lines. We use exhaustive SIFT descriptor matching between pairs of frames as described in Section 2.6. Next, the Five Point relative pose algorithm [Nistér, 2004] inside a RANSAC loop [Fischler and Bolles, 1981] is used to robustly compute pairwise camera orientations (see Section 2.7). The output of the automatic matching procedure is a graph structure denoted as epipolar graph $\mathcal{G}$, that consists of the set of vertices $\mathcal{V} = \{V_1 \ldots V_N\}$ corresponding to the images and a set of edges $\mathcal{E} = \{e_{ij} | i, j \in \mathcal{V}\}$ that are pairwise reconstructions (see Section 2.7.3). The epipolar graph $\mathcal{G}$ encodes relative orientations and pairwise reconstructions. Chaining all relative orientations together should result in a global consistent 3D structure. We follow a greedy, incremental reconstruction approach as described in Section 2.8.2 to iteratively reconstruct the scene from an initial image pair. Structure and camera pose refinement is done using robust bundle adjustment (Section 2.9). Figure 3.11 illustrates an orientation result of 3000 aerial images reconstructed with our fully automated aerial triangulation framework.

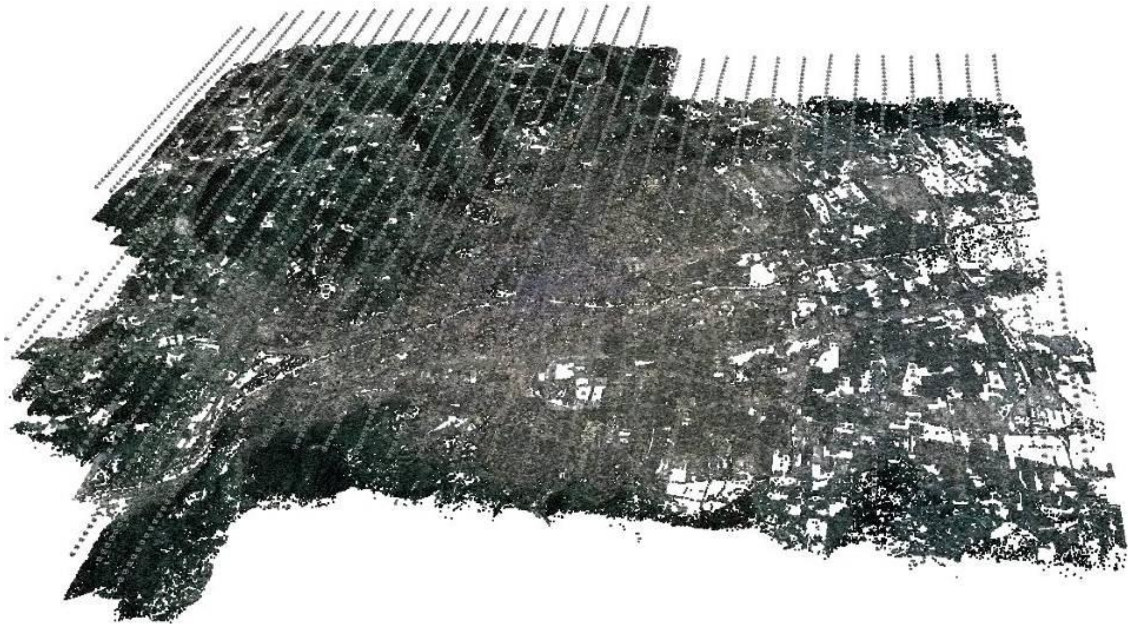


Figure 3.11: Aerial Triangulation (AT) result from 3000 aerial images covering an area of approximately 150km^2 of Graz and surrounding.

3.2.2 SIFT Implementation for Large Images

The large resolution of aerial images imposes special requirements to image processing algorithms. One drawback of a standard SIFT implementation [Lowe, 2004] is the high memory requirement and the expensive processing time. This is especially true for large aerial images, e.g. UltraCamXP images comprising $17,310 \times 11,310$ pixel. Due to data dependencies in the difference of Gaussian cascade filtering, SIFT requires a multiple of memory with respect to the raw pixel input. Let s be the number of scales per octave, SIFT requires $s+1$ gradient images and $s+2$ Difference of Gaussian (DoG) images for each octave. The standard SIFT implementation suggests $s = 3$, therefore requiring $9 \times w \times h$ memory, where w,h is the image width and height, respectively. Furthermore, Lowe suggests to double the size of the input images prior to build the first level of the pyramid which allows the detection of features at highest spatial frequency. This further increases memory by a factor of 4. Hence, SIFT requires 36 times the memory of an original input image.

We propose a tiled SIFT implementation which relaxes memory requirements and takes advantage of multi-core processors at the same time. In our approach, the base image of each octave

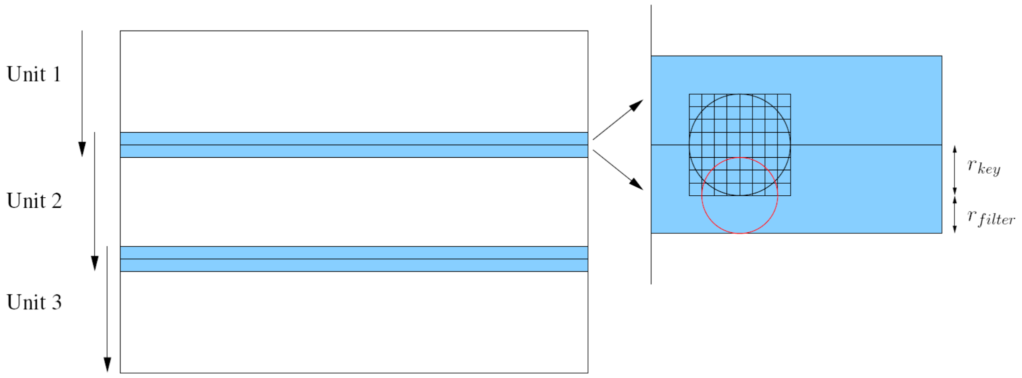


Figure 3.12: Tiled parallelized SIFT extraction. In each octave the base image is subdivided into overlapping tiles. The DoG detection and descriptor computation is performed separately on each data block and in parallel on multi-core processors. The overlap guarantees that no boundary effects occur.

is subdivided into tiles of several rows and feature detection is employed subsequently or parallel on the individual data blocks. An overlap between the individual tiles guarantees that no artifacts in DoG detection and descriptor computation occurs. As a consequence we set the size of the overlapping region to $b = r_{key} + r_{filter}$, where r_{key} is the maximal descriptor radius and r_{filter} the radius of the largest discrete Gaussian kernel used in the cascade filtering. In our standard settings the maximal descriptor radius within an octave is $2 \times 1.5 \times 8$ pixels and the maximal filter radius is 10 pixels, respectively. The detection scheme is depicted in Figure 3.12. On a dual core processor we achieve a speed up of about 1.5 and can process high resolution images with minimal memory requirements at the same time.

3.2.3 Camera Network and Redundancy

In this section we analyze how a multi-view dense matching approach compares to standard stereo matching in terms of achievable depth accuracy for aerial photogrammetry. As shown in [Gallup et al., 2008], the depth uncertainty of a rectified stereo pair can be directly determined from the disparity error,

$$\epsilon_z = \frac{bf}{d} - \frac{bf}{d + \epsilon_d} \approx \frac{z^2}{bf} \cdot \epsilon_d \tag{3.8}$$

where z is the point depth, f the focal length and b the image baseline. Hence, the depth precision is mainly a function of the ray intersection angle. In contrast, for multi-view image matching and triangulation the redundancy not only implies more measurements but additionally constrains the 3D point location through multiple ray intersections. These entities are not independent but are coupled, since they rely on the network geometric configuration that determines image overlap (i.e. redundancy) and baseline, simultaneously. Note, the uncertainty reduces with the square root of the number of intersecting rays while the uncertainty grows quadratically with depth. Given a photogrammetric network of cameras and correspondences with known error distribution, the precision of triangulated points can be determined from the 3D confidence ellipsoid (i.e. covariance matrix $C_{\mathbf{X}}$), as shown in [Beder and Steffen, 2006]. An empirical estimate of the covariance ellipsoid corresponding to multi-view triangulation can be computed by statistical simulation. For the moment we assume that camera orientations and 3D structure are fixed and known. The cameras are distributed along a 2D grid (corresponding to the flight paths) in order to achieve a 80% forward overlap and 60% side-lap (see Figure 3.11). According to a large format digital aerial camera (e.g. UltraCamD) the image resolution is set to 7500×11500 pixel with a field of view $\alpha = 54°$. Furthermore, 3D points are evenly distributed on a 2D plane that corresponds to the bold earth surface, observed from a flying height of 900m. Therefore, an average Ground Sampling Distance (GSD) of 8cm/pixel is achieved.

Given the cameras $P_i \in \mathcal{P}$ (i.e. calibration and poses) and 3D points $X_j \in \mathcal{X}$, respective ground truth projections are produced $x_{ij} = P_i X_j$. Therefore, for every 3D point a set of point-tracks (i.e. 2D measurements) is generated $m = (< x_1, y_1 >, < x_2, y_2 > \ldots, < x_k, y_k >)$. Next, 2D projections are perturbed by zero mean Gaussian isotropic noise $\hat{x} = x + \mathcal{N}(0, \Sigma)$,

$$\Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{3.9}$$

with standard deviation $\sigma_x = \sigma_y = 1$ pixel (i.e. $\sim$ 8cm GSD). Given the set of perturbed point tracks $\hat{m} = (< \hat{x}_1, \hat{y}_1 >, < \hat{x}_2, \hat{y}_2 > \ldots, < \hat{x}_k, \hat{y}_k >)$ and ground truth projection matrices $P_i \in \mathcal{P}$, the 3D position of the respective point in space is determined. This process requires the intersection of at least two known rays in space. Hence, we use a linear triangulation method [Hartley and Zisserman, 2000] to determine the 3D position of point tracks. This method generalizes easily to the intersection of multiple rays providing a least-squares solution. Optionally, a non-linear optimizer based on the Levenberg-Marquardt algorithm (see Section2.9.1) is used to refine the 3D point by minimizing the reprojection error. Through Monte Carlo Simulation on the perturbed measurement vectors $\hat{m}$, we obtain a set of 3D points X_i around a mean position $\hat{X}$. From the Law of Large Numbers it follows that for a large number N of simulations, one can approximate the mean 3D position by,

$$E_N[X_i] = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{3.10}$$

and its respective covariance matrix by,

$$C_X = E_N[(X_i - E_N[X_i])(X_i - E_N[X_i])^\top] \tag{3.11}$$

Using the singular value decomposition the covariance matrix can then be decomposed,

$$C_{\mathbf{X}} = U \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} V^\top \tag{3.12}$$

where U represents the main diagonals of the covariance ellipsoid and σ_i are the respective standard deviations. The decomposition of the covariance matrix (Equation 3.12) into its main diagonals directly relates to the uncertainty in $x - y$ and z direction. Under the assumption of fronto-parallel image acquisition the largest singular value σ_1 corresponds to the uncertainty in depth and σ_2 and σ_3 to the uncertainty in $x - y$ direction, respectively.

We compare the multi-view triangulation result with those of fused pairwise triangulated stereo pairs which can be regarded as the standard approach in aerial photogrammetry (see Figure 3.13). Stereo pairs $(< P_1, P_2 >, < P_2, P_3 > \ldots, < P_{k-1}, P_k >)$ are selected from consecutive views $< P_i, P_{i+1} >$ for $i = 1 \ldots k - 1$ along the flight path. Each pair is used for triangulation of $k - 1$ 3D points $X_{<i,i+1>}$ that belong to one and the same point-track m. The mean of this set is determined representing the fused depth estimate. The covariance ellipsoids corresponding to the uncertainty of one exemplary 3D point fused from a varying number of stereo pairs is depicted in Figure 3.13(d). Note, by using more stereo pairs, the uncertainty in depth decreases but overall the fused stereo result cannot compete with multi-view triangulation/matching (Figure 3.13(c)). For instance, while the uncertainty of 16 fused stereo pairs gives a depth error $\sigma_z \approx 25$cm, the multi-view-triangulation leads to a $\sigma_z \approx 5$cm. For multi-view triangulation we make two observations. On the one hand the overall accuracy along each axis clearly decreases with increasing number of measurements. On the other hand, the roundness $r = \frac{\sigma_3^2}{\sigma_1^2}$ increases by adding more views. While for stereo matching the minimal uncertainty is $\sigma_z = 12cm$ in depth and $\sigma_{x,y} = 5cm$ within the plane, multi-view matching of the aerial dataset leads to a $\sigma_z = 5cm$ and $\sigma_{x,y} = 1.8cm$, respectively.

3.2.4 Globally Optimal Multi-View Dense Matching

For photogrammetric end-products like ortho-image creation or Digital Surface Model (DSM) extraction, dense 3D geometry is required. Our solution to dense depth estimation is based on a multi-view plane sweep approach [Zach et al., 2006a] with global optimization on a 3D voxel space [Pock et al., 2008].

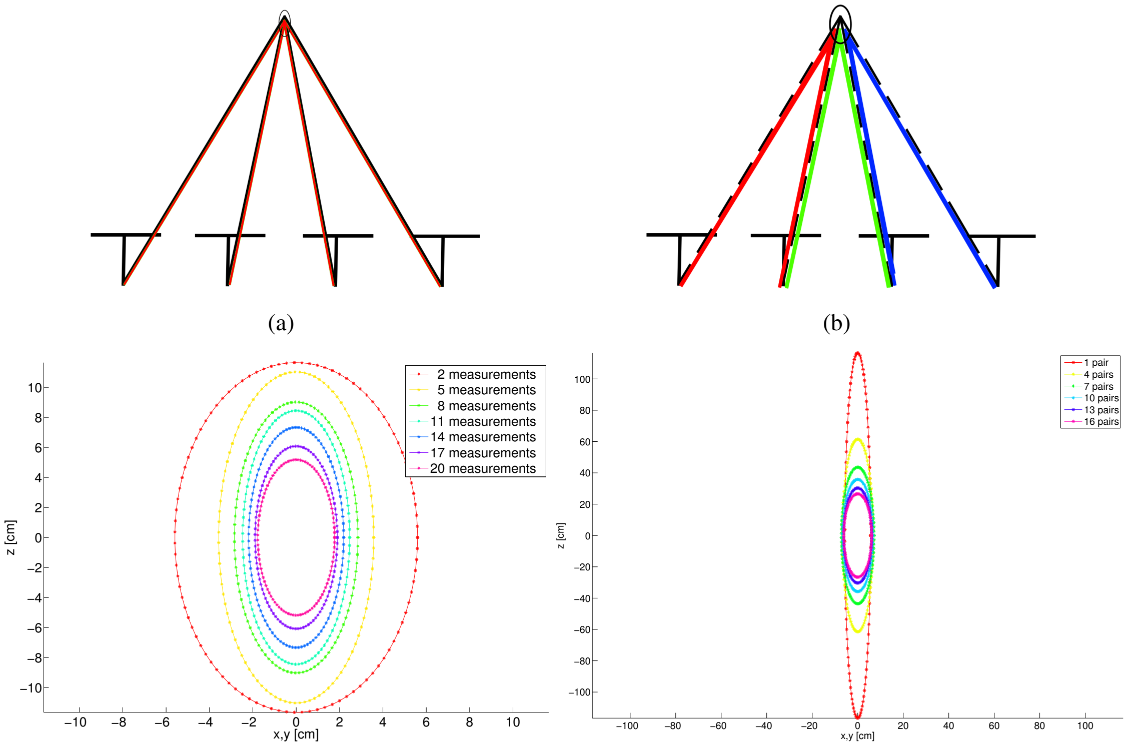
Plane sweep techniques in computer vision are simple and elegant approaches for image based reconstruction with multiple views, since image rectification is not required. The 3D space is iteratively traversed by parallel planes which are usually aligned with a particular key view (Figure 3.16). The plane at a certain depth d from the key view induces homographies for all other views, thus the sensor images are warped to the current plane $\pi = (n^\top, d)$. Here, n is the plane normal and d the current depth hypothesis. The key view is assumed in canonical coordinates $P = K[I \mid 0]$ according to the appropriate homography,

$$H = K'(R - tn^\top/d)K^{-1}, \tag{3.13}$$

which transfers coordinates x from the sensor view to image positions x' of the key view $x' = Hx$. Here, K is the intrinsic matrix of the key view and R, t is the relative pose of the sensor view $P' = K'[R \mid t]$ with respect to the key view. Given two projection matrices $P_1 = K_1[R_1 \mid t_1]$ and $P_2 = K_2[R_2 \mid t_2]$ the relative pose between P_1 and P_2 is computed from,

$$R = R_2 R_1^\top \tag{3.14}$$

$$t = t_2 - R_2 R_1^\top t_1 \tag{3.15}$$



(c)

(d)

Figure 3.13: (a) True multi-view triangulation and corresponding covariance ellipsoid (b) for one 3D point depending on the number of image measurements. Pairwise triangulation (c) and (d) covariance ellipsoid of pairwise 2.5D stereo fusion.

and the normal vector of the plane $n = [0, 0, -1]$. If the plane at a certain depth passes exactly through the surface of the object, the color values from the key view and from the mapped sensor views should coincide at appropriate positions. By sweeping the plane through the 3D space, a cost volume is filled with image correlation values that corresponds to the disparity space image (DSI) in traditional stereo [Seitz et al., 2006].

Initialization

Image-space algorithms usually constrain the maximum disparity range or interval, in which depth values can occur. Respectively, the extent of scene geometry is determined to lie between a near and far plane from the camera center of a key view as depicted in Figure 3.14. Minimal and maximal depth range $[z_{near}, z_{far}]$ can either be estimated from the sparse scene reconstruction from SfM or explicitly set to a global value if prior knowledge about the minimum and maximum scene depth is available, e.g. from a coarse digital surface model. Such a model may be already available through previous aerial mapping surveys, or alternatively, can be generated by combining multiple public domain geographic information sources.

The Shuttle Radar Topography Mission (SRTM) [Farr et al., 2007] provides a digital elevation

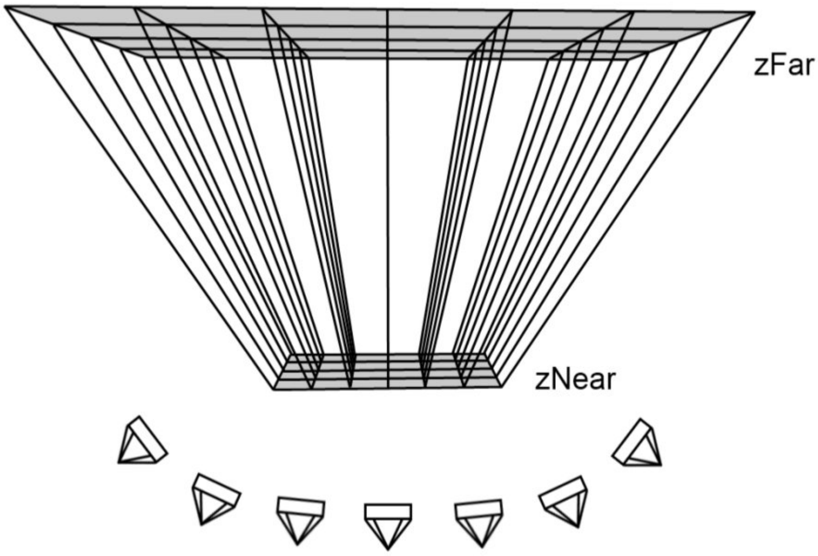


Figure 3.14: Volumentric multi-view dense matching. A near and far plane parallel to the image plane of the reference camera define the bounding volume.

model (DEM) of the Earth at near-global scale, covering about 80% of the Earth's total landmass. The dataset is available to the public in 2.5D raster format at 1 arc-sec resolution (SRTM-1, approximately 30 meters) over the United States and its territories and at 3 arc-seconds resolution (SRTM-3, approximately 90 meters) for the rest of the world. We combine the DEM with information for buildings from freely available 2D vector map data from the OpenStreetMap[1] (OSM) project, which is is a rich source for geographic information. Besides street networks and manifold points of interest (POIs), the OSM project provides outlines of buildings for many cities around the world.

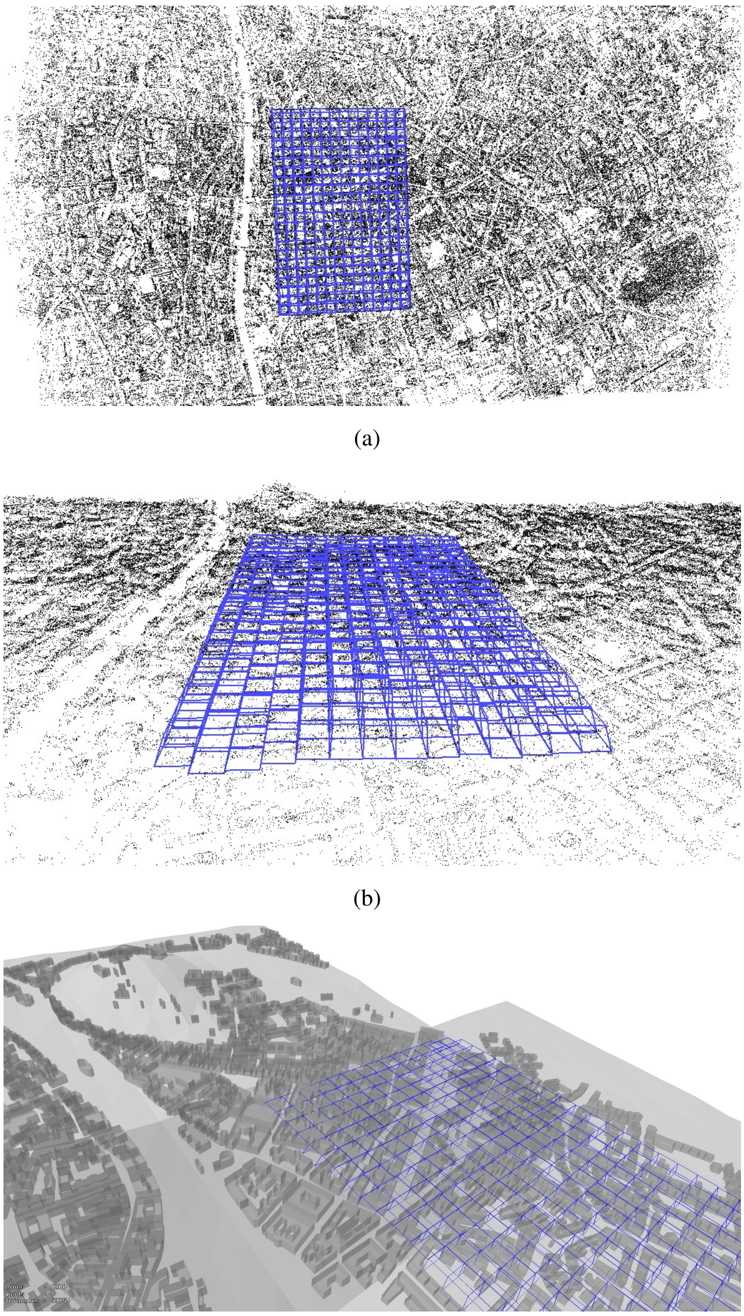
Geometry of the initial DSM is represented as a triangulated irregular network (TIN) [Peucker et al., 1978] of 3D points. Buildings are modeled as polyhedral objects by extruding building footprints to predefined height values for the maximum expected building height (Figure 3.15). This may also allow dense reconstruction algorithms to take advantage of already known scene geometry for applications such as visibility checks and occlusion handling.

In addition to the scene volume extent, a depth sampling Δd and the number of depth steps in the volume is chosen such that sub-pixel accurate matching is achieved. The depth step Δd is adaptively computed such that the Nyquist criterion [Shannon, 1949] $f_s > 2$ pixel is satisfied for at least half of all sensor views. This means that for 50% of potential sensor views i, the following condition must be satisfied,

$$\text{median}\left(||p(P_i, X(d)) - p(P_i, X(d + \Delta d))||\right) < 0.5 \text{ pixel}, \tag{3.16}$$

where $X(d)$ is the point passing trough the center of every tile at depth d, P_i is the projection

[1] http://www.openstreetmap.org



(a)

(b)

(c)

Figure 3.15: Tiling and depth range estimation for one specific key view of the *Graz* dataset from sparse points (a),(b) and by DSM approximation from public domain geographic data sources (c).

matrix of view i and p the projection operator. This ensures that sampling artifacts are avoided for at least 50% of all sensor views.

Image Correlation

We use normalized cross correlation (NCC) as photo consistency measure for plane sweep cost computation. The correlation between two signals (cross-correlation) is a robust approach for dense matching. One advantage of normalized cross correlation (NCC) compared to simpler methods like the sum of absolute differences (SAD) and sum of squared differences (SSD) is the invariance to linear intensity changes which often occur in aerial images. Given two intensity vectors $I_1 \in \mathcal{R}^n$ and $I_2 \in \mathcal{R}^n$, the normalized cross-correlation is computed by,

$$\rho = \frac{\sum_{k=1}^{n}(I_1(k) - \bar{I}_1)(I_2(k) - \bar{I}_2)}{\sqrt{\sum_{k=1}^{n}(I_1(k) - \bar{I}_1)^2 \sum_{k=1}^{n}(I_2(k) - \bar{I}_2)^2}} \tag{3.17}$$

where $\bar{I}_1$, $\bar{I}_2$ are the mean intensities and n is the length of the intensity vector. Note that if two image patches match perfectly, the normalized cross-correlation value is 1. We use an efficient implementation proposed by [Nistér et al., 2004],

$$\begin{aligned}
A &= \sum_{i=1}^{n} I && (3.18)\\
B &= \sum_{i=1}^{n} I^2 && (3.19)\\
C &= \frac{1}{\sqrt{nB - A^2}} && (3.20)\\
D &= < I_1, I_2 > && (3.21)\\
\rho &= (nD - A_1A_2)C_1C_2 && (3.22)
\end{aligned}$$

where the values A, B, C can be precomputed, therefore matching requires only $2n + 5$ multiplications instead of $3n$.

Cost Aggregation and Implicit Occlusion Handling

In order to handle occlusion that often occur in a multi-view setup, we use truncated correlation measures between the key view and the N sensor views,

$$C(x, d) = \frac{1}{N} \sum_{i=1}^{N} \min(d_W(I(x), I_i(x, d)), t) \tag{3.23}$$

where x is the pixel position in the key view I, d the current depth and I_i the respective sensor view. The image similarity function d_W is evaluated in a $k \times k$ neighborhood and t is a constant threshold that accounts for occlusions and outliers.

Since NCC delivers correlation values ρ between $[-1 \ldots 1]$, the image similarity score (i.e. matching costs) is computed using,

$$d_W(I(x), I_i(x, d)) = \frac{1 - \rho}{2}, \tag{3.24}$$

where a perfect correlation value implies zero costs, i.e. $d_W(I(x), I_i(x, d)) = 0$.

Depth Map Extraction

From the 3D cost volume, dense depth maps can be extracted using global optimization methods. Given a graph with node set $\mathcal{V}$, edges $\mathcal{E}$ and a label set $\mathcal{L} \subset \mathcal{Z}$, an optimal labeling $l \in \mathcal{L}^{\mathcal{V}}$ for the energy of the form,

$$\min_{l} \sum_{(u,v)\in\mathcal{E}} P(l(u) - l(v)) + \sum_{v\in\mathcal{V}} D(l(v)) \tag{3.25}$$

where $P(l(u) - l(v))$ are pairwise potentials and $D(l(v))$ is the unary term, respectively. Solving this problem corresponds to a minimal cut on a graph in higher dimensions where labels are ordered. In [Ishikawa, 2003] a minimum cut algorithm is presented that exactly solves this class of Markov Random Field (MRF) problem. This problem perfectly fits to dense depth estimation, where $l(v) \in \mathcal{L}$ are depth labels, $v \in \mathcal{V}$ pixels and $\mathcal{E}$ describes the connection of pixels. Such a labeling combines a certain pairwise regularity term $P(\cdot)$ with an arbitrary data term $D(\cdot)$. In [Pock et al., 2008] a continuous formulation to the discrete multi-label problem of Ishikawa is given. The corresponding variational problem to Equation (3.25) is,

$$\min_{u}\{\int_{\Omega} |\nabla u| + \int_{\Omega} C(x, u(x))dx\}, \tag{3.26}$$

where $u : \Omega \rightarrow \Gamma$ is the unknown function and $\Omega \subseteq \mathcal{R}^2$ is the image domain. $\Gamma = [\gamma_{min}, \gamma_{max}]$ is the range of u. The left term $|\nabla u|$ is the total variation (TV) term that allows for sharp discontinuities in the solution while still being a convex function. This is a desired property for dense matching where edges should be preserved in the solution. The right term of (3.26) is the data term measuring the matching quality for a given u between the key view and sensor views. The spatial continuous formulation comes along with several advantages over the discrete approach. On the one hand continuous optimization can be implemented using simple and efficient primal-dual optimization techniques which can be easily accelerated on parallel architectures such as graphics processing units (GPUs). On the other hand these methods require considerably less memory which makes the method applicable for large practical problems [Pock et al., 2010].

3.2.5 Dense Matching Results

We perform dense matching for a sub-block of the aerial dataset *Graz* as shown in Figure 3.11. For each key view the set of overlapping sensor views is determined. The overlap is computed from sparse correspondences obtained by the aerial triangulation (see Section 3.2.1). Only images with an overlap of more than 10% are considered, which means that each key view has about ten overlapping sensor views. Our dense matching algorithm requires a cost volume of size $W \times H \times D$ which depends on the image width W and height H of each image and the number of depth labels D. Since a global cost volume would be too large to fit into GPU memory, the area of interest has to be divided into tiles (e.g. 512×512). Each tile is processed independently, but with a sufficient overlap in order to suppress boundary effects. Figure 3.15 shows a 512×512 tiling of

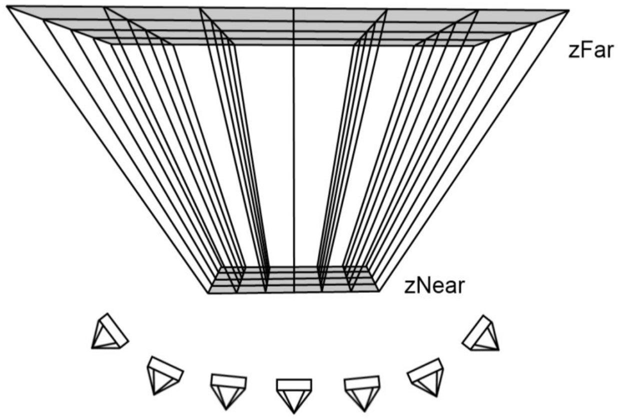


Figure 3.16: Volumentric multi-view dense matching. A near and far plane parallel to the image plane of the reference camera define the bounding volume.

one specific key view. For each tile a minimal and maximal depth range $[z_{near}, z_{far}]$ based on the sparse point cloud is estimated.

For our experiments we set the NCC matching window radius to $r = 1$ pixel and $t = 0.5$ for the outlier and occlusion threshold in the cost accumulation step. The regularization parameter $\lambda = 20$ is used in the continuous optimization method. This parameter balances between data and regularity term and determines the degree of smoothness of the extracted depth maps. Processing of a cost volume of size $512 \times 512 \times 128$ requires about 1.5 minutes on a Nvidia GeForce GTX280. Performance metrics and detailed processing timings for dense matching are summarized in Table 3.4.

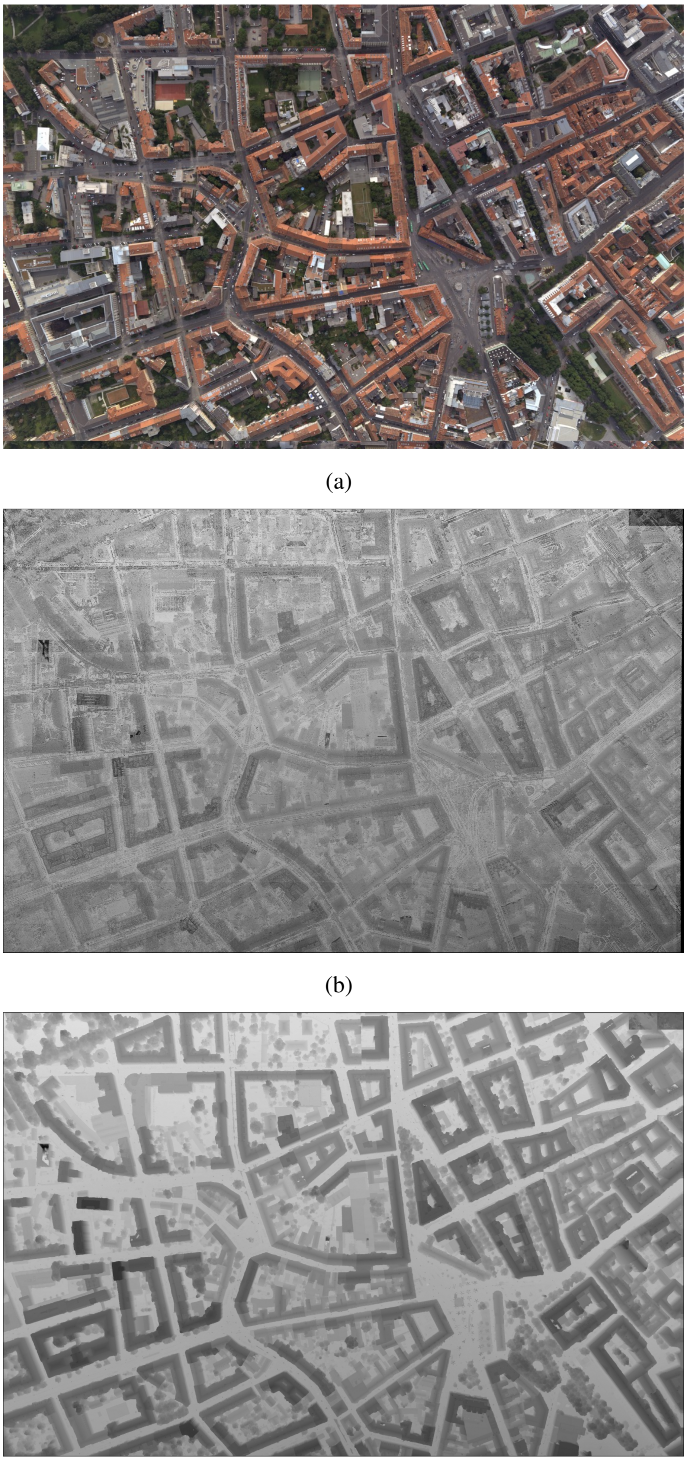
Figure 3.17 shows depth maps computed by a local winner takes all (WTA) approach and the global multi-label optimization. While the WTA approach leads to noisy depth maps due to matching ambiguities, the global method produces clean results while still preserving sharp edges at discontinuities. This can be seen from Figure 3.18 that depicts an oblique view of the textured depth map.

image resolution [pixel]	7500×11500
tile size [pixel]	512×512
number of tiles	384
max number of depths [s]	160
matching time per tile [s]	0.076
global optimization time [s]	74
total time per tile [s]	~ 90

Table 3.4: Performance metrics and timings for processing one high resolution image on a singe GPU (Nvidia GeForce GTX280) into a dense depth map.

3.2.6 Conclusion and Discussion

We presented an approach for fully automated triangulation and dense matching from large aerial images. The method relies on image data only and does not require any external orientation sensor such as GPS/INS. Hence, the proposed method is very flexible to apply. We present an algorithm for efficient and fully automated aerial dense matching using a multi-view approach based on plane-sweep. A global optimization algorithm based on a continuous energy minimization framework delivers globally optimal solutions. We successfully demonstrated that using multi-view matching techniques highly accurate reconstruction results can be obtained. An aerial survey using the UltraCamD at flying height $900m$ with 80% forward overlap and 60% sidelap achieves a reconstruction accuracy of about $5-20$cm in depth and about $2-8$cm for in-plane measurements. From our experiments we conclude that true multi-view matching and triangulation outperforms two-view stereo approaches by about one order of magnitude in terms of achievable geometric accuracy.

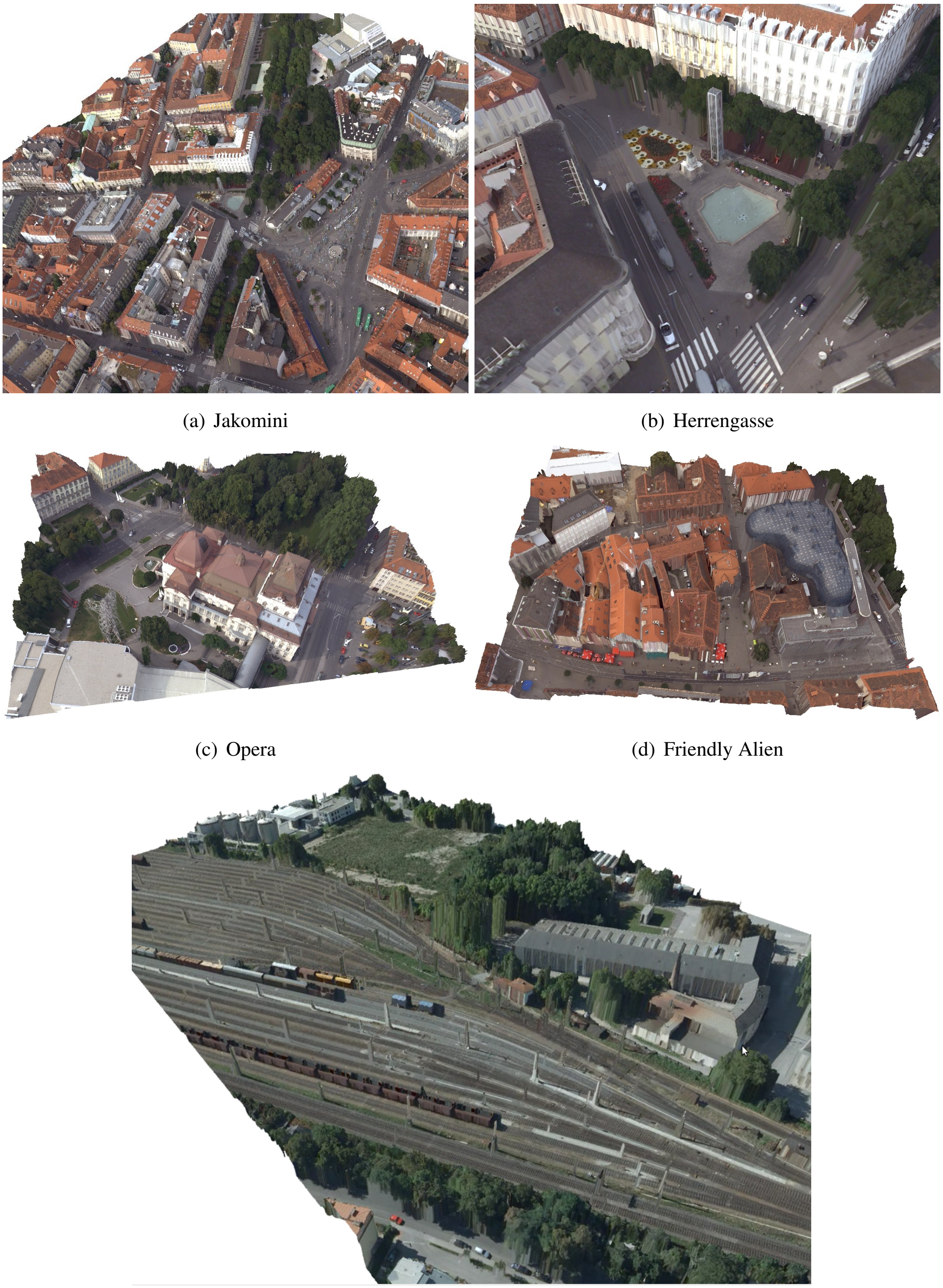


(a)

(b)

(c)

Figure 3.17: (a) Key image and depth maps produced by multi-view dense matching using winner takes all (b) and continuous multi-label optimization (c).



(a) Jakomini

(b) Herrengasse

(c) Opera

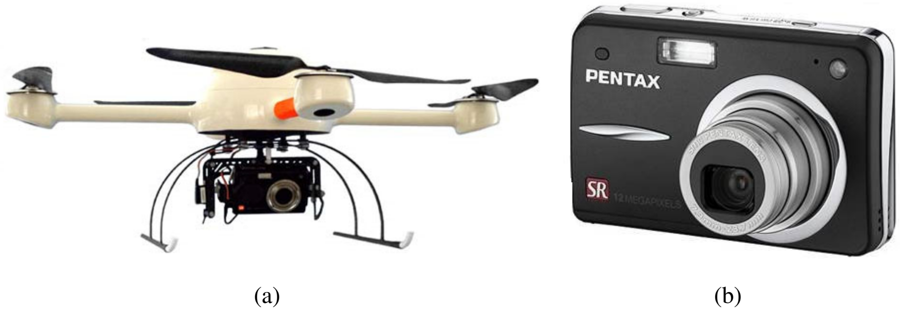
(d) Friendly Alien

(e) Railway station

Figure 3.18: Oblique point of view of texturized depth maps from landmarks of Graz taken at a GSD of 10cm.

3.3 Micro Aerial Vehicles for 3D Reconstruction

In the last few years, advances in material science and control engineering have turned micro aerial vehicles into cost efficient, flexible and rapidly deployable geodata acquisition platforms. For instance the micro-drone md4-200[1] depicted in Figure 3.19 has the ability for vertical take off and landing, provides position hold and autonomous way-point navigation and is equipped with a standard digital consumer camera that can be tilted (up to $90°$) to capture images from different angles. Due to the low operation altitude a very high resolution in terms of ground sampling distance can be achieved. At a distance of $10m$ to the object, MAVs allow a ground sampling distance that is $< 2cm$ and thus can compete with traditional surveying techniques like total stations and Differential Global Positioning Systems (DGPS) for the task of land survey and cadastral map generation. According to [Colomina et al., 2008], MAVs are a new paradigm for high-resolution low-cost photogrammetry and remote sensing, especially given the fact that consumer grade digital cameras provide a sufficiently high accuracy for many photogrammetric tasks [Gruen and Akca, 2008]. The presence of on board navigation, Global Positioning System (GPS) and Inertial Measurement Units (IMUs) allows MAVs to act as autonomous systems that fly in the air and sense the environment. The MAV can fly in altitudes of several hundred meters as well as in short distance to the object of interest. Hence, images can be obtained with much higher resolution in terms of ground sampling distance than possible by classical airborne large format digital camera system (e.g. UltraCamXp[2]) as shown in Figure 3.20.



(a) (b)

Figure 3.19: (a) Micro-drone md4-200 with attached PENTAX Optio A40 (b).

The main advantage of a MAV system acting as a photogrammetric sensor platform over more traditional manned airborne or terrestrial surveys is the high flexibility that allows image acquisition from unconventional viewpoints. Consider Figure 3.21: While the camera network in standard airborne and terrestrial surveys is normally restricted to flight lines or street paths, a MAV system enables more flexible, e.g. turntable like network configurations, which maximize scene coverage and allow superior accuracy in terms of triangulation angles. Furthermore, the photogrammetric network planning task [Chen et al., 2008] can be optimized and adapted to the scene since nearly any desired viewpoint can be reached.

[1] http://www.microdrones.com

[2] http://www.microsoft.com/ultracam

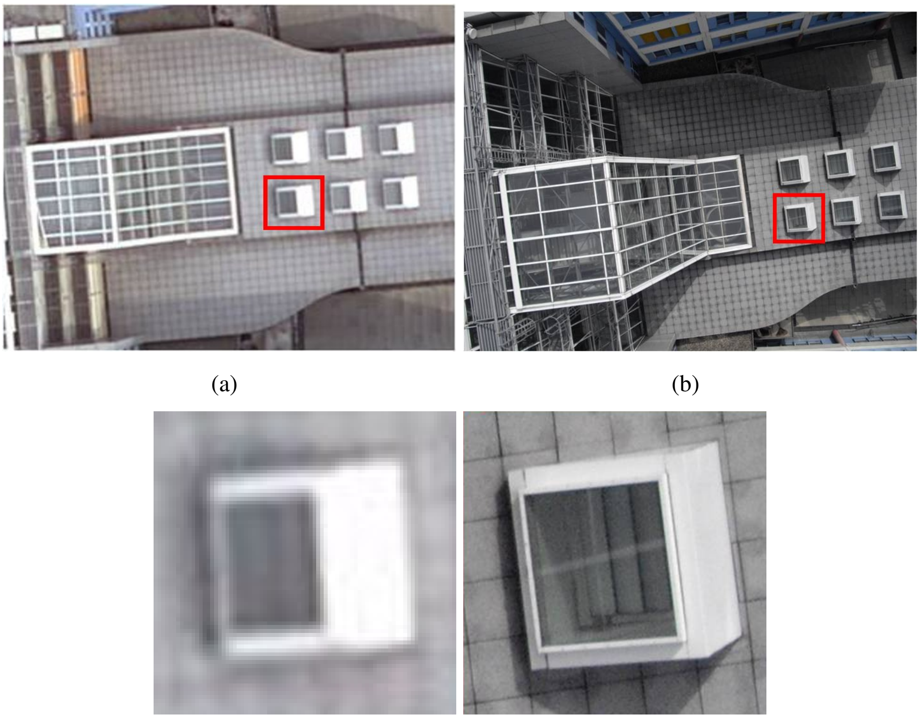
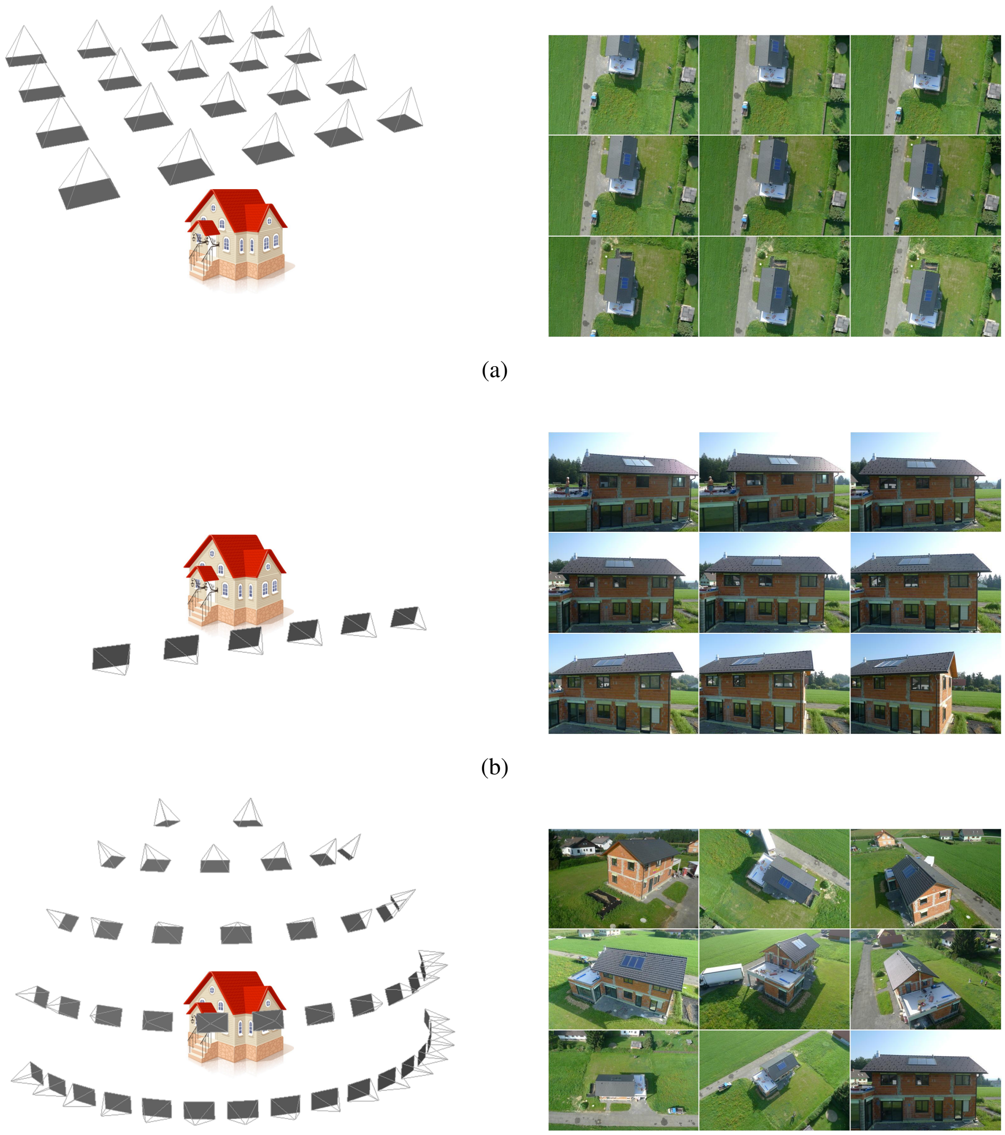

(a)
(b)
(c)
(d)

Figure 3.20: Comparison of images taken by a large format airborne camera (UltracmXp) with GSD 15cm at flying height 1000m to images taken by a MAV with a standard digital camera (Pentax Option A40) with GSD 1cm at flying height 10m.

3.3.1 GPS/INS Supported Matching

Micro Aerial Vehicles (MAVs) equipped with global positioning system and inertial sensors allow instant geo-referencing of acquired images. In contrast to aerial photogrammetry (see Section 3.2) where the external pose information can often be directly used for dense matching, the accuracy of light weight and low cost GPS/IMU sensors on MAVs normally does not reach the required level of precision for pixel accurate image alignment. However, it delivers a rough estimate of the camera poses and orientations. In this section we present an approach that leverages such imprecise prior information to speedup structure from motion computation in terms of feature matching and geometric estimation. We propose a view selection strategy that advances vocabulary tree based coarse matching by also considering the geometric configuration between weakly oriented images. Real world experiments are performed using data acquired by a micro aerial vehicle attached with GPS/INS sensors. Furthermore, in Section 3.3.3 we compare our method to a semi-automatic orientation approach based on the commercial PhotoModeler[1] software and demonstrate superior performance in terms of automation, accuracy and processing time.

[1] http://www.photomodeler.com



(a)

(b)

(c)

Figure 3.21: Typical camera networks and respective images from aerial (a), terrestrial (b) and (c) (MAV) survey. MAVs allows flexible image acquisition from unconventional viewpoints that enables a regular sampling of the visual hull of the scene of interest.

Feature Matching

Feature matching of unordered images is often the most time consuming part of a SfM algorithm. A typical solution to restrict the number of images for detailed feature matching is to use a coarse matching strategy (Section 2.5) to determine a reduced set of potentially matching image pairs. However, due to repetitive structure, the result of coarse matching techniques can often be arbitrarily wrong. Images that achieve a high similarity score do not necessarily show the same part of an environment (e.g. the images shown in Figure 3.22 are visually similar but are taken from two different buildings). Using global pose information as provided by a GPS/IMU unit, such ambiguous images can be filtered which further reduces the matching effort. Given the set of images $\mathcal{I} = \{I_1, \ldots, I_n\}$, associated with approximate knowledge of external pose measurements $\mathcal{G} = \{G_1, \ldots, G_n\}$, we select a subset V_i of potentially matching view pairs.

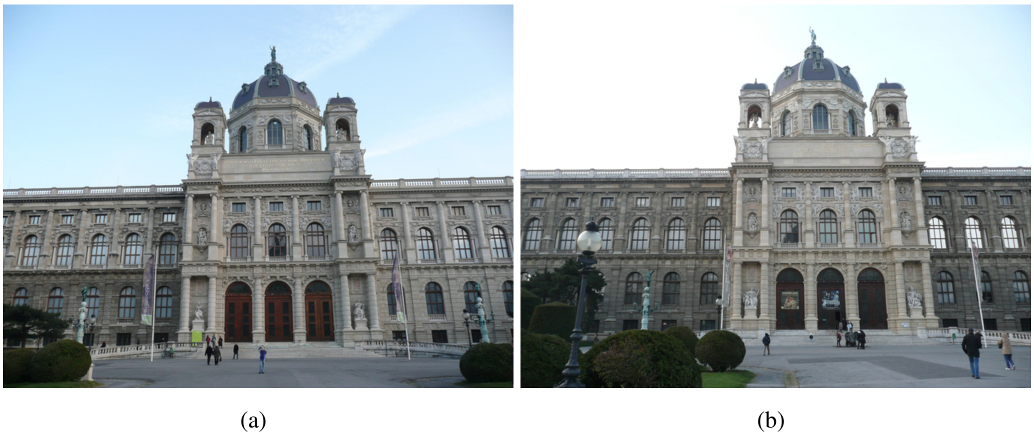


Figure 3.22: Visual similar facade images from two different buildings, (a) Museum of Art History and (b) Museum of Natural History, both located in Vienna.

External Pose Information

The external pose $G_i = [\, R \mid \mathbf{t} \,]$ is achieved by GPS and IMU, where R is a 3×3 rotation matrix and $\mathbf{t}$ a 3-space vector representing camera orientation and translation, respectively. Global position information is delivered by a standard GPS receiver in the WGS84 coordinate system which describes a position on earth as longitude, latitude and height. For further processing, the GPS datum is transformed in the Earth Centered, Earth Fixed (ECEF) coordinate system, which is a Cartesian system capable of representing reconstructions in global scale. The camera orientation is described by three angles yaw, pitch, and roll, where the yaw angle is aligned to magnetic north and the camera down vector points to the earth center. Thus, the external pose G_i is composed of a GPS datum transformed to the ECEF coordinate system and three rotation angles. In conjunction with the known intrinsic parameter K of the camera, we obtain the full projection matrix $\hat{P}_i$ for each image I_i,

$$\hat{P}_i = KG_i = K[\, R \mid \mathbf{t} \,]. \tag{3.27}$$

The retrieved projection matrices give a rough estimate of the camera position and orientation that is used for further processing.

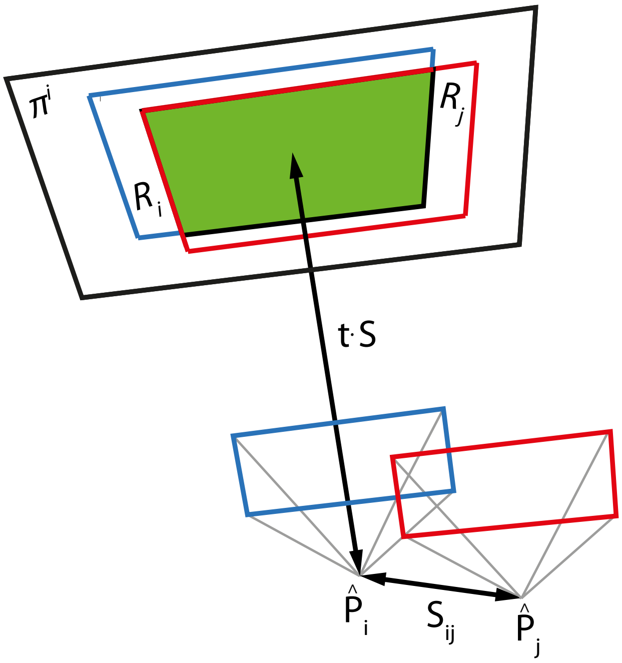


Figure 3.23: View overlap estimation. In front of camera $\hat{P}_i$, a plane π^i is defined with distance $t \cdot S$. R_i and R_j are the visible areas of π^i in $\hat{P}_i$ and $\hat{P}_j$ respectively.

View Selection

To identify images that potentially share corresponding points, we select for each image I_i a set $T_i = \{T_1 \dots T_k\}$ images that achieve a sufficient high probabilistic similarity score (Section 2.5.5). Next, images in T_i are filtered according to their GPS/IMU information using a coarse overlap criterion. If a detailed 3D model of the environment is available, the image overlap can be easily obtained by projecting and back-projecting the view frustum of view I_i into view I_j. In case that no model of the environment is present, we can only make weak assumptions on the maximum scene depth S_i that restrict the area observed by an image I_i. For instance, given a rough Digital Surface Model (DSM) the height above ground can be estimated which limits the maximal depth range for cameras looking towards the earth-surface (e.g. aerial image surveys). The Shuttle Radar Topography Mission (SRTM) [Farr et al., 2007] provides a DSM of the Earth at near-global scale, covering about 80% of the Earth's total landmass. For terrestrial data (i.e. horizontal looking cameras) the S_i can be fixed to a user defined threshold that is based on the maximal expected scene size. Furthermore, the maximum scene depth S_{ij} that can be recovered by an image pair $< I_i, I_j >$ depends on their baseline. We define,

$$S_{ij} = t \cdot d(G_i, G_j), \tag{3.28}$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance and t is factor that determines the required reconstruction accuracy. Given these constraints, we estimate the maximum scene depth S that can be reconstructed by an image pair $< I_i, I_j >$,

$$S = min(S_{ij}, S_i, S_j). \tag{3.29}$$

Furthermore, the images must have an overlap. To calculate a coarse overlap criterion $\mathcal{O}_j^i$, we define a plane π^i that is parallel to image I_i and whose distance to the camera center G_i is S. R_i and R_j denote the image extend of view I_i and I_j on the plane π^i. The image overlap $\mathcal{O}_j^i$ is computed by,

$$\mathcal{O}_j^i = \frac{a(R_i \cap R_j)}{a(R_i \cup R_j)}, \tag{3.30}$$

where $a(\cdot)$ returns the area of the projected rectangle. Figure 3.24 illustrates this concept.

Since feature descriptors like SIFT may tolerate only a maximum view angle change of approx. 30°, we require that the view vector of $\hat{P}_j$ and the normal of π^i enclose a maximum angle α. Otherwise $\mathcal{O}_j^i$ is set to zero. An illustration of the overlap calculation is shown in Figure 3.23. For every image pair $< I_i, I_j >$ with $I_j \in T_i$ we compute the overlap $\mathcal{O}_j^i$. If the overlap is above a fixed threshold, I_j is inserted into the set V_i which are later used for detailed feature matching. The first eight images of the set T_i and the set V_i for a sample image are depicted in Figure 3.25. The corresponding GPS positions are shown in Figure 3.26.

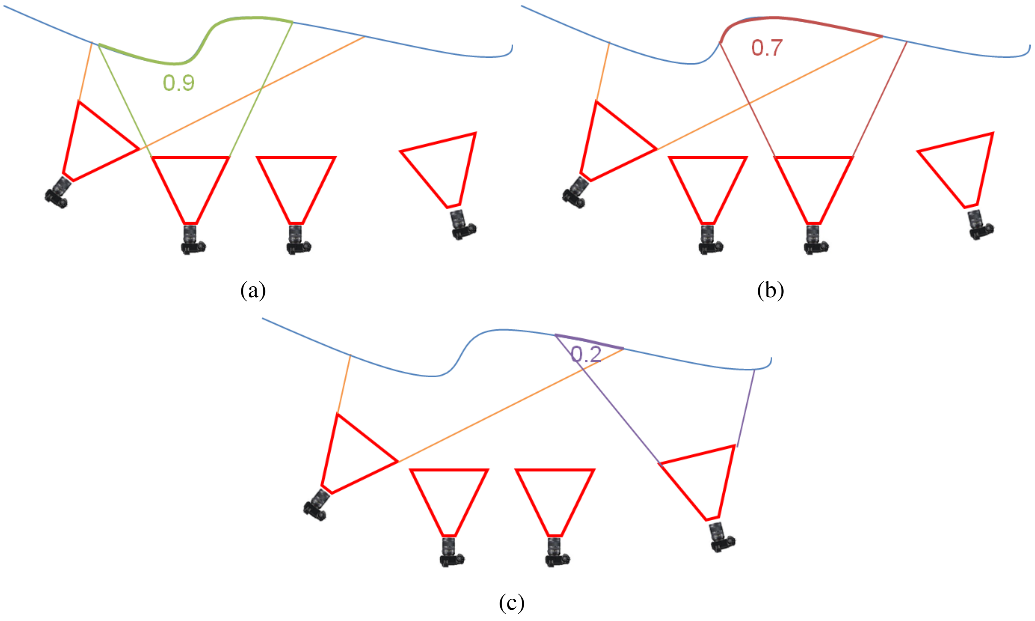


Figure 3.24: Computation of image overlap with known scene depth according to Equation (3.30)

3.3.2 View Selection Experiments

We evaluate our view selection approach on real world data acquired by a micro aerial vehicle[1] with integrated GPS/IMU sensors. The attached camera system delivers images of resolution

[1] AscTec Falcon 8 http://www.asctec.de

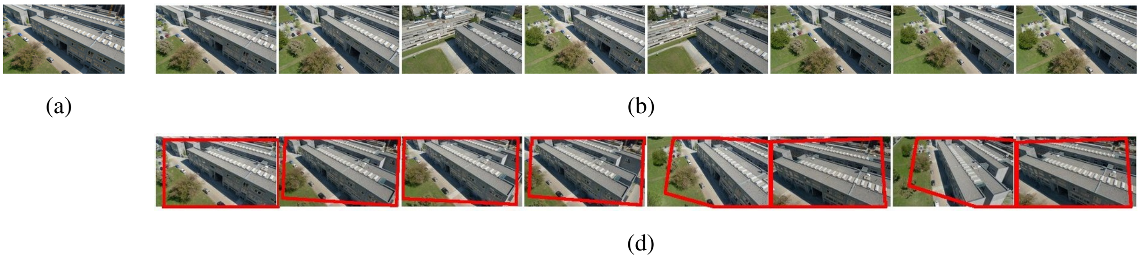


Figure 3.25: Vocabulary tree vs. overlap criterion. The first row shows the first eight images returned by the vocabulary tree given the query image (a). The second row shows the first eight images of the filtered vocabulary tree result sorted by their overlap value. The red box depicts which part of the rectangle R_i is visible in the current image.

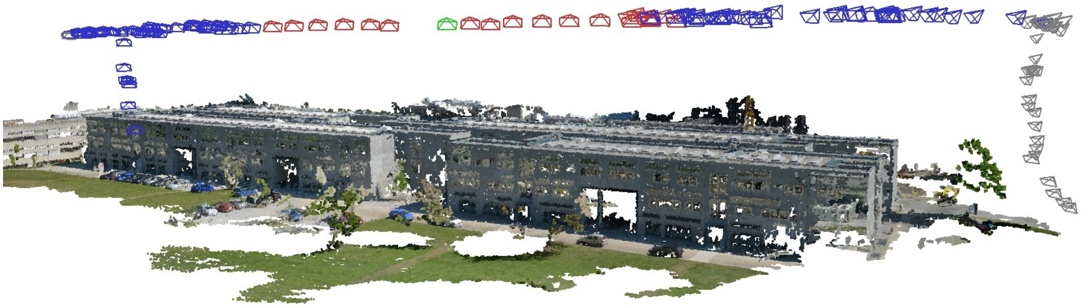


Figure 3.26: View selection of a trajectory containing 196 images. Absolute pose information G_i is provided by a GPS and IMU (colored pyramids indicate camera positions and orientations). Given a query image I_i (green) results in a set T (blue + red) of images with similar appearance delivered by the vocabulary tree. Images with an overlap value $\mathcal{O}^i_j$ of at least 50% are depicted in red.

3968×2232, the accuracy of the GPS receiver is about $2m$ and the relative position precision approximately $0.5m$. We assume that the camera origin and GPS/IMU sensor shares the same 3D position in space. This is a valid approximation if the distance to the scene is much larger than the offsets between the individual sensors. Images are acquired at an interval of 3s and a GPS/IMU tag is stored for each image, the synchronization accuracy between image and GPS/IMU datum is less than 0.5 seconds. During one flight mission of 10-15 minutes, about 200 images are acquired. For each input image we use a maximal number of d neighbors from the GPS view selection strategy (as described in Section 3.3.1) and compute the epipolar graph (we use $d = 20$). Our view selection approach reduces the matching effort from $O(n^2)$ to $O(nd)$. Compared to exhaustive image matching that considers ~ 40000 view pairs, only 4000 matching operations are performed which gives a ten-fold speedup. Figure 3.27 shows potential matching candidates from a vocabulary tree approach and matching pairs computed by our proposed geometric view selection criterion.

We use the method described in Section 2.8.1 to initialize camera orientations and 3D points. Given pairwise relative rotations, this algorithm computes global camera orientations and refines the orientation of the raw IMU initialized projection matrices $\hat{P}_i$. Next, bundle adjustment is

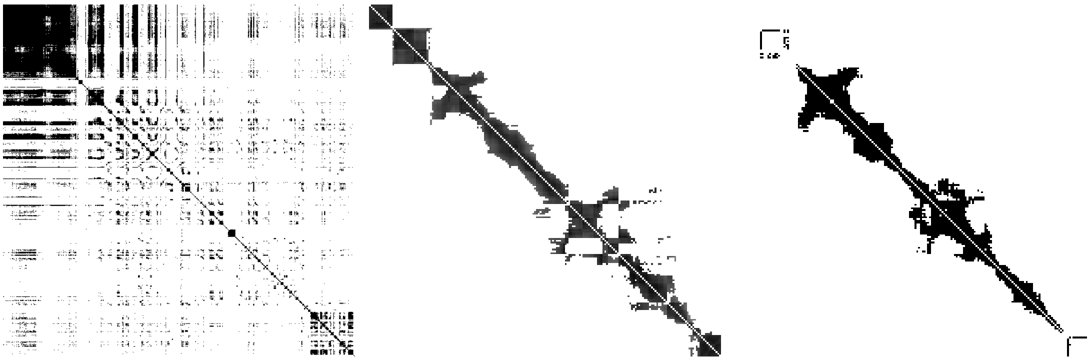

(a) (b) (c)

Figure 3.27: (a) Adjacency matrix showing potential matching candidates (dark pixel) between view i,j from a vocabulary tree scoring and (b) potentially matching image pairs as computed by our geometric view selection strategy. (c) Epipolar geometries determined by exhaustive image matching representing the ground truth.

executed to minimize the reprojection error. While the Root Mean Squared Error (RMSE) between global registered rotation matrices and the final optimized bundle adjustment result is about $0.1°$, the initial IMU orientation deviates on average more than $10°$. Figure 3.28 shows GPS/INS camera positions (blue) and respective optimized cameras (red) using image based measurements and bundle adjustment.

We compare the reconstruction results of our proposed approach to the one obtained by an incremental structure from motion pipeline. After aligning both results with a robust similarity transform in a metric coordinate system, the mean as well as the median residual error between camera centers is about $0.023m$. On average, the deviation between the view vectors of the reconstructed camera positions is $0.03°$, only. Hence our approach is competitive to incremental SfM in terms of accuracy. Moreover, our proposed approach is computationally more efficient. Table 3.5 gives detailed timings of the algorithm.

While an incremental structure from motion pipeline (e.g. the bundler software[1]) requires a repeated call of a bundle adjustment optimizer (with time complexity O(n^3), where n is the number of frames), our proposed algorithm only requires rotation an track initialization and one single bundle adjustment call.

3.3.3 Accuracy Comparison to PhotoModeler

We performed two test-flights to acquire images for 3D reconstruction of buildings with the microdrone md4-200 and an attached PENTAX Optio A40 camera (Figure 3.19). The camera was precalibrated and the zoom was fixed to a wide angle. The survey was performed by manual remote control, 615 still images with a resolution of 4000×3000 square pixels were captured from different viewpoints. The acquired images are unordered, since sequential image acquisition

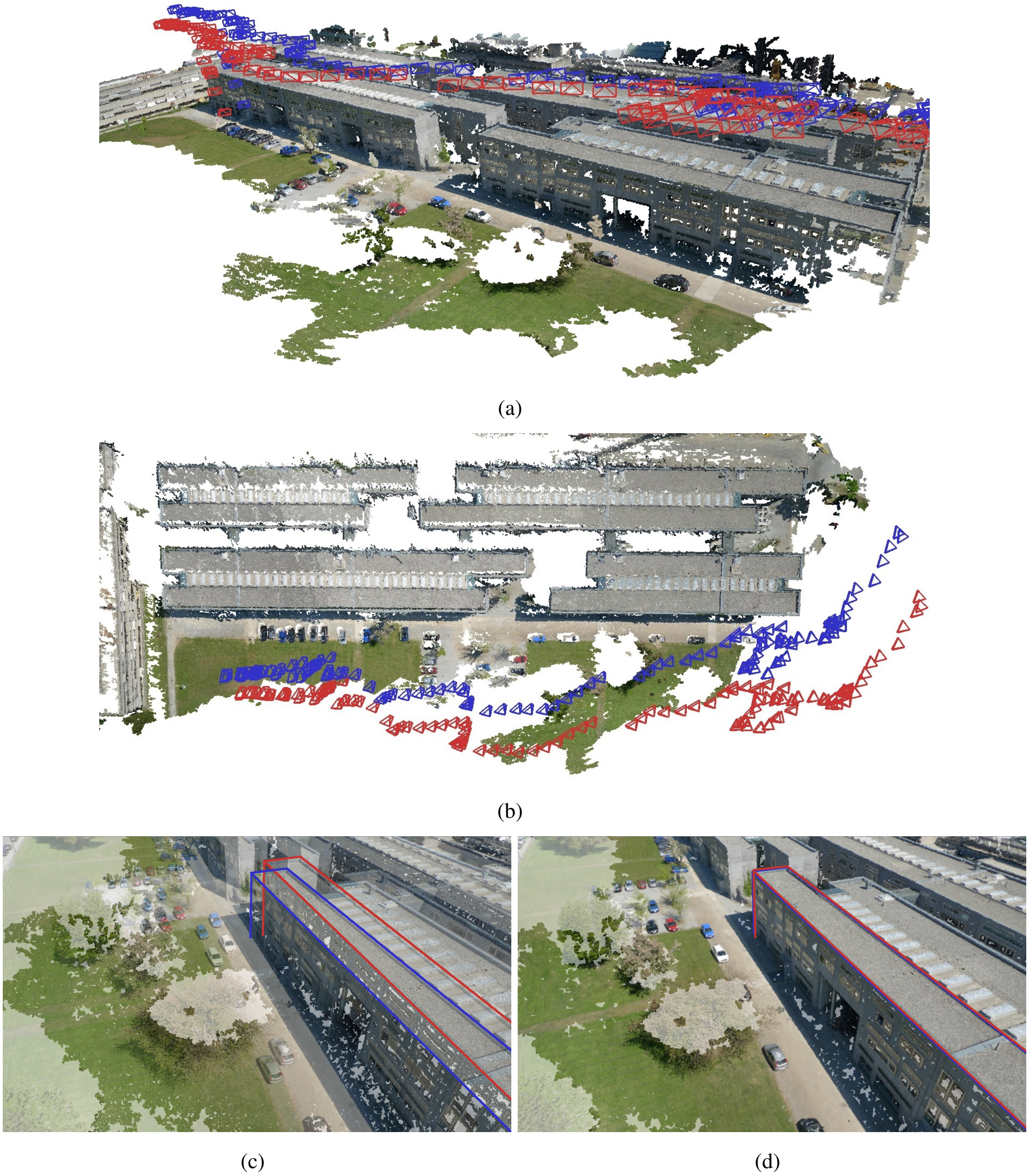
[1] http://phototour.cs.washington.edu/bundler/

Operation	time [s]
SIFT (3968 × 2232 pixel)	0.4
Coarse Matching	0.05
Matching (5000 × 5000)	$d \times 0.044$
RANSAC (5-pt, N=2000)	$d \times 0.12$
Incremental SfM (200 views)	1800
Our approach (200 views)	190

Table 3.5: Timings for individual processing steps (per image) and comparison to an incremental structure from motion approach. Note, d reflects the number of considered images for detailed feature matching and geometric verification.

could not be guaranteed due to flight path restrictions. To determine the quality of the image based 3D reconstruction from MAV images, eight ground control points were determined using a total station (with an accuracy of $\epsilon \pm 1cm$, see Figure 3.29). This data is considered as ground truth and is later used to asses the object space error of the automatic computed structure from motion results. Images were processed by the reconstruction pipeline described in Chapter 2. Figure 3.30 shows the epipolar graph and respective reconstruction results from 615 acquired still images (Figure 3.31).

We compare our fully automatic structure from motion approach to the semi-automatic PhotoModeler software (version 6) for the task of exterior image orientation. Since it turns out that processing hundreds of images is impracticable for a semi-automatic system, we restrict our evaluation to a subset of 23 manually selected images from one building facade (corresponding to reconstruction R1, see Figure 3.32). The processing steps of the PhotoModeler approach include the semi-automatic measurement of tie and control points, bundle adjustment and fine tuning. Four different orientation methods were conducted: selfcalibration with constant/variable intrinsics and with/without reference point constraints by using fifteen 3D control points, respectively. All methods give consistent results, on average a reprojection error of 0.5 pixel is reported. A detailed, quantitative comparison of the PhotoModeler orientation output with results from our structure from motion pipeline is summarized in Table 3.6. The PhotoModeler software provides a semi-automatic, incremental approach for structure from motion computation. Image correspondences are established manually, the epipolar geometry guides the correspondence search. Even though an experienced user performed the PhotoModeler reconstructions, the orientation of a subset of 23 images still requires about eight man hours (and is troublesome and strenuous work). On the other hand, with our fully automated system, the full set of 615 images can be processed at once and within a timeframe of 3.5 hours on a standard PC and a single GPU. We achieve identical results in terms of reprojection error, but with a higher confidence in the solution, since many more tie points are utilized. Furthermore, the automatic approach is scalable and allows registration of many more images much faster. For instance, in our pipeline processing one image takes about 20s, whereas orientation with the PhotoModeler software requires more than 20min man workload.



(a)

(b)

(c)

(d)

Figure 3.28: (a) Side view and (b) top view of GPS/IMU camera orientations (blue) and adjusted camera orientations (red) obtained by our global structure from motion approach. (c) Offset of GPS/INS oriented image (blue lines) to ground truth 3D model (red lines) and (d) photoconsistent camera orientation by structure from motion.

3.3.4 Object Space Error

The reprojection error is a suitable measure to assess the precision of camera orientations in image space, but for a practical application, the error in object space is of interest. Therefore, we rely on

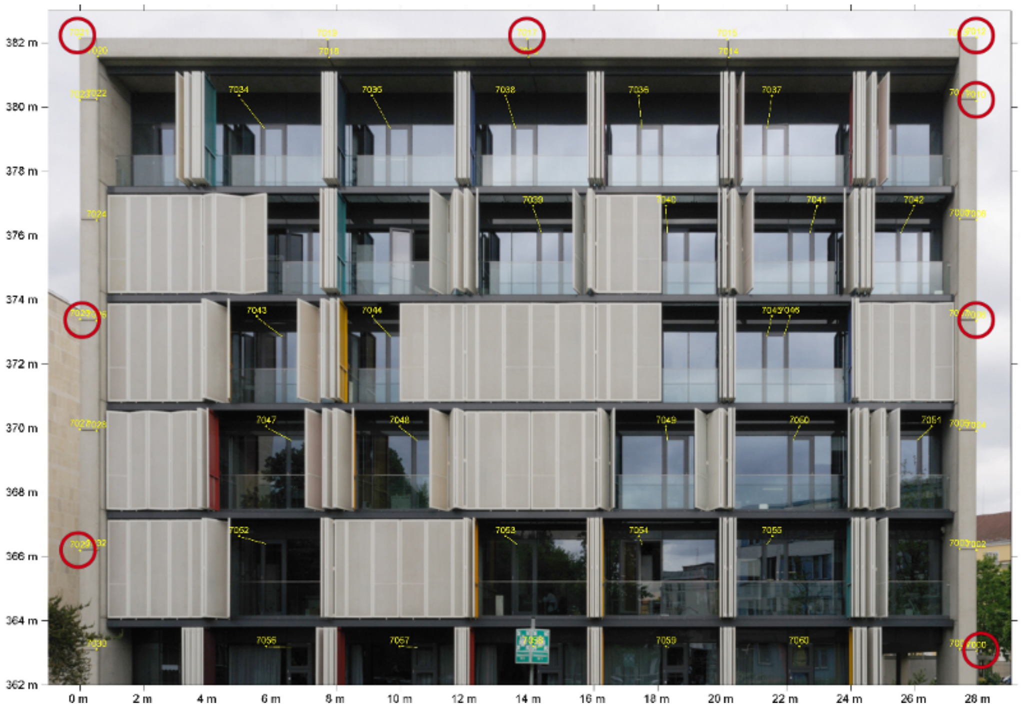


Figure 3.29: Orthographic projection of a building facade with the eight ground truth control points (red circles) used in our evaluation.

	PhotoModeler	sfm-approach
# processed views	**23**	**615**
# registered views (R1)	23	239
# 3D points	237	58791
avg. # points/image	**99**	**3160**
avg. # rays/3D point	10	13
avg. triangulation angle	10°	6.7°
avg. reprojection error	0.458	0.460
processing time [h]	8	3.5
processing time/image [s]	**1252**	**20**

Table 3.6: Comparison of the semi-automatic PhotoModeler orientation to our proposed fully-automatic structure from motion system (sfm-approach), the values correspond to reconstruction result R1 (see Figure 3.32).

control points measured by a total station to estimate an absolute error measure. The landmarks are determined at well localized structures, like building corners and junctions (see Figure 3.29). Thus, image measurements with respect to the corresponding landmark are easy to establish. We restrict our evaluation to one building facade and eight well localized control points. For each image we estimate the 2D coordinates belonging to the 3D control point (manually by visual

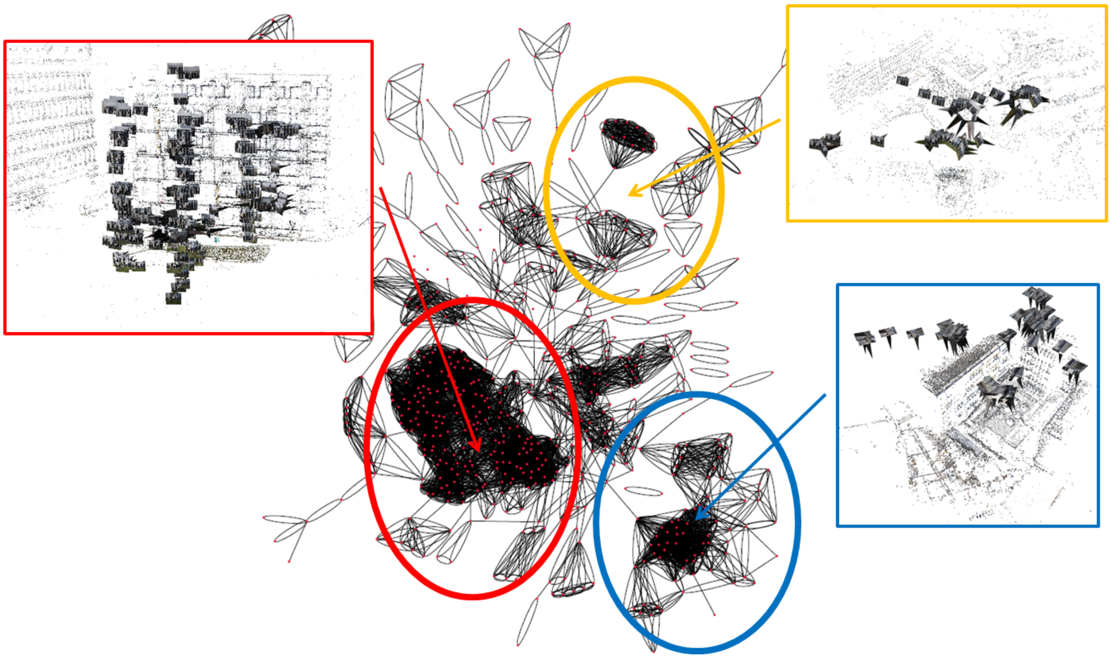
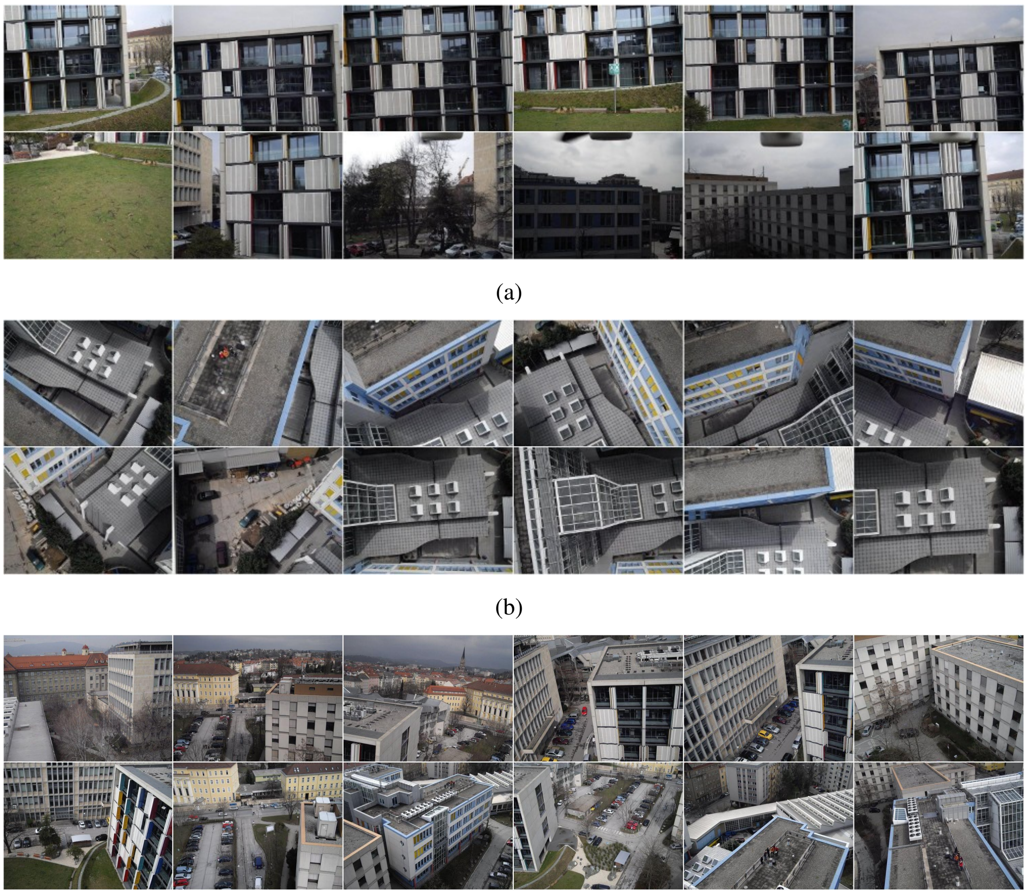


Figure 3.30: Epipolar connectivity graph of the whole dataset, clusters in the graph represent a high degree of geometric connectivity. Three connected components corresponding to disjoint 3D models are obtained.

inspection) and link the measurements into point tracks. In practice, we only use a subset of images to measure observations, but ensure that for each control point at least three measurements are provided and the triangulation angle is sufficiently high ($\bar{\alpha} > 20°$). Next, we use a linear triangulation method [Hartley and Zisserman, 2000] followed by bundle-adjustment to triangulate the measurements into 3D space. To measure the object space error, we computed the 3D similarity transform between 3D control points and respective triangulated tie points. The alignment can be computed with a minimal number of three point correspondences, but using more than three points in a least squares manner will result in a closer alignment. Hence, we use the leave-one-out cross-validation [Kohavi, 1995] technique to assess the accuracy of our orientation results. We take seven correspondences to compute the parameters for the similarity transform and use the remaining point to estimate the object space error ϵ between observation X and ground truth point $\hat{X}$,

$$\epsilon = \sqrt{(X_x - \hat{X}_x)^2 + (X_y - \hat{X}_y)^2 + (X_z - \hat{X}_z)^2}. \tag{3.31}$$

Table 3.7 summarizes our evaluation, the error varies between 0.4 to 5.4cm, overall a RMSE of 3.2cm is achieved. Note, the reprojection error of the triangulated tie points varies between $1.1 - 2.5$ pixel, this is in accordance to the expected uncertainty induced by the manual tie point extraction. A subpixel accurate measurement of tie points (e.g. 0.5 pixel) would lead to a RMSE of about 1.5cm which is close to the precision of the total station.



(a)

(b)

(c)

Figure 3.31: Sample images corresponding to connected component extracted from the epipolar graph shown in Figure 3.30

.

Point ID	7000	7006	7010	7012	7021	7017	7025	7029
# measurements (images)	3	6	3	3	10	3	10	6
avg. triangulation angle [°]	107.2	21.9	23.2	23.2	33.4	54.7	69.5	84.6
avg. reprojection error [pixel]	1.18	1.67	2.24	1.63	1.58	1.16	2.44	0.85
object space error [cm]	4.2	0.4	2.5	4.5	0.6	2.8	1.7	5.4

Table 3.7: Reprojection error and object space error determined by leave-one-out cross-validation for eight ground truth control points.

3.3.5 Conclusion and Discussion

Micro aerial vehicles equipped with digital cameras are flexible platforms for capturing images for 3D reconstruction. A MAV can fly at altitudes of several hundred meters as well as close to

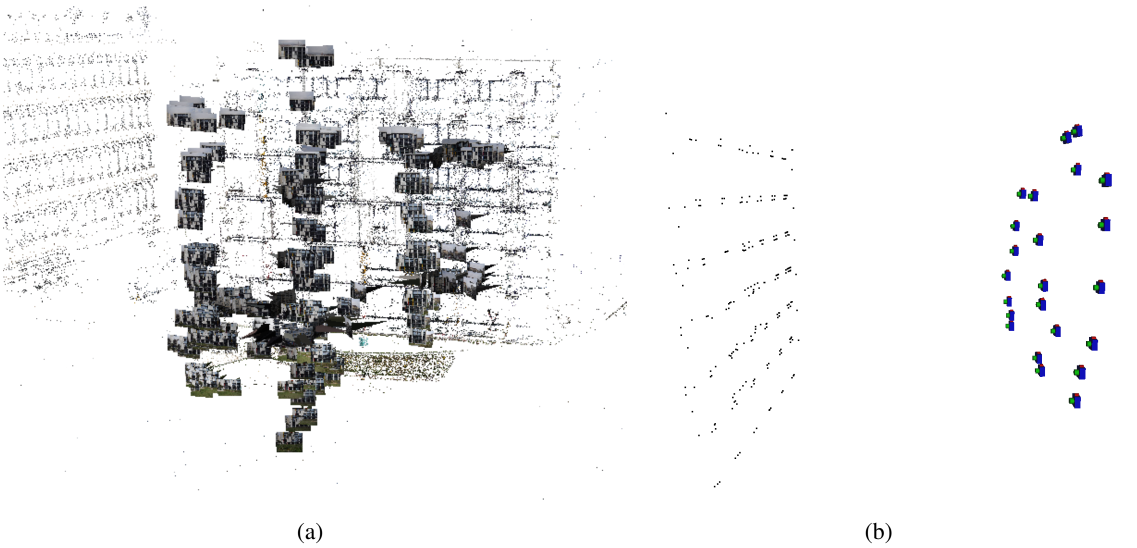


Figure 3.32: Orientation reconstruction R1: (a) perspective view of camera orientations (239) and respective 3D points (58791) obtained by our automatic structure from motion system. (b) Orientation result obtained by semi-automatic processing using the PhotoModeler software, a subset of 23 manually selected images is used.

the ground. This enables the reconstruction of large areas at a high accuracy in terms of Ground Sampling Distance (GSD). These platforms are often equipped with global positioning systems and inertial measurement units that allow rough geo-referencing of respective images. In this section we presented a structure from motion algorithm that effectively takes advantage of GPS/IMU information for matching, view selection and geometric estimation. The main contributions of the proposed method are (i) a view selection strategy based on global position/orientation information that limits the matching effort and (ii) a fast and scalable reconstruction approach that relies on global rotation registration and robust bundle adjustment. We tested our approach on real world scenarios using data acquired by a MAV. From our experiments we conclude that our approach is robust, scalable and computationally more efficient than previous methods. The quality of camera orientations and 3D structure is on a par with state-of-the-art SfM approaches. We compared the orientation results of our fully automatic structure from motion pipeline to a standard, semi-automatic approach based on the PhotoModeler software. Our system achieves the same accuracy in terms of reprojection error, but at a higher confidence, since many more tie points are utilized than in the semi-automatic approach. Furthermore, our method is scalable to larger datasets and allows much faster image orientation. In our experiments we achieved a speedup of about 60× over semi-automatic processing with the PhotoModeler software. The achieved accuracy the reconstruction results from our experiments is in the range of the accuracy of current total station measurements and DGPS.

Chapter 4

Robust Reconstruction by Bayesian Reasoning

Most existing structure from motion approaches for unordered image collections have problems if duplicate structures/objects appear in the scene. Examples of such cases are symmetric and repeating facades in city environments or multiple instances of (man made) objects. When matching image pairs containing visually similar but different instances of an object, wrong epipolar geometries may arise. Conventional structure from motion algorithms [Snavely et al., 2006, Irschara et al., 2007, Martinec and Pajdla, 2007, Frahm et al., 2010] perform only monotone reasoning on the epipolar geometry which often leads to catastrophic failures in the reconstruction [Snavely, 2008, Zach et al., 2010, Roberts et al., 2011]. This is especially true when large structures of a captured scene have similar appearance. For instance, conventional incremental structure from motion algorithms fail on the castle scene as depicted in Figure 4.1 due to the large repetitions of similar structure. The algorithm delivers folded ghost structures caused by the wrong data association (see Figure 4.2(b),(c)) .

In this Chapter we describe an algorithm for non monotone reasoning about view triplets which enables to identify epipolar geometries that are caused by scene repetitions. After detection and exclusion of wrong geometric relations, a consistent 3D model can be obtained (see Figure 4.2(d)(e)).

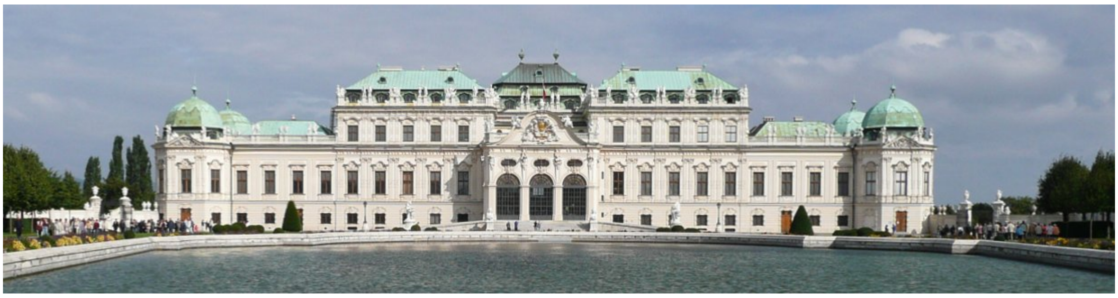
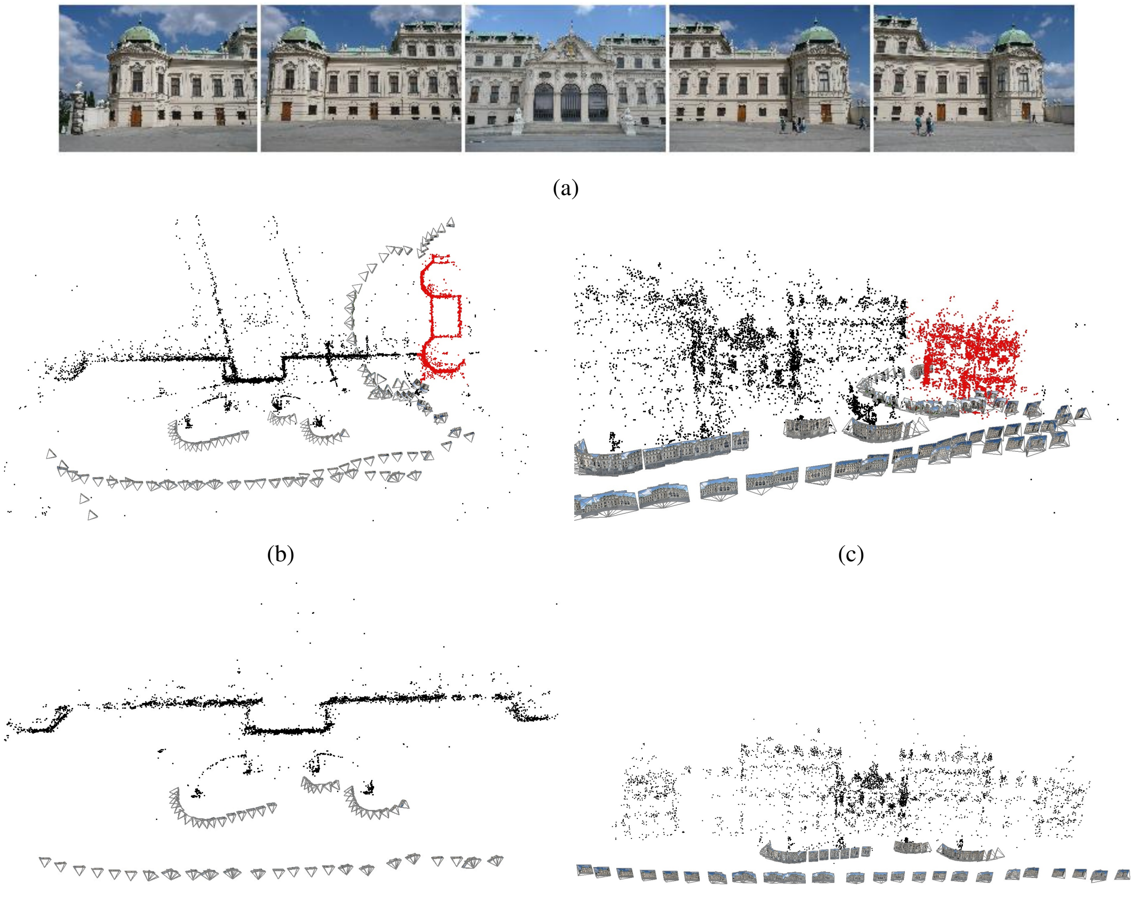


Figure 4.1: Belvedere Castle showing symmetric scene structure.



(a)

(b)

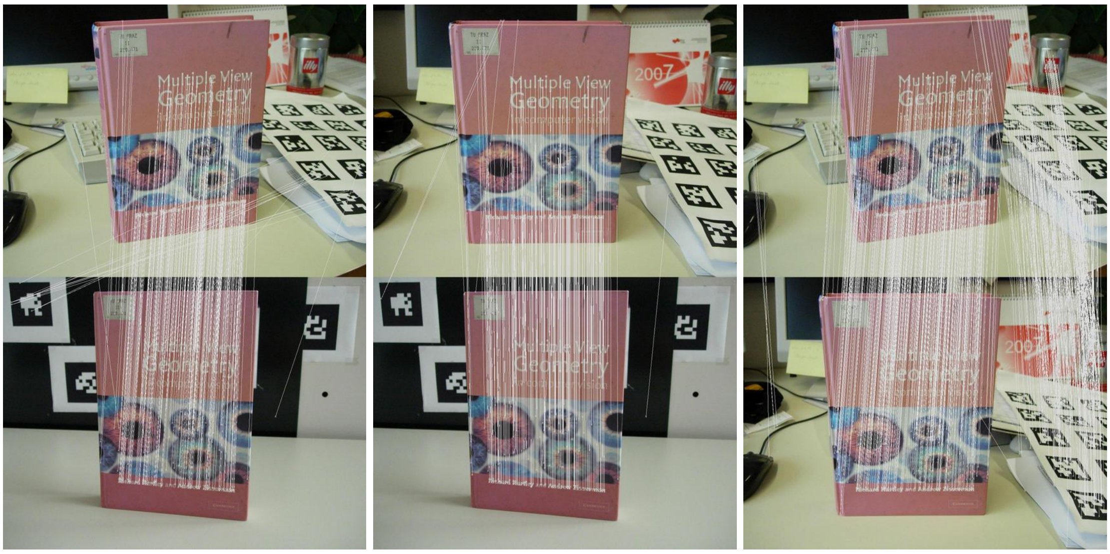
(c)

(d)

(e)

Figure 4.2: (a) Sample images from the Belvedere dataset. The identical structures in the scene result in a folded reconstruction (b) (c). The red structure is mistakenly reconstructed due to the duplicate scene structure. (d) (e) Geometrically consistent model obtained by our proposed approach.

Let us consider the somewhat artificial, but nevertheless illustrative example depicted in Figure 4.3(a): there is a substantial number of correspondences between the two views established on the common object appearing in both scenes, but the overall scenery is clearly different and the hypothesized epipolar geometry (EG) between these two views is obviously wrong. Note, that from a geometric viewpoint we do not know the depth of the background, and the obtained EG indicated by the correspondences might be indeed right. Rejecting the EG solely on the basis of two views can be done by incorporating a prior assumption on the depths found in the scene. Such assumption limits the epipolar search from an infinite corresponding line to a bounded line segment. Alternatively, a higher understanding of the captured scene e.g. by estimating depths from monocular cues [Torralba and Oliva, 2002, Saxena et al., 2005] will allow reasoning about the validity of the hypothesized EG.



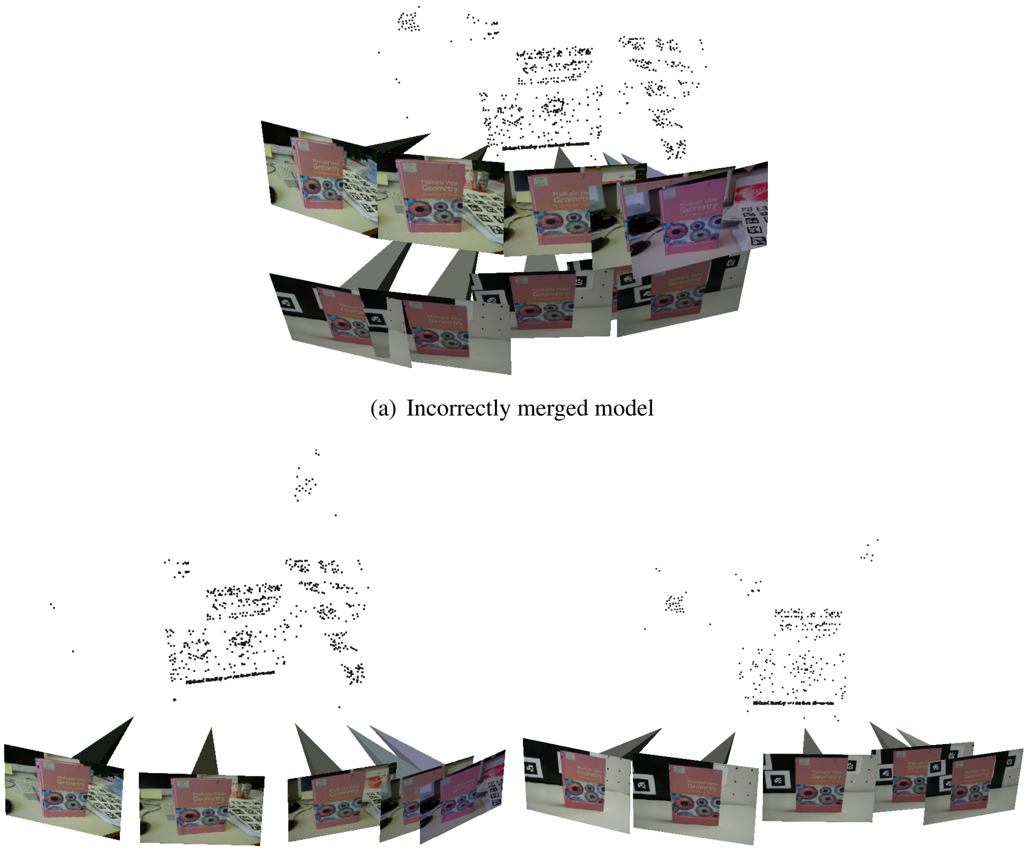
(a) $V_1 \leftrightarrow V_3$ (b) $V_2 \leftrightarrow V_3$ (c) $V_1 \leftrightarrow V_2$

Figure 4.3: Correspondences found for a view triplet (V_1, V_2, V_3).

We follow an approach that takes a different route not requiring scene understanding or strong assumptions on the scene depths. Imagine, there is a third image available, which has a correct EG with one of the originally provided views. As indicated in Figure 4.3(b) and (c), there is again significant evidence for EGs between all three images. Nevertheless, the first and the second view share more and spatially better distributed correspondences. Under the assumption of correct EGs between all view pairs (and the correctness of the relative poses between the views) there is a substantial amount of correspondences found between the first and the second view not appearing in the predicted position in the third view. These “missing correspondences” provide a strong evidence, that there is something wrong with the EG between the first and the second view. Note, that the underlying reasoning is only performed on geometric relationships between multiple views. Figure 4.4 shows camera poses and respective 3D models of incorrectly merged views and the correctly separated models using our proposed approach.

This kind of reasoning about correct and false image relationships from additionally provided images is useful, if the target application is to obtain 3D models by a vision based structure from motion approach. Our goal is to augment a 3D reconstruction pipeline with the ability to detect and to recover previously incorrectly established EGs between images. Hence we propose a method to detect incorrect EGs that arise due ambiguous objects. The proposed approach enables the seamless integration of non-monotone reasoning into structure from motion computation. Therein it is different from almost all visual modeling approaches proposed so far.

This chapter is organized as follows: Section 4.1 outlines relevant earlier work. Section 4.2 describes our approach to detect implausible two view geometries. The remaining two view relations are collected to constitute consistent reconstructions as described in Section 4.3. The 3D structure and motion computation for individual reconstruction is briefly sketched in Section 4.4,



(a) Incorrectly merged model

(b) Individual model #1

(c) Individual model #2

Figure 4.4: Generated 3D models by found correspondences only (a), and correctly separated models obtained by our proposed method (b) and (c).

and Section 4.5 depicts experimental validations. Finally, Section 4.6 summarizes this work and indicates future research directions.

4.1 Related Work

In [Martinec and Pajdla, 2006] a system to upgrade relative poses computed for image pairs to a full 3D reconstruction is proposed. The authors introduced the notions of *importance* and *reliability* of epipolar geometries between two images. The importance of an EG estimates the impact of the particular EG on the overall 3D geometry, whereas the reliability indicates the certainty about the EG. In their subsequent work [Martinec and Pajdla, 2007] the separation of rotation registration and translation registration is made more explicit, and their approach was substantially accelerated by considering only appropriately selected 3D points. Identification and removal of non-existent epipolar geometries is explicitly considered, but only by detecting image pairs with large error residua. We suppose that incorrect EG as depicted in Figure 4.3 still remain unnoticed.

The approach presented in [Steele and Egbert, 2006] shares several features with our proposed one: utilization of camera adjancency graphs and minimum spanning trees (MST), and the explicit validation step for camera poses. The camera adjacency graph is initially created by estimating the image similarities using a histogram measure. During MST construction the induced camera poses are verified by comparing dense depth maps. This is in contrast to our approach, where we first verify epipolar geometries using potential view triplets, and perform the MST construction step afterwards.

Schindler et al. [Schindler et al., 2007b] employ reasoning about missing (respectively invisible) feature matches to infer the temporal ordering of an unordered collection of images. Valid orderings of the images are those not violating a continuity constraint on the observed features. Thus, the main task is to solve a constraint satisfaction problem (CSP) in order to infer a suitable temporal ordering. The intractability of global CSP approaches is addressed by a greedy and local algorithm. Note that this approach is currently not fully automated: feature detection and matching is performed by a human operator.

Predicting correspondences in order to refine matches in a wide-baseline multiple view framework is part of the approach described in [Ferrari et al., 2003]. The aimed transitivity of matching relations guides the matching procedure, thus increasing the number of correspondences found in multiple views.

Using view triplets for structure and motion computation is well established, e.g. by utilizing the trifocal tensor [Hartley, 1995]. In [Beardsley et al., 1996] triplets are used to determine structure and motion for image sequences using a robust approach for trifocal tensor computation. In [Fitzgibbon and Zisserman, 1998], this idea is extended further to merge overlapping and consecutive image triplets into longer subsequences using a hierarchical approach. The implicit assumption, that three consecutive views are good candidates for trifocal tensor estimation, was relaxed in [Nistér, 2000], where "wide tensors" spanning between appropriate keyframes are employed. Since we assume calibrated cameras, and in order to avoid special handling of dominant planar scenes we utilize the five-point method [Nistér, 2004] in conjuction with robust rotation and translation registration.

Explicit Bayesian reasoning for 3D reconstruction is typically encountered in two very different topics: most prominently, dense stereo computation incorporating a smooth shape prior has its roots in the Bayesian formulation of the dense stereo problem [Geiger et al., 1995, Belhumeur, 1996, Sun et al., 2003], where it naturally leads to Markov random field approaches. Moreover, probabilistic methods are employed for least-squared model estimation and selection in multiple view geometry [Torr, 2002, Pollefeys et al., 2002].

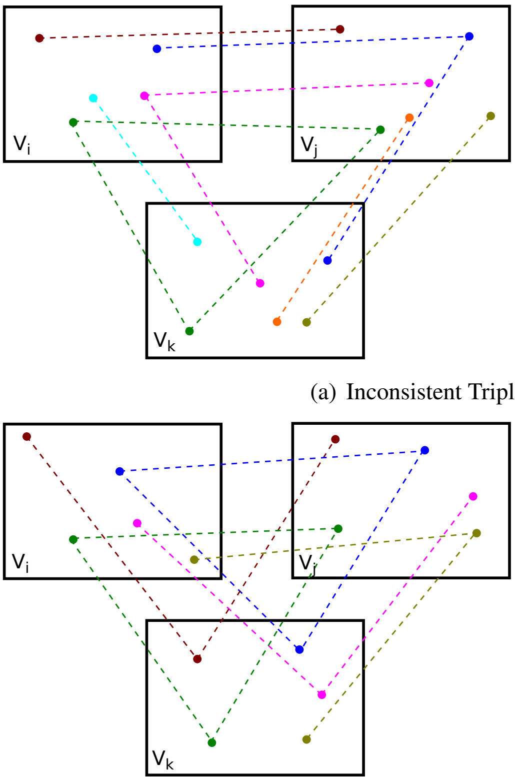
Enforcing consistency of geometric relations by chaining transformations over cycles along a loop in a graph of epipolar geometries is described in [Zach et al., 2010]. Large deviations between chained and actually observed transformations indicate conflicting edges among the involved relations. The proposed method uses a Bayesian network to infer the most likely set of incorrect transformations in the graph. Recently, [Roberts et al., 2011] proposed an approach that includes temporal inference to remove erroneous match pairs which can occur when different structure instances are matched based on visual similarity. This concept is related to the work of [Schindler and Dellaert, 2010] where temporal inference on structure from motion reconstructions is performed.

4.2 Reasoning About View Triplets

Recent work in 3D reconstruction has shown that structure from motion based on reasoning about view triplets [Fitzgibbon and Zisserman, 1998, Havlena et al., 2009] in general is more robust against false feature matches than monotone reasoning on pairwise epipolar relations. In [Klopschitz et al., 2010] we have demonstrated that implicit loop closing using correspondence information from locally matched image triplets is more robust than reliance on the global structure. Given a set of pairwise correspondences for input images, considering correspondences merged into image triplets reduces the number of outliers considerably. However, previous approaches do not directly address the data association problem. We propose to resolve these ambiguities using features that are matched between two images but not detected in a third image. Consider an image triplet (V_i, V_j, V_k), under the assumption that the geometric relations between two views is consistent, a large fraction of image features matched between two images (V_i, V_j) must be regained in a third view V_k. On the other hand, if features between (V_i, V_j) are not observed in a third one, it is likely that the third image observes a different instance. The concept is depicted in Figure 4.5. We formulate the problem as a probabilistic graphical model which enables us to detect incorrect two view geometries by reasoning about missing correspondences retrieved from image triplets.

4.2.1 Basic Formulation

Let $\mathcal{V}$ be a set of views $\mathcal{V} = \{V_1, \ldots V_n\}$. We denote the event that two particular views V_i and V_j are related by a *visually observable* epipolar geometry by $V_i \wedge V_j$. We will denote $V_i \wedge V_j = 1$, if there is a true epipolar relationship between these views, and $V_i \wedge V_j = 0$ otherwise. Establishing or rejecting this hypothesis is based on image observations and correspondence search. Let C^+_{ij} denote the set of robustly determined inliers of the potential image correspondences between V_i and V_j, e.g. by using a RANSAC [Fischler and Bolles, 1981] approach. We do not aim on predicting the exact positions of the inliers, hence we rather focus on the number of observed correspondences, $N^+_{ij} := |C^+_{ij}|$. Assume, we can estimate the prior probability $P(N^+_{ij}|V_i \wedge V_j)$ that we observe those correspondences under the assumption of $V_i \wedge V_j$ (either 0 or 1). Now, let us look at view triplets (V_i, V_j, V_k): First, we use the abbreviation $V_i \wedge V_j \wedge V_k = 1$ for $V_i \wedge V_j = 1$, $V_i \wedge V_k = 1$ and $V_j \wedge V_k = 1$, and $V_i \wedge V_j \wedge V_k = 0$ if any EG in this triplet is wrong. Under the premise of $V_i \wedge V_j \wedge V_k = 1$ (i.e. (V_i, V_j, V_k) forms a visually well-founded view triplet), we can take correspondences between e.g. V_i and V_j, and we expect to find the respective features again in V_k (since $V_i \wedge V_j \wedge V_k$ is assumed to be true). Examining regained features in V_k would only strengthen the belief in $V_i \wedge V_j \wedge V_k = 1$. More interesting are those correspondences between V_i and V_j which are *not* found in V_k. Observing many of these missing features consequently reduces the belief in $V_i \wedge V_j \wedge V_k = 1$. Denote the correspondences between V_i and V_j not detected in V_k by $C^-_{ij \to k}$. Again, we assume that the prior probability $P(N^-_{ij \to k}|V_i \wedge V_j \wedge V_k)$ can be estimated



N^+_{ij}	4
N^+_{ik}	3
N^+_{jk}	4
$N^-_{ij\to k}$	1
$N^-_{ik\to j}$	1
$N^-_{jk\to l}$	2

(a) Inconsistent Triplet

N^+_{ij}	3
N^+_{ik}	4
N^+_{jk}	5
$N^-_{ij\to k}$	0
$N^-_{ik\to j}$	0
$N^-_{jk\to l}$	0

(b) Consistent Triplet

Figure 4.5: (a) Consistent epipolar relations between views (V_1, V_2), (V_2, V_3), (V_1, V_3) but substantial amount of missing correspondences within the triplet. Our algorithm is able to detect such configurations and to reject inconsistent epipolar geometries. (b) Example of a plausible/consistent triplet configuration according to our quality criterion.

(where $N^-_{ij\to k} = |C^-_{ij\to k}|$). For simplicity, and because the third view V_k truly adds information to the view pair (V_i, V_j), we assume that the observable events N^+_{ij} for all i and j, and $N^-_{ij\to k}$ are pairwise independent. Of course, N^+_{ij} depends on the hidden variable $V_i \wedge V_j$, but not on the truth of epipolar geometries between other views. Likewise, $N^-_{ij\to k}$ only depends on the three latent variables $V_i \wedge V_j$, $V_i \wedge V_k$ and $V_j \wedge V_k$ constituting this view triplet.

These assumptions on the statistical independence result in a directed graphical model as depicted in Figure 4.6. Of course, the belief network can be extended to cover all view triplets. We do not examine this approach further for the following reasons:

- Firstly, the undirected belief network after normalization results in a loopy graph, hence exact inference is expensive. The mutually recursive dependence of the latent variables $V_i \wedge V_j$ on the other variables $V_i \wedge V_k$ and $V_j \wedge V_k$ in the same triplet is apparent, since e.g. the belief in $V_i \wedge V_k$ depends *and* influences the belief in $V_i \wedge V_j$.

- Secondly, we aim for a primarily incremental 3D reconstruction pipeline. Performing a full

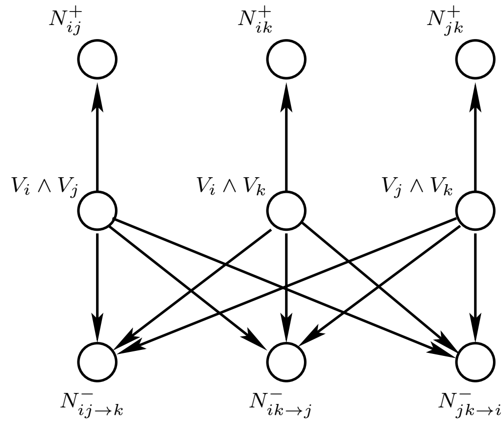


Figure 4.6: The Bayesian network for view triplet reasoning.

> global reasoning after insertion of a new image would undo the advantages of an incremental approach.

Since we have only three binary hidden variables for a view triplet, we can perform the probabilistic inference efficiently by explicit calculation of the posterior probabilities. Given the observation of the actual number of epipolar correspondences N^+_{ij} and the missing features $N^-_{ij\to k}$ for all permutations of i, j and k, we phrase the joint probability density *pdf* according to the belief network in Figure 4.6:

$$\begin{aligned} &pdf(\{V_i \wedge V_j\}, \{N^+\}, \{N^-\}) = \\ &\prod_{(i,j)\in\{(1,2)(1,3)(2,3)\}} P\left(V_i \wedge V_j\right) \\ &\prod_{(i,j)\in\{(1,2)(1,3)(2,3)\}} P\left(N^+_{ij} | V_i \wedge V_j\right) \\ &\prod_{(i,j)\in\{(1,2)(1,3)(2,3)\}} P\left(N^-_{ij\to k} | V_i \wedge V_j \wedge V_k\right). \end{aligned} \tag{4.1}$$

The posterior probabilities $P\left(\{V_i \wedge V_j\} | \{N^+\}, \{N^-\}\right)$ can be directly computed from the joint density *pdf*.

The posterior distribution provides additional information on the confidence of the most likely hypothesis. If all EGs in a view triplet are accepted (i.e. $V_i \wedge V_j \wedge V_k = 1$), then the ratio

$$\frac{P\left(V_i \wedge V_j \wedge V_k = 1 | \{N^+\}, \{N^-\}\right)}{\max P\left(V_i \wedge V_j = 0, V_i \wedge V_k, V_j \wedge V_k | \{N^+\}, \{N^-\}\right)}, \tag{4.2}$$

assesses the confidence in that decision with respect to a particular EG $V_i \wedge V_j$. We use the logarithm of that ratio as the actual confidence value for $V_i \wedge V_j$ with respect to the triplet (V_i, V_j, V_k). The overall confidence of an EG participating in several view triplets is the minimum of those confidences. These values are later used as edge weights of the camera adjacency graph during the generation of the individual reconstructions (see Section 4.3).

4.2.2 Choice of Prior Probabilities

Basically, each of the latent variables $V_i \wedge V_j$, $V_i \wedge V_k$ and $V_j \wedge V_k$ can take either 0 or 1, resulting in 8 possible configurations. It turns out, that only four of those configurations are plausible: the case, that all epipolar geometries are discarded can be excluded, since it needs a likely EG between two views to verify the third one using the proposed reasoning. Likewise, the case that exactly one EG is rejected, is not plausible either, since such configuration would require very specific camera poses and image content. Consequently, these undesirable configuration can be easily excluded by setting the prior probability of those cases to zero. We consider the remaining four configurations as equally likely.

Positive Support from Pairwise Correspondences

This section describes the utilized choice for the prior probabilities. We employ point features to establish correspondences between images. In particular, DoG points with associated SIFT feature vectors [Lowe, 2004] are extracted from the supplied images. Let N_i and N_j denote the number of feature points detected in view V_i and V_j, respectively. If we presume, that (V_i, V_j) forms a visually related image pair (in terms of epipolar geometry), one can expect to recover a certain fraction of the features in V_i and V_j as correspondences. In order to obtain a symmetric model, we merge the features from V_i and V_j yielding $N_i + N_j$ items. The expected number of correspondences N_{ij}^+ is now "close" to $(N_i + N_j)/2$ (since one correspondence represents two detected features). Of course, it is unlikely to find correspondences for all feature points. The repeatability of the feature point detector, image content overlap, occlusions due to the scene structure, perspective distortion etc. influences the number of recovered correspondences. We simply accumulate these effects into one probability p_1, which is the likelihood of regaining a feature extracted in one view in the other view under the assumption $V_i \wedge V_j = 1$. Hence, recovering N_{ij}^+ correspondences from $N_i + N_j$ features points is modeled by a binomial distribution with parameters $N_{ij} := (N_i + N_j)/2$ and p_1:

$$N_{ij}^+ \sim B(N_{ij}, p_1) \ \text{ if } V_i \wedge V_j = 1.$$

If the two images V_i and V_j are visually unrelated (i.e. $V_i \wedge V_j = 0$), finding correspondences is just coincidental. We denote the probability of finding an incidental correspondence by p_0. Again, the observed number of correspondences in this case can be approximately modeled using a binomial distribution, but now with a much lower success probability $p_0 \ll p_1$:

$$N_{ij}^+ \sim B(N_{ij}, p_0) \ \text{ if } V_i \wedge V_j = 0.$$

Figure 4.7 shows the binomial probability density function $N_{ij} \sim B(N_{ij}, p_1)$ with parameters $N_{ij} = \{100, 200, 300, 400, 500, 600\}$ and $p_1 = 0.1$.

Negative Belief from Missing Correspondences

This section addresses the estimation of $P(C_{ij\to k}^- | V_i \wedge V_j)$. Note, that the role of the single views in a triplet is not symmetric: $P(C_{ij\to k}^- | V_i \wedge V_j)$ is different from $P(C_{ik\to j}^- | V_i \wedge V_k)$ and

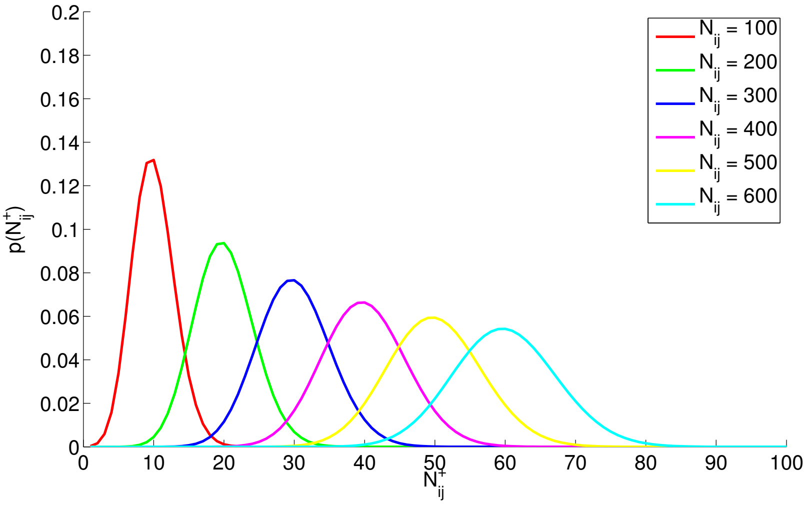


Figure 4.7: Binomial probability density function $N_{ij} \sim B(N_{ij}, p_1)$ with parameters $N_{ij} = \{100, 200, 300, 400, 500, 600\}$ and $p_1 = 0.1$.

$P(C^-_{jk \to i} | V_j \wedge V_k)$.

Let C^+_{ij} denote the correspondences between view i and j, and let N_{ij} be the number of triangulated points from C^+_{ij} lying inside the view frustum of V_k (i.e. actually visible in view V_k). Furthermore, N_{ijk} is the number of inlier correspondences across the whole triplet. As in the pairwise case, we expect N_{ijk} not to be much smaller than N_{ij}, if $V_i \wedge V_j \wedge V_k = 1$. One might use a binomial distribution again as described in the previous section for view pairs. But note that the considered view triplets already has some support from the correspondences over all three views, i.e. some image content is common in all three views. Hence, the binomial distribution parameters q_1 (in the case $V_i \wedge V_j \wedge V_k = 1$) and q_0 (if $V_i \wedge V_j \wedge V_k = 0$) are less distinct than the values p_1 and p_0 used for the pair prior, and the appropriate choice is rather critical. Therefore, we approximate the distribution of $N^-_{ij \to k}$ by a Poisson distribution:

$$N^-_{ij \to k} \sim Pois(\lambda_1) \;\text{ if } V_i \wedge V_j \wedge V_k = 1 \tag{4.3}$$

$$N^-_{ij \to k} \sim Pois(\lambda_0) \;\text{ if } V_i \wedge V_j \wedge V_k = 0, \tag{4.4}$$

with $\lambda_1 \ll \lambda_0$.

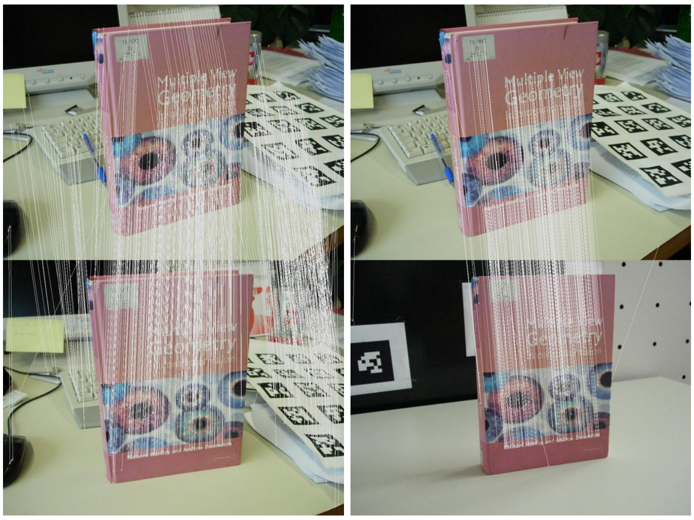
(or alternatively $N_{ijk} \sim B(N_{ij}, 1 - q_1)$). On the other hand, if any of the assumptions $V_i \wedge V_j$, $V_i \wedge V_k$ or $V_j \wedge V_k$ is not satisfied, finding any correspondences between V_i and V_j in the third view V_k is purely incidental, hence

$$N_{ij} - N_{ijk} \sim B(N_{ij}, q_0) \;\text{ if } V_i \wedge V_j \wedge V_k = 0, \tag{4.5}$$

for q_0 close to one.

4.2.3 Practical Considerations

In the discussion above we considered only the number of found or missing correspondences. The distribution of point features (or missing ones) in a particular image gives an additional cue about the prior probabilities. Consider two view pairs as depicted in Figure 4.8. The first image pair in Figure 4.8(a) (having a true underlying epipolar relation) not only has more correspondences, but these are better distributed over the image. Figure 4.8(b) shows the correspondences for a false epipolar view pair, where the correspondences are spatially concentrated on the "repetitive" scene content. A low number of found correspondences may indicate a false epipolar geometry or may be the result of little image structure. In order to partially disambiguate these possibilities, we replace the raw number of features by an *effective* quantity computed as described in Section 2.7.2.



(a) Good distribution (b) Concentrated distribution

Figure 4.8: Two correspondence distribution. In (a) the detected correspondences are better distributed over the image than in (b).

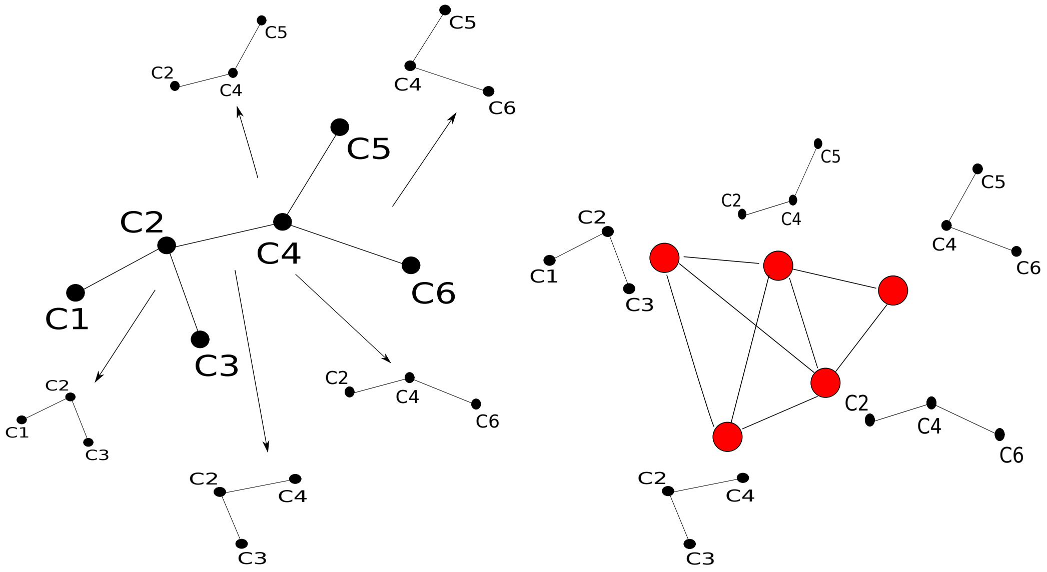
In practical experiments it turned out, that the relatively strong assumption on the detector repeatability yields to wrong rejections of true EGs. Particularly, the repeatability of the employed DoG points is low if there are substantial scale changes between the images. These incorrectly detected missing correspondences can be identified, since they are typically interspersed with found correspondences. Hence, missing correspondences are suppressed, if a found correspondence is spatially close.

4.3 Grouping of EGs

In the last section we described, how pairwise epipolar geometries are verified using a third view. It is not sufficient to collect those triplets containing only accepted pairs directly, since false epipolar geometries may still be included through undetected false epipolar pairs. Epipolar geometries are typically rejected only if there is a sufficiently strong indication for rejection. Hence, rejecting EGs is not a transitive operation.

The procedure to combine correct EGs is based on view triplets and performs several steps. First, all view triplets containing a rejected view pair are discarded. Afterwards, view triplets sharing a common view pair are collected to constitute individual reconstructions as shown in Figure 4.9(a). Next a graph $G_{\mathcal{T}}$ is constructed having view triplets as nodes. From this graph, connected components [Hopcroft and Tarjan, 1973] are extracted, each connected component will result in a 3D reconstruction. The concept is visualized in Figure 4.9(a).

Edges between nodes are present in this graph, if the respective view triplets have a view pair in common. Each of these resulting reconstructions can be easily registered into a common coordinate frame (see Section 4.4), but these reconstructions may still connect unrelated views. By adding a new image several new view triplets may be generated and the following procedure (and the steps outlined in Section 4.4) needs to be applied on the affected components.



(a) (b)

Figure 4.9: (a)*Triplet enumeration.* This example shows how a MST is used to enumerate image triplets. The MST consists of 6 cameras $C_1 ... C_6$ and 5 triplet candidates. (b) *Triplet Graph.* Each node in this graph represents a reconstructed image triplet. Common views of the reconstructed image triplets are used to determine the connectivity in this trifocal graph $G_{\mathcal{T}}$.

A reconstruction containing the views $V_1, \ldots, V_n$ naturally induces an undirected camera adjacency graph with the edges being the verified two-view geometries (e.g. [Steele and Egbert, 2006]). We designate such a graph as *consistent*, if there is no path from V_i to V_j for any rejected EG between V_i and V_j, i.e. $V_i \wedge V_j = 0$ implies that V_i and V_j are in different components. This definition of consistency is quite conservative, since views with incorrect EGs might be correctly part of the same reconstruction. In the current framework this definition of consistency potentially results in too many small individual reconstructions, but none of these will include a rejected EG.

Splitting an inconsistent graph into several consistent subgraphs is performed using a modified version of Kruskal's algorithm for minimum spanning tree computation. Essentially, we extent the test for cycle prevention with additional checks for consistency. Two disjoint sets (i.e. individual reconstructions) are not merged, if this would yield an inconsistent tree (see Algorithm 3). The employed edge weights are just the EG confidence values calculated from the posterior probabilities (recall Section 4.2), thus highly reliable view pairs are merged first. Of course, our algorithm delivers a forest instead of a spanning tree.

Algorithm 3: Modified Kruskal's method

```
Procedure   $F$ = Modified MST
Input:   A potentially inconsistent weighted graph $G = (V, E)$
  $F := \emptyset$; $\forall i$ : MAKE-SET$(DS, i)$
  for each edge $(i, j) \in E$ in order of nonincreasing weight do
    $r_i$ ←FIND-SET$(DS, i)$; $r_j$ ←FIND-SET$(DS, j)$
    if  $r_i \neq r_j$ and
        $\forall k \in$ SET$(DS, i)$, $\forall l \in$ SET$(DS, j)$: $V_k \wedge V_l = 1$ then
      UNION-SET$(DS, i, j)$
      $F \leftarrow F \cup (i, j)$
    end if
  end for
```

4.4 3D Structure Extraction

After the verified pairwise epipolar geometries are collected into a set of consistent reconstructions, the initial structure and motion remains to be determined. Currently, we follow an approach inspired by [Martinec and Pajdla, 2007] to obtain the extrinsic camera parameters, which relies only on the robustly estimated relative poses. If a newly added image does not change the topology of the EGs (i.e. two or more reconstructions are not merged and no additional EG is rejected), an initial estimate of its pose and respective 3D points can be immediately determined (e.g. by perspective pose computation). In all other cases the structure and motion parameters of affected individual reconstructions are determined as follows:

First, the given relative rotations $\{R^{ij}\}$ between two views are upgraded into a consistent set of rotations $\{R_i\}$ by solving the overdetermined system of equations, $R_j = R^{ij}\, R_i$ [Govindu, 2004]. As described in [Martinec and Pajdla, 2007] we solve the system initially for approximate rotation matrices and subsequently enforce the orthonormality of R_i using the SVD. Implementation details are discussed in Section 2.8.1. The registered translations are computed using a two-step procedure in order to always obtain physically meaningful results. At first, the global scales are determined using a linear approach. Separating scale and translation estimation has the advantage, that positive scales can be easily enforced.

With the knowledge of the registered rotations and scales, the coordinate frames of view triplets differ only by translational offsets, which are determined linearly as well. Algebraic least squares solutions for the offsets and the camera centers are obtained using the respective normal equation. The initial 3D structure is created by linear triangulation of the inlier correspondences. Finally, a metric bundle adjustment is applied (see Section 2.9).

At first, the scales of all triplets are linearly determined: if two view triplets k and l with common views i and j are given, the global scales s^k and s^l fulfill,

$$s_k \| C_i^k - C_j^k \| - s_l \| C_i^l - C_j^l \| = 0, \tag{4.6}$$

where $C_{(\cdot)}^{(\cdot)}$ are the camera centers in the coordinate frame of the respective view triplet. The resulting equation system for the $s_{(\cdot)}$ is large, but sparse and can be solved by the SVD.

With the knowledge of the registered rotations and scales, the coordinate frames of view triplets differ only by an translational offset (denoted by T^k for view triplet k). Hence, the registered camera centers $C_{(\cdot)}$ and the translational offsets must satisfy,

$$C_i = C_i^k + T^k \quad \text{for all cameras } i \text{ and triplets } k. \tag{4.7}$$

Algebraic least squares solutions for $C_{(\cdot)}$ and $T^{(\cdot)}$ are obtained using the respective normal equation. Since the x, y and z-components are independent, the size of the equation system can be substantially reduced by solving for each component separately.

4.5 Experimental Results

This section provides additional real-world examples, where incorporating the proposed approach employing missing correspondences results in substantially enhanced reconstructions. In these experiments the basic probability parameters $(p_0, p_1, \lambda_0, \lambda_1)$ are set to $(0.001, 0.1, 0.95, 0.2)$, respectively. Figures 4.11(a) and (b) depict example views of highly similar, but nevertheless different facades. Without the incorporation of our proposed method all views are incorrectly combined into one common reconstruction. Enabling the rejection of two view geometries results in splitted 3D models as illustrated in Figure 4.11(c) and (d). The second example is an indoor environment with two very similar fire extinguishers appearing in the images (Figure 4.12(a)–(c)). This common object acts as an "visual anchor" linking all views into a common frame (Figure 4.12(d)). Separation of individual reconstructions is not perfect in this case, since a few images actually belonging to scenery depicted in Figure 4.12(c) are attached to the middle one. This example shows, that a sufficient number of absent features is required for perfect reasoning.

Note that the purpose of these examples is to provide evidence, that incorrect EGs can be detected and handled even with very limited and visually misleading image data.

4.6 Conclusion and Discussion

In this chapter we proposed an approach for structure from motion computation that is able to detect incorrect two view geometries by reasoning about missing correspondences retrieved from

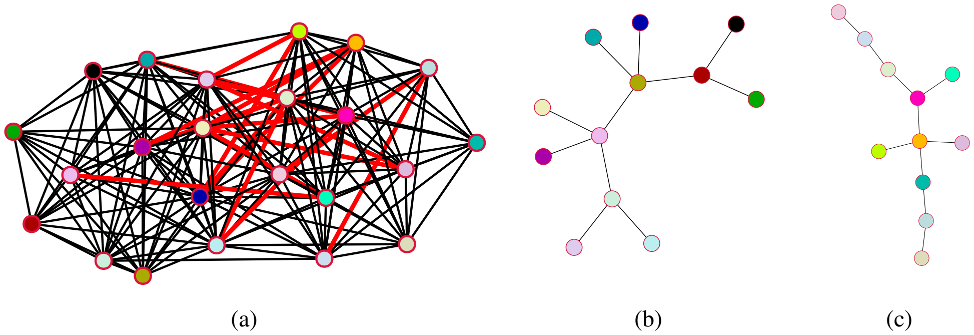
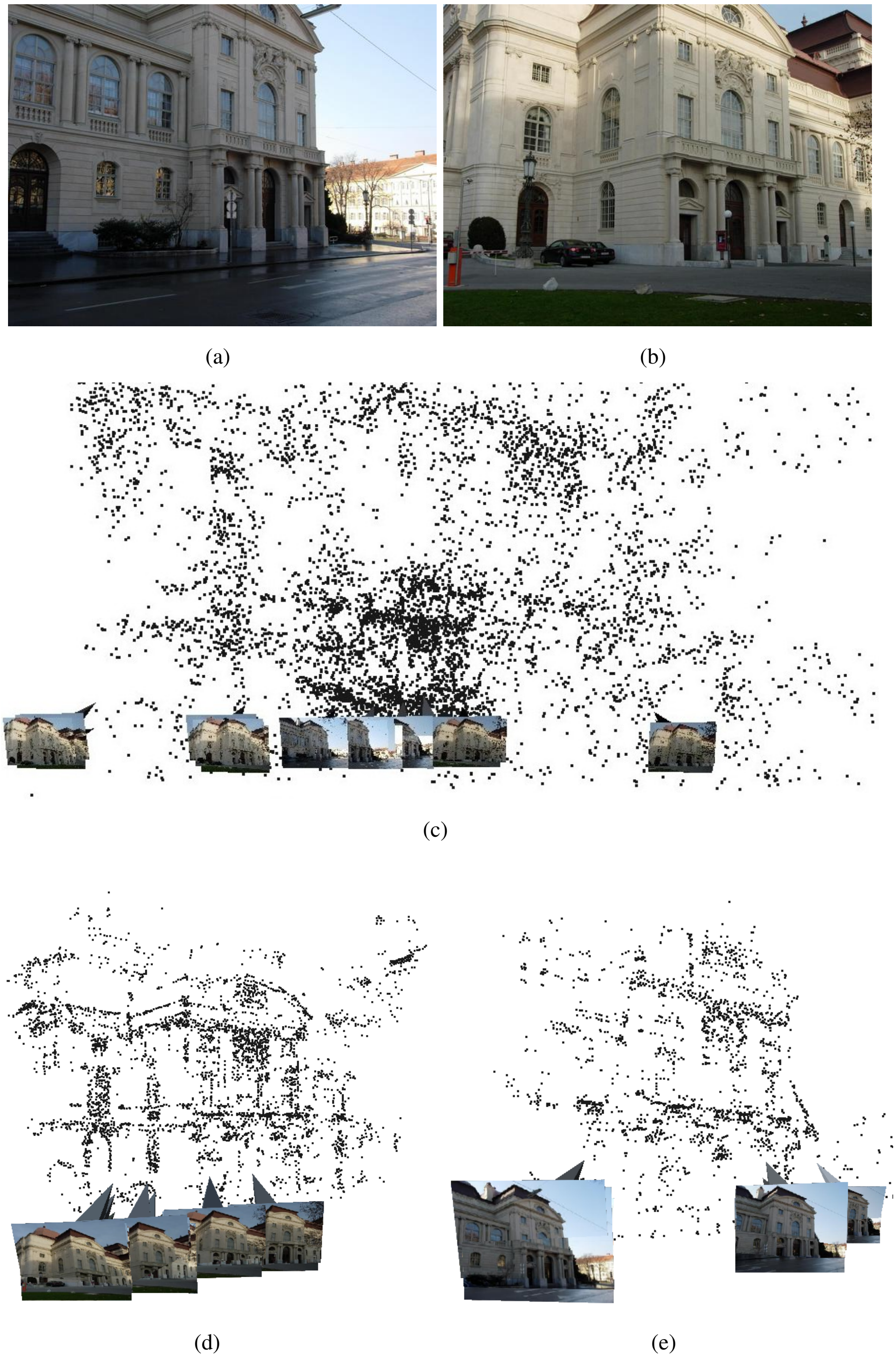


Figure 4.10: (a) Epipolar Graph with respect to the Opera image collections. Red edges are rejected epipolar geometries from Bayesian Reasoning. (b), (c) extracted connected components from the triplet graph $G_{\mathcal{T}}$.

view triplets. Such wrong two view epipolar geometries often occur in scenes with duplicate structure instances, such as man made objects and buildings. Traditional reconstruction pipelines that only perform monotone reasoning on the epipolar geometry are often not able to handle/detect such cases and lead to wrong reconstructions like folded/duplicate structures. Our algorithm is able to infer and remove such erroneous match pairs. The method can be used to augment existing 3D reconstruction systems with little additional costs. Furthermore, our proposed approach naturally fits into incremental systems for online 3D reconstruction. Recent work in structure from motion for duplicate scenes [Zach et al., 2010, Roberts et al., 2011] demonstrate that for scenes with large duplicate structures pure geometric reasoning alone is often not sufficient to disambiguate between multiple hypotheses. In [Roberts et al., 2011] the missing correspondence cue is combined with an image timestamp cue. While the first cue performs geometric reasoning, the second cue relies on causality. If a single photographer captures a scene, approximate sequence information is provided which means that pairwise matches relatively close in time are less likely to be erroneous than those far in time.

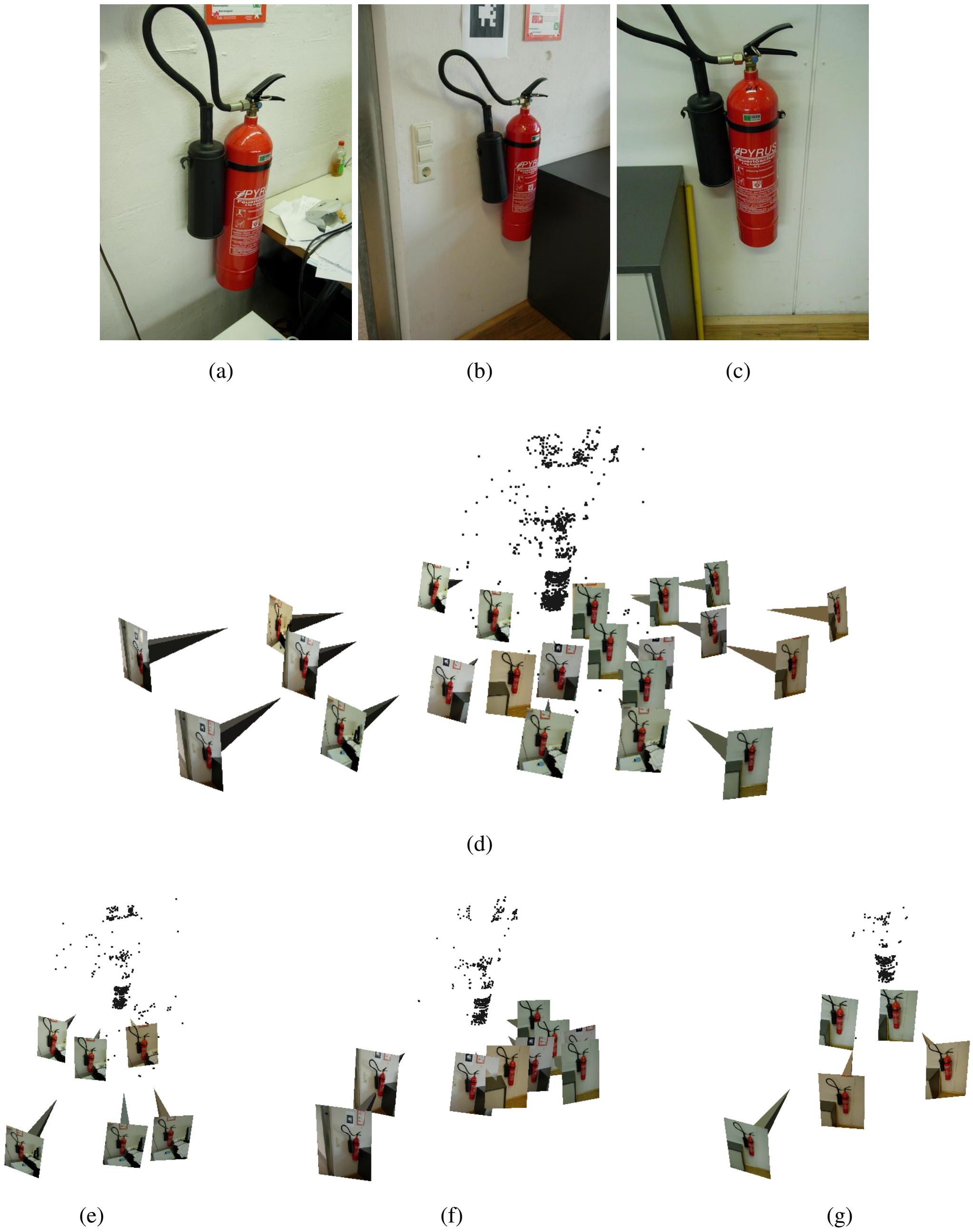


(a) (b)

(c)

(d) (e)

Figure 4.11: (a)–(b) Source images showing symmetric facades representing opposite sides of the building (23 in total). Small panoramas are used to capture the full height. (c) The 3D reconstruction obtained without EG verification incorrectly merges all views. (d)–(e) Two separate reconstructions obtained by our proposed approach.



(a) (b) (c)

(d)

(e) (f) (g)

Figure 4.12: (a)–(c) Source images showing the same kind of fire extinguisher at different places (out of 24). (d) Incorrectly fused result of structure and motion without EG verification. (e)–(g) The three individual components obtained by our method. The separation between (f) and (g) is not perfect due to insufficient background structure.

Chapter 5

Image based Localization

Despite tremendous progress in image retrieval [Nistér and Stewenius, 2006, Chum et al., 2009] and matching [Sivic and Zisserman, 2003], the demand for image based location recognition has not been satisfied. Our proposed approach to this problem leverages the recent progress of 3D scene reconstruction from images/videos [Pollefeys et al., 2008, Snavely et al., 2006, Li et al., 2008, Agarwal et al., 2009, Frahm et al., 2010] for building a superior location recognition system. Both research areas have independently made enormous progress in the last decade. Our proposed approach employs the fact that the obtained 3D models allow to impose stronger geometric constraints on possible scene views than traditional image based methods. These geometric constraints are mostly orthogonal to the image based constraints and deliver the pose for the query image directly. Accordingly, we can also utilize the significant progress in image based recognition that occurred over the past decade leading to near real-time image retrieval [Nistér and Stewenius, 2006] from huge databases containing millions of images. Our proposed approach combines these two disciplines and uses their state-of-the-art techniques to advance location recognition. The core component of our system is a compressed scene representation that consists of a representative set of 3D point fragments that cover a 3D scene from arbitrary view points. Additionally, we build upon recent advances in image retrieval and use a vocabulary tree data structure as described in Section 2.5.1 for fast feature indexing. A subsequent matching approach and geometric verification directly delivers the pose of the current query image with respect to the reconstructed 3D models. Our proposed workflow for efficient view registration is shown in Figure 5.1.

Previous approaches for view registration where either based on matching an input image to a small 3D point cloud [Skrypnyk and Lowe, 2004b] or to a large set of geo-registered reference images [Schindler et al., 2007a]. However, matching a query image to a large set of millions of 3D features is not feasible because of computation time (see Table 2.2) and the high background noise that leads to many ambiguous matches. On the other hand, location recognition approaches using image retrieval techniques are fast and efficient but require that database images are sufficiently similar to the query view [Sivic and Zisserman, 2003]. While localization of images that are sufficiently close to the original ones work well, matching of images from viewpoints that are beyond original camera positions is more challenging. To overcome this limitation, we introduce

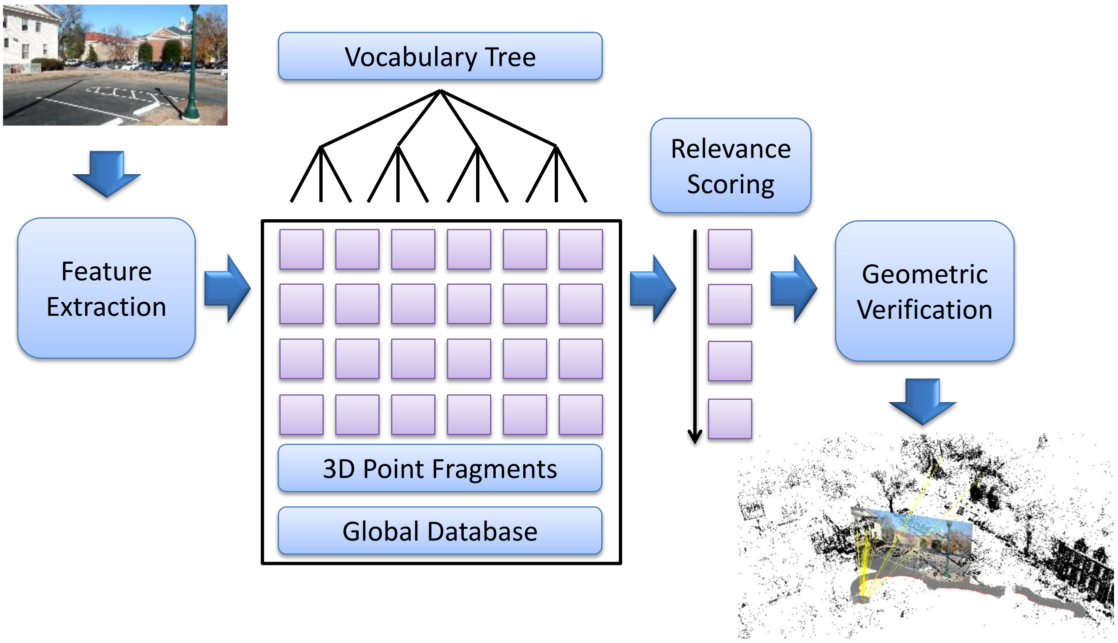


Figure 5.1: Registration of video frames with respect to a sparse 3D scene, reconstructed by structure from motion techniques. Each input image is individually matched and registered to a global database of multiple 3D models.

the concept of synthetic views that additionally partition the 3D model into small fragments of 3D points as described in Section 5.1.2. This strategy allows efficient registration of images of significantly different viewpoints than the original views used for model reconstruction. Additionally, we propose a compressed 3D scene representation [Simon et al., 2007] which improves recognition rates while simultaneously reducing the computation time and the memory consumption. The design of our method is based on algorithms that efficiently utilize modern graphics processing units to deliver real-time performance for view registration. Our algorithm is scalable and runs at 15 fps on current GPU accelerated desktop computers. In Chapter 6 three different applications of the proposed approach are presented.

5.1 3D Scene Representation

This section describes the compact representation of a 3D model (or a set of models) that we use to register new query images. Naturally, the underlying 3D models are created from images using multiple-view vision methods as described in the previous chapters. A set of images registered to the 3D model is always required in order to retrieve the necessary image features and associated descriptors for the 3D points of the model. Since we employ point features in the query image, only a sparse point cloud with associated 3D descriptors is required. Hence, we can omit the costly dense geometry generation for our purpose.

5.1.1 Visual Landmark

In order to generate a 3D visual landmark with associated feature descriptors, any robust multi view reconstruction method can be used. In addition to texture information from the images, structure from motion techniques provide sparse 3D geometry that is used to determine an exact 6DOF pose from 2D-to-3D point correspondences. We utilize the very effective SIFT keypoint detector and descriptor [Lowe, 2004] as the primary tool to represent point features. Note, our approach is not restricted to SIFT, any other robust scale invariant feature detector/descriptor would also be suitable. Every 3D point $\mathbf{X}$ in the resulting sparse model has a set of associated image features $\mathbf{d} \in \mathcal{D}$. In addition every reconstructed 3D point has an associated scale S induced by the keypoint detector. Under the fronto-parallel surface assumption, the scale s found in the image can be extrapolated to a global 3D scale,

$$S = \frac{sd}{f} \tag{5.1}$$

where f is the focal length of the camera and d the point depth. In practice an average scale over all projections is used to describe the size of a 3D point. Furthermore, each descriptors has a directional component $\mathbf{v}$, that corresponds to the view vector of the respective camera. A 3D point therefore is represented by,

$$\mathbf{X} = \{< x, y, z >, S, \{< \mathbf{d}_1, \mathbf{v}_1 > ... < \mathbf{d}_n, \mathbf{v}_n >\}\} \tag{5.2}$$

where $< x, y, z >$ is the 3D location of the feature point, S the scale and $\{< \mathbf{d}_1, \mathbf{v}_1 > ... < \mathbf{d}_n, \mathbf{v}_n >\}$ the set of view dependent descriptors. An illustrative example of a 3D feature point is shown in Figure 5.2. The list of descriptors can be very long for highly stable 3D points, i.e. points visible and matchable in many source images. Figure 5.3 depicts image patches (corresponding to SIFT descriptors) that belong to the same triangulated 3D point. Typically, the descriptor list $\mathcal{D}$ for such points shows high redundancy and can be compressed to a small codebook $\mathcal{P} \subset \mathcal{D}$ without loss in registration performance. The objective here is to vector quantize the descriptors into a reduced set of clusters that represent the scene. Therefore, lowering the memory footprint of the 3D representation. Furthermore, the reduced number of descriptors makes feature matching more efficient. In particular we apply mean-shift clustering [Comaniciu and Meer, 2002] to quantize SIFT descriptors belonging to each 3D point, though other methods (K-medoids, histogram binning, etc) are certainly possible, too. Figure 5.3 depicts image patches (corresponding to SIFT descriptors) that belong to the same triangulated 3D point and the respective clustering result produced by the Mean-shift algorithm. Mean-shift clustering enables to set a global threshold h (bandwidth) on the maximally allowed inter-cluster dissimilarity $2h$. Hence, if two feature descriptors have a distance d before mean-shift clustering, the distance of the cluster centers is at most $d + 2h$. Figure 5.3 shows image patches of respective SIFT descriptor and the grouping after mean-shift clustering.

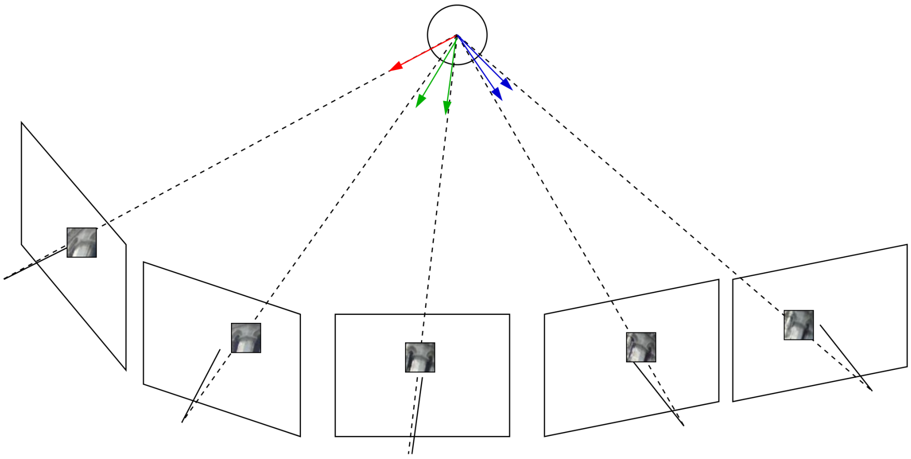


Figure 5.2: 3D point reconstructed from multiple views. Each measurement corresponds to an image patch associated to the region where the feature point was extracted from. Under the assumption of fronto-parallel surfaces, the 2D feature scales are extrapolated to a mean 3D scale.

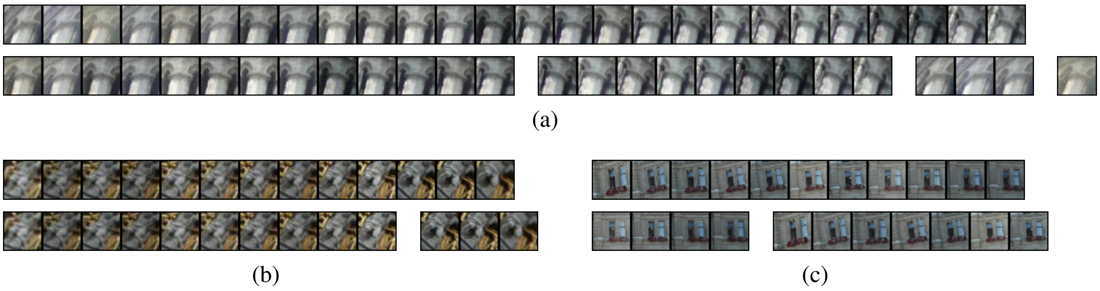


Figure 5.3: (a)-(c) The first row of each figure shows image patches belonging to the same triangulated 3D point (track). The second row depicts the grouping result after mean-shift clustering (bandwidth $h = 0.22$). For track (a) 26 SIFT descriptors are reduced to 4 clusters (26/4), in (b) 13/2 and (c) 11/2.

5.1.2 Synthetic Views

As described in the previous section, the reconstructed model is represented as a 3D point cloud providing associated scale values and feature descriptors. In addition, the set of images used to build the model with known orientation is available. The views used for 3D reconstruction already provide a natural partition of the 3D model into compact 3D point fragments. While this information allows registration of views that are close to original ones (in terms of orientation, resolution and field of view) it is not suitable for matching images with considerable viewpoint change. For instance, consider the two illustrative examples shown in Figure 5.4. In (a) several 3D features of original structure from motion images (black) have to be combined in order to determine the pose of the blue camera. This is in contrast to the example shown in (b), where only a subset of 3D points is required for registration.

To overcome this limitations, we introduce “synthetic” views located at positions not covered

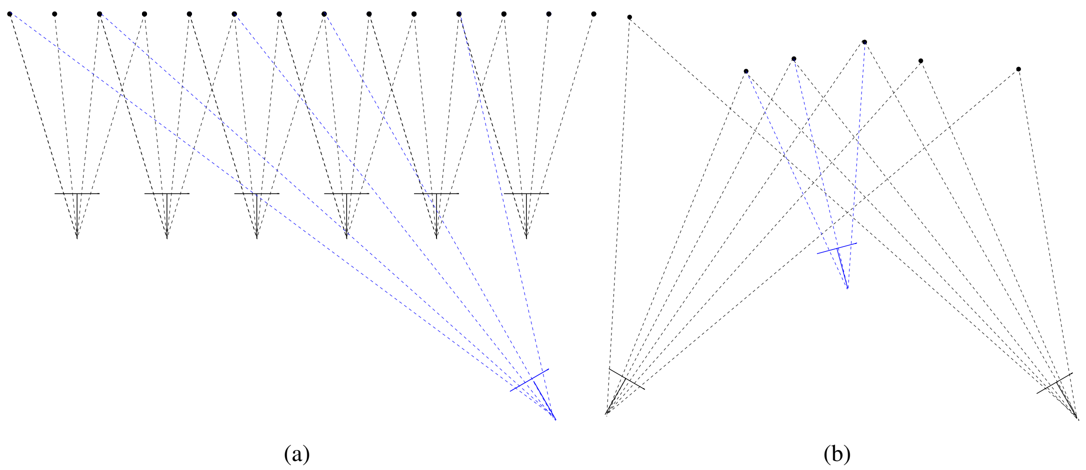


Figure 5.4: The 3D model representation obtained by the original views (black) may not cover sufficiently the field-of-view of new inserted cameras (blue). (a) View registration requiring a combination of features from multiple original views (views from model reconstruction). (b) Registration can be done by utilizing only a subset of features extracted from original views.

by the original images that provide an additional partitioning of the 3D scene points into manageable descriptor sets that are suitable for matching. Furthermore, the synthetic views are chosen in order to match the resolution and field of view of the target camera. For instance a full panoramic camera offering 360° field of view at location X would observe a different set of features than a standard digital camera at the same position with 60° field of view. On the other hand, the image resolution determines the minimal and maximal scale of potentially visible features. For the moment we assume that we have a camera with fixed field of view and resolution. An image taken at distance d to a given 3D scene will capture different features than an image taken at distance $2d$. This concept is illustrated in Figure 5.5, e.g. due to the limited resolution, the blue camera can only detect 3D points at a given minimal scale. An example of synthetic views and respective 3D point fragments of a real world scene is shown in Figure 5.6.

Our application is targeted towards localization in urban environments. Hence we can restrict the placement of synthetic cameras to the "eye-level" plane induced by the original views to simplify the problem. Generally, our approach is not limited to terrestrial camera positions. A more powerful descriptor like the VIP features [Wu et al., 2008] might prove beneficial for registering images captured from significantly different viewpoints (e.g. aerial views).

For terrestrial localization, it is sufficient to place synthetic cameras uniformly on this plane and at this point we do not consider any optimal placement strategy (like proposed in [Chen and Li, 2004]). Depending on the application, the grid size can be adapted to meet the desired target accuracy. At each grid position view directions are sampled on the unit sphere. We use 12 directions for the camera rotations. This corresponds to a 30° rotation between the cameras. The 30° are approximately the off image plane rotation that the SIFT descriptor is robust against [Mikolajczyk et al., 2005]. Figure 5.7 depicts the placement of synthetic views into a structure from

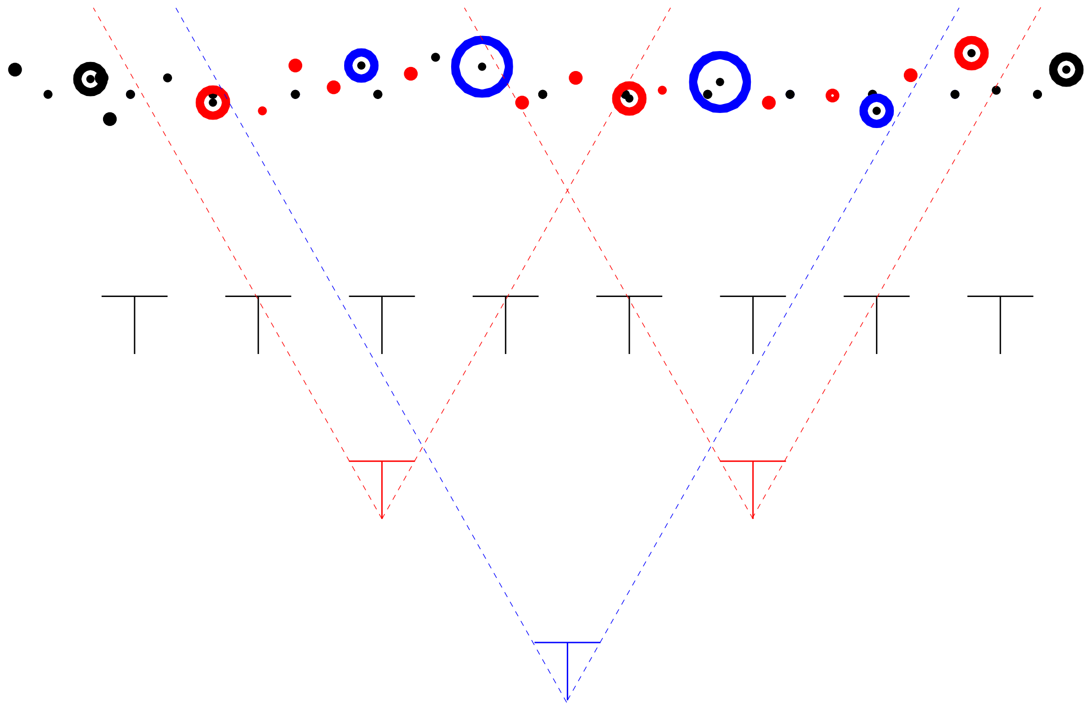


Figure 5.5: Synthetic view concept. Varying camera positions capture different sets of 3D points with respect to scale and orientation.

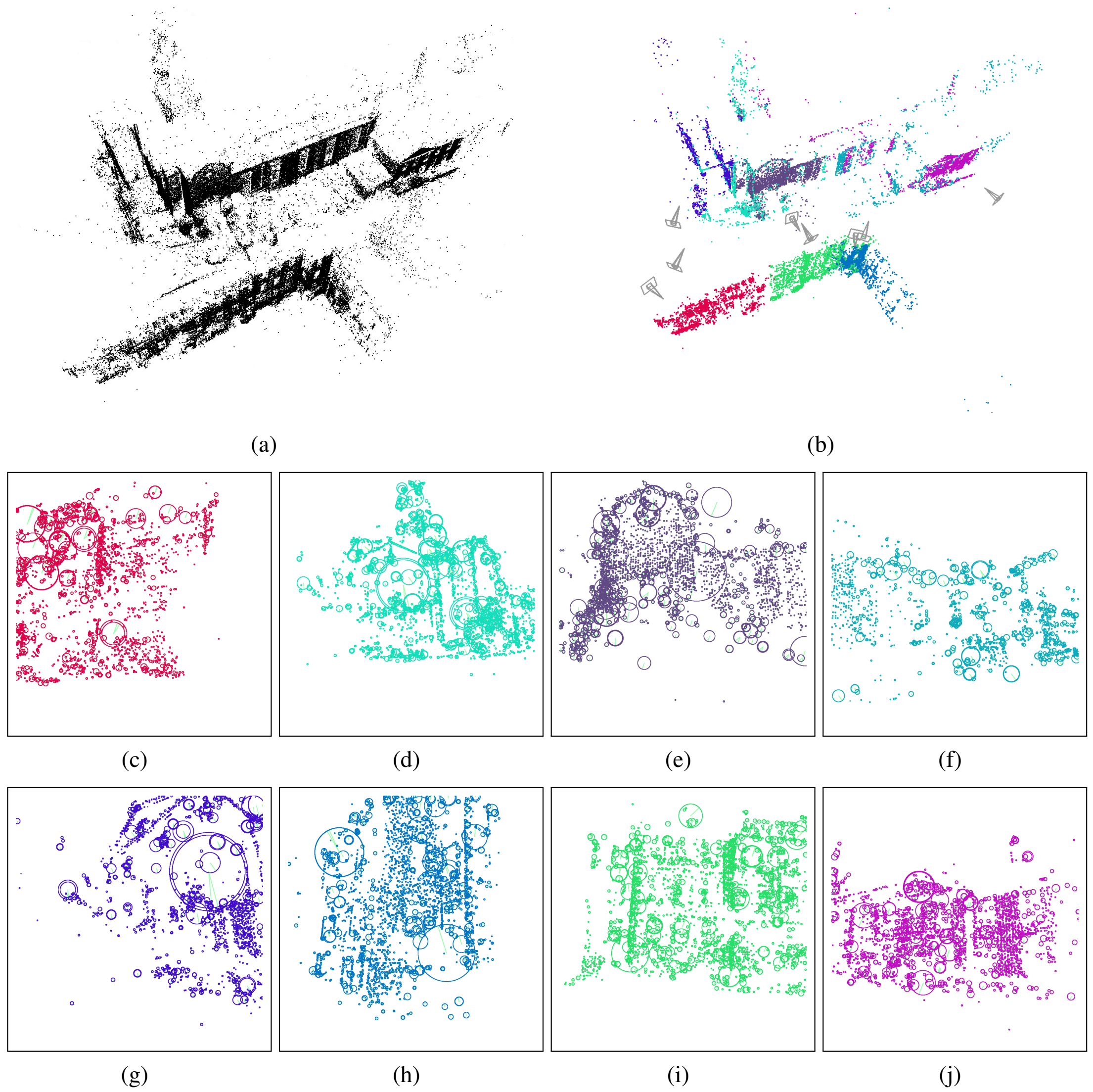
motion reconstruction of a square. In-plane camera rotations are largely handled by the rotational invariance of the SIFT descriptor. The intrinsic parameters for the synthetic views are empirically set to a field of view α and $m \times n$ pixel resolution. Not all generated synthetic views are really useful. Given the 3D position and the respective scale of each triangulated point in the sparse model, one can estimate the projected feature size in the synthetic images and therefore infer the visibility of each 3D point given the set of visible features. More precisely, a 3D point is potentially visible in a synthetic view, if the following criteria is satisfied:

1) **Visibility:** The projected feature must be in front of the camera and lie within the viewing frustum.

2) **Scale:** The scale of the projected 3D feature must be larger or equal to one pixel in terms of the respective DoG scale space extrema to ensure detectability;

3) **Viewing Angle:** One of the associated descriptors is extracted from an original image with a sufficiently similar viewing direction due to the limited repeatability of the SIFT descriptor under perspective distortion [Mikolajczyk et al., 2005].

For the viewing angle criterion we set the angle threshold to 30°, which again corresponds to the stability region of the SIFT descriptor. This criterion acts as a "face culling" test by removing 3D points oriented away from the synthetic camera. As can be seen in Figure 5.6, there is a one-to-one correspondence between synthetically generated views and the 3D points visible therein.



(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 5.6: (a) Sparse reconstruction of the church scene and (b) shows a subset of 8 out of 500 3D documents (synthetic views).(c)-(j) Feature sets of respective 3D point fragments visible in each synthetic view.

The set of 3D points (potentially) visible in a particular synthetic view represents the document later retrieved through the vocabulary tree search which is used in the subsequent 2D-3D point correspondence estimation. Analogously, the 3D points triangulated in the original images form "3D documents" with respect to the original views. In general the created synthetic views will have a high degree of redundancy especially given the fact that the original views additionally sample the scene. In the next section we will discuss a technique to perform a compression of these views into a representative subset of views covering the scene.

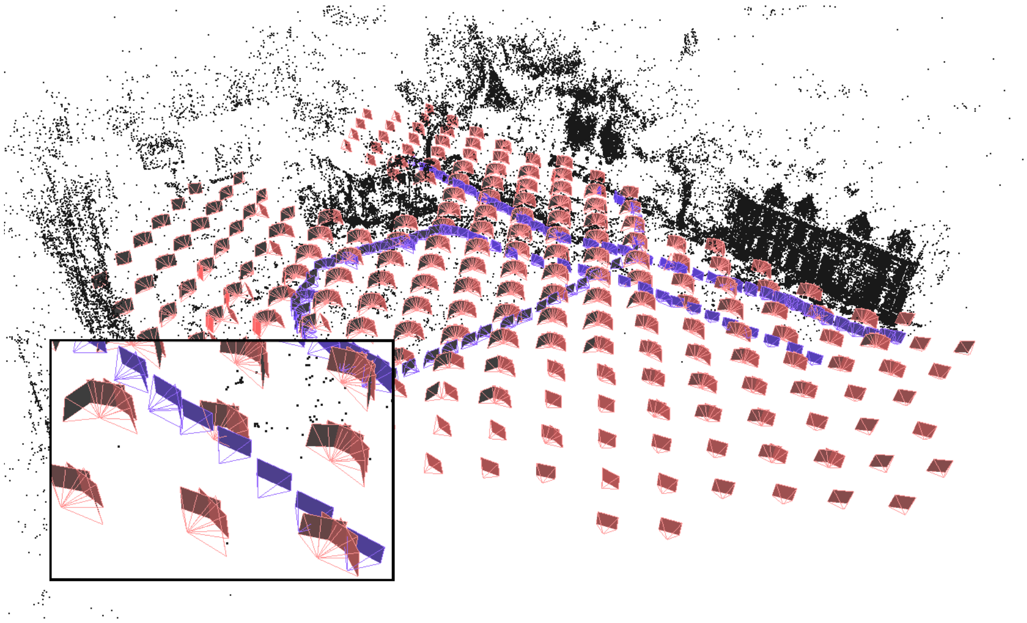


Figure 5.7: Structure from motion point cloud and the raw views/documents (blue camera glyphs for real images and red ones for the full set of synthetic views).

5.1.3 Compressed Scene Representation

The aim of our compression procedure is to build a compact as well as efficient 3D document database. A reduced set of documents has two major advantages over utilizing the full set of real and synthetic views:

(i) the signal-to-noise ratio for vocabulary tree queries (see Section 5.2.1) is increased, since it is expected that a reduced document set is more discriminative for their respective scene content;

(ii) the smaller database size has a positive impact on the run-time efficiency in general.

We take a different approach than [Schindler et al., 2007a], where visual words voting for a particular document in the vocabulary tree also support documents associated with spatially close views. The overall goal of our proposed compression strategy is to keep a minimal number of documents while still ensuring a high probability for successful registration of new images. Thus, the key question in evaluating a document summarization is, whether a particular set of documents is sufficient to determine the pose of admissible images. In order to reduce the computational complexity of determining a representative document set, we only consider views which are subsets of real and synthetic views. Thus, we do not create new 3D documents during the compression process. In the following we state these objectives more precisely.

Let V be an admissible view. The sparse 3D model projects into this view as a set of putatively visible 2D point features with associated descriptors. Under the assumptions for the image resolution (see Section 5.1.2), only a fraction of 3D points is estimated to be visible due to the corresponding scale of the features. 3D points with a too small scale in their projection will be discarded besides the features that are not within the field of view of the camera. We choose not to perform additional visibility reasoning, since small occluded points are already removed and for prominent occluded features our method does not provide the required precision of occlusion prediction. Accordingly we still associate them with the generated view. We assume that a view V can be successfully registered by a set of 3D points $\mathcal{P}$, if a certain number of 3D points from $\mathcal{P}$ is visible in V and has a good spatial distribution in the image.

We weight the raw number of features (or correspondences) by an estimate for the covered image fraction yielding an effective feature/correspondences count. This weighting is utilized for determining the effective number of correspondences for view registration (Section 5.2), too. In other words, the effective number of projected features must be larger than a specified threshold. For the document reduction procedure we require n effective 3D points from $\mathcal{P}$ to be visible in V (according to the above-mentioned assumptions on feature repeatability). Currently we use a rather conservative value of $n = 150$ for this threshold. For given sets of 3D documents and views a binary matrix can be constructed, which has an entry equal to one, if the respective document covers the particular view, and zero otherwise. Since in our setting the 3D documents correspond to combined (real and synthetic) views, this matrix is square. In order to have every view covered by at least one document, a document covers its corresponding view by default. This situation can arise if a particular real image has only a few extracted features and thus only a few triangulated 3D points are visible at all. The objective is now to determine a subset of the documents, such that every view is still covered by at least one 3D document. This is an instance of the set cover problem, one of the earliest problems known to be NP-complete [Karp, 1972b]. We use a straightforward greedy approach [Johnson, 1974] to determine a reduced but representative subset of documents with low time complexity. Algorithm 4 illustrates the greedy algorithm for a given binary view cover matrix A. The actual implementation employs a sparse, set-based representation for A. Figure 5.7 depicts synthetic camera positions and the respective compressed scene representation after applying our proposed approach is shown in Figure 5.8.

Algorithm 4: Greedy Set Cover

```
Input: Binary matrix $A \in \{0,1\}^{n \times n}$
Output: $\mathcal{S} \subseteq \{1, \ldots, M\}$

$\mathcal{S} \leftarrow \emptyset$
while $A \neq \mathbf{0}$ do
    $i^* \leftarrow \arg\max_i \sum_j A_{i,j}$
    $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{i^*\}$
    $A_{i,:} \leftarrow \max(0, A_{i,:} - A_{i^*,j})$ for all $i$
end
```

Algorithm 4 delivers a representative subset of views needed to cover the 3D scene. This subset can now be deployed for an efficient recognition of the scene context. The next section will describe our search method.

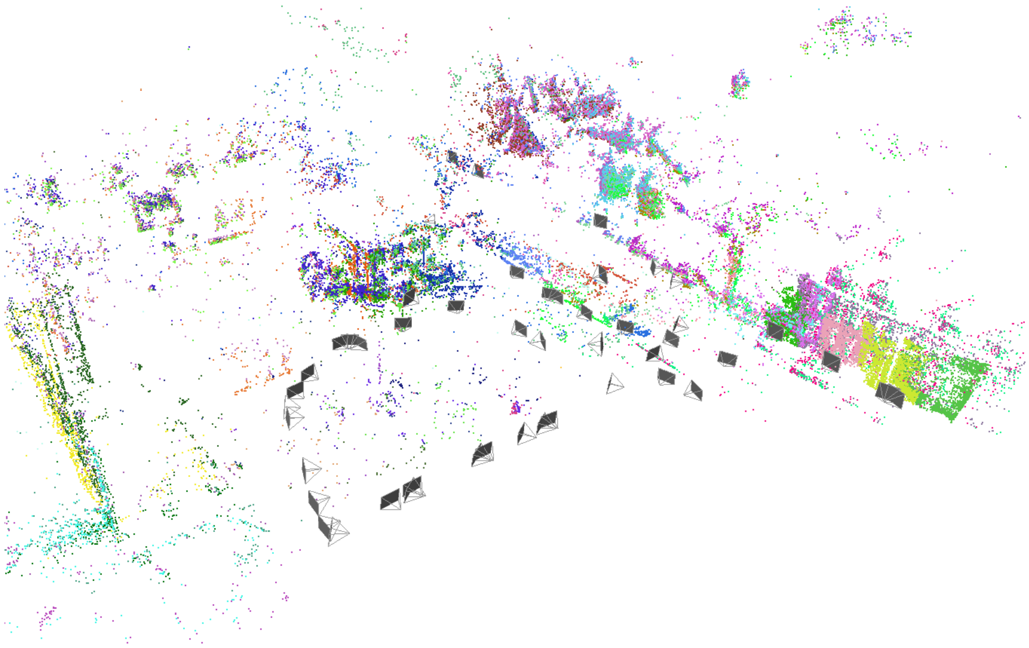


Figure 5.8: Compressed view/document set of Figure 5.7 with the color coding indicating the associated 3D points.

5.2 View Registration

Geometric registration of an incoming query image $\mathcal{Q}$ to the existing 3D database involves finding potentially matching relevant documents, for which we employ a vocabulary tree (Section 2.5) with a subsequent geometric verification (Section 2.7). This verification step simultaneously validates the putative matches and determines the pose of the query image with respect to the 3D model. If maximal run-time performance is targeted, 3D document retrieval needs to be very precise in order to avoid costly geometric verification of irrelevant documents. Thus, we use the probabilistic scoring function as described in Section 2.5.5 to rank documents according to the raw votes obtained by the vocabulary tree, and we utilize the computational power of modern graphics processing units to accelerate several highly data-parallel steps in the view registration procedure.

5.2.1 Vocabulary Tree and Document Scoring

A critical step in the overall approach is to determine relevant documents that are tested for geometric plausibility later on. We employ a vocabulary tree approach [Nistér and Stewenius, 2006] to obtain potential matches between query image features and the keypoint descriptors associated with the 3D documents in an efficient manner. The utilized tree is a complete tree with $D = 3$ levels and $K = 50$ children for every internal node. The leaves of the tree correspond to quantized feature descriptors (visual words) obtained by a hierarchical K-means clustering procedure. The tree structure allows the efficient determination of the approximately closest visual word by $K \cdot D$ descriptor comparisons. We employ a CUDA-based approach executed on the GPU for faster determination of the respective visual words. The speed-up induced by the GPU (about 15 - 20 on a GeForce GTX280 vs. Intel Pentium D 3.2Ghz) approach allows to incorporate more descriptor comparisons, i.e. a deeper tree with a smaller branching factor can be replaced by a shallower tree with a significantly higher number of branches. The implementation details are described in Section 2.5.1.

5.2.2 Feature Matching and Pose Verification

After the score of 3D documents with respect to a new query image is determined, the geometric relationship between the top-ranked documents and the query image needs to be established. First, the extracted features in the query image are exhaustively compared with the descriptors associated with the 3D points in the tested document. Our approach to feature matching consists of a call to the dense matrix multiplication in the CUBLAS library with subsequent instructions to apply the distance ratio test and to report the established correspondences. The implementation details are described in Section 2.6.

If enough putative feature matches are obtained, the actual pose for the query image needs to be determined (if such pose exists at all with respect to the currently considered document). We assume that the intrinsic parameters of the camera are known, hence we can rely on the fast Three-Point algorithm and RANSAC (e.g. [Raguram et al., 2008]) to determine the absolute pose from three point correspondences [Fischler and Bolles, 1981, Haralick et al., 1991].

5.3 Experimental Evaluation

In order to evaluate our view registration method we create a 3D model database representing different locations of a city. For model reconstruction a pre-calibrated digital consumer camera is used. Images are of resolution 3072×2304 and taken at wide angle ($65.4°$ FOV). The models are reconstructed using the algorithm described in Section 3.1. In addition, we acquired several video sequences of resolution 848×480 pixel from the same scene. The videos are taken by a freely moving hand held camera. Due to the unconstrained camera motion, individual frames do not necessarily have a visual overlap to the 3D landmark, i.e. the camera may sometimes point to directions where no 3D information is available (e.g. ground, sky etc.).

Some typical video frames showing the challenging view conditions are depicted in Figure 5.12(a)(b), 5.10(a) and 5.11(a). The acquired video sequences contain large position changes, vibrations and blur caused by fast hand-held camera movements. Additionally, the images include strong changes in appearance due to motion, object occlusion and texture and illumination changes. The videos are later used for evaluation. For 3D model reconstruction 1093 images are processed and 450.000 points triangulated from 1.600.000 SIFT descriptors. Sparse point clouds and camera orientations of the respective models are shown in Figure 5.9. Table 5.1 summarizes our 3D model dataset. After applying mean-shift clustering, the number of descriptors reduces on average to 40% of the original size. These value varies between the seven 3D models with respect to the scene complexity and the number of redundant views used in the reconstruction pro-

3D Model	$\#f$	$\#d$	$\#\tilde{d}$	# real views	# syn. views	# comp. scene
M1 (Street1)	76197	243403	126653	207	1548	432
M2 (Church)	80355	326873	117258	128	1026	196
M3 (Square1)	75258	312753	134153	190	1802	538
M4 (Street2)	69157	232980	134159	284	2008	776
M5 (Street3)	31707	119555	51899	59	476	91
M6 (Square2)	61223	215635	98133	137	843	258
M7 (Square3)	56681	185922	86534	88	3987	712
Total	450578	1637121	748789	1093	11690	3003

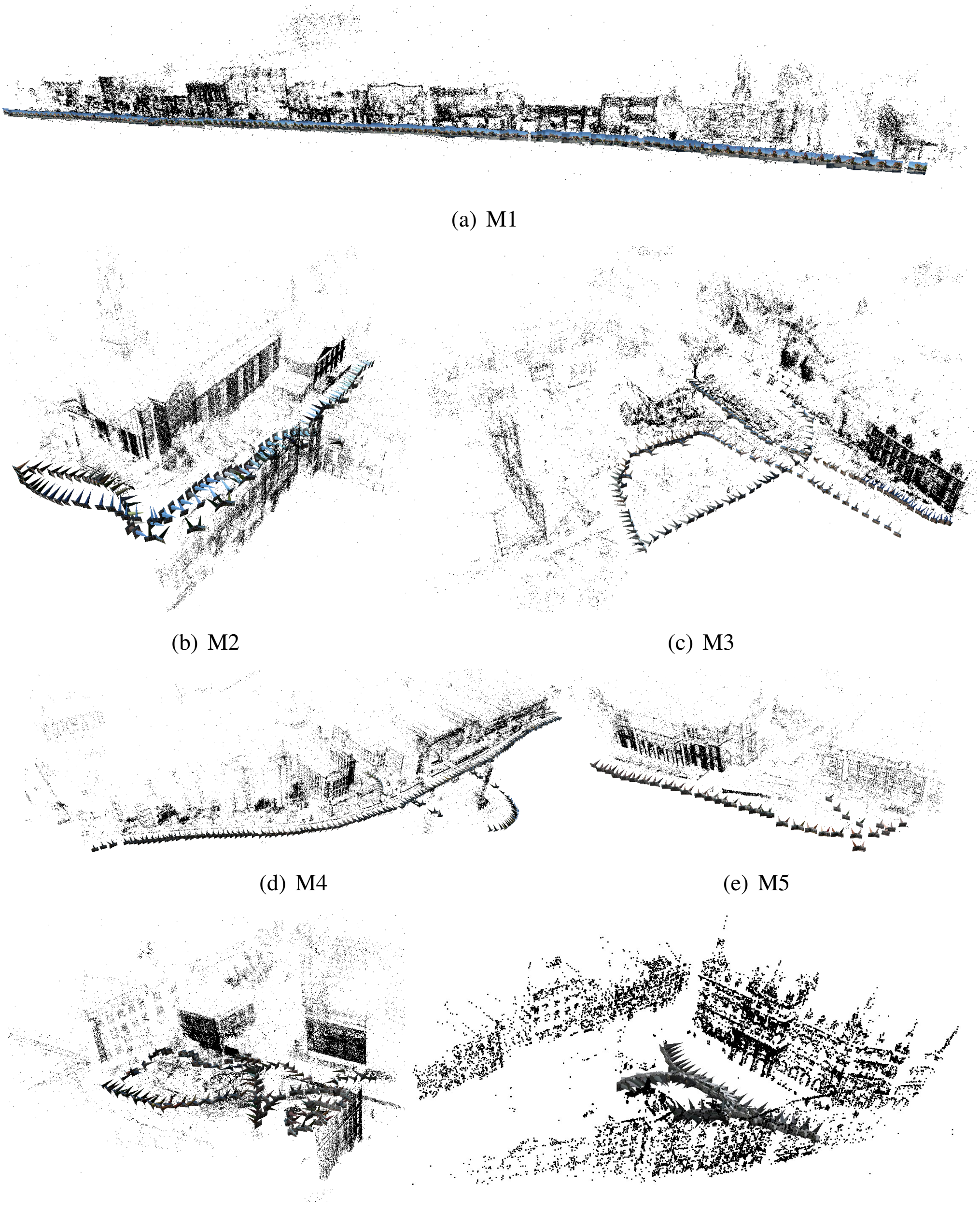
Table 5.1: List and properties of a sparse 3D model database. Each row lists: $\#f$, the number of features in the 3D model; $\#d$, the number of raw descriptors, $\#\tilde{d}$, the number of descriptors after mean-shift clustering; # *real views* the number of original structure from motion views; # *syn. views* the number of synthetic views; # *compressed views*, the number of views used for the compressed scene representation.

cess. Note, the reduction in memory consumption is significant, e.g. 1.600.000 SIFT descriptors ($\sim 800MB$) are compressed to less than 750.000 descriptors ($350MB$). For each 3D model we estimate an average ground plane and evenly place synthetic views with a distance of equivalently $2m$ in between. At each grid position we insert 12 synthetic views with field-of-view $\alpha = 65°$ and resolution 1024×1024 pixel (to model portrait and landscape mode images simultaneously). The heading between cameras is 30°, therefore a full panoramic view at the given position is covered. Since 3D structure is only expected above the ground plane, the cameras are tilted 10° towards the positive horizon. The full set of synthetic and real views contains 12700 documents, which are subsequently reduced to 25% of the original size by our compression procedure.

5.3.1 Registration Performance

We evaluate the view registration performance by measuring the percentage of video frames for which a valid pose is found after considering the k-th top ranked 3D document from the vocabulary tree scoring. A pose returned by the RANSAC procedure is only considered as reliable, if ten effective inliers are found. The effective number of inliers is determined in terms of coverage times the raw number of inliers as described in Section 2.7.2. This is a more robust measure than the standard raw inlier count, since also the spatial distribution of points is taken into account. Of course, the effective inlier number does not reflect a ground truth, but at least in our large scale experiments (registering thousands of views) we did not find false positives among the set of registered frames.

A detailed quantitative evaluation of our proposed compressed scene representation with respect to a pure image based approach is shown in Figure 5.10 and 5.11. Experimental results confirm that our compressed scene representation based on synthetic views delivers superior reg-



(a) M1

(b) M2

(c) M3

(d) M4

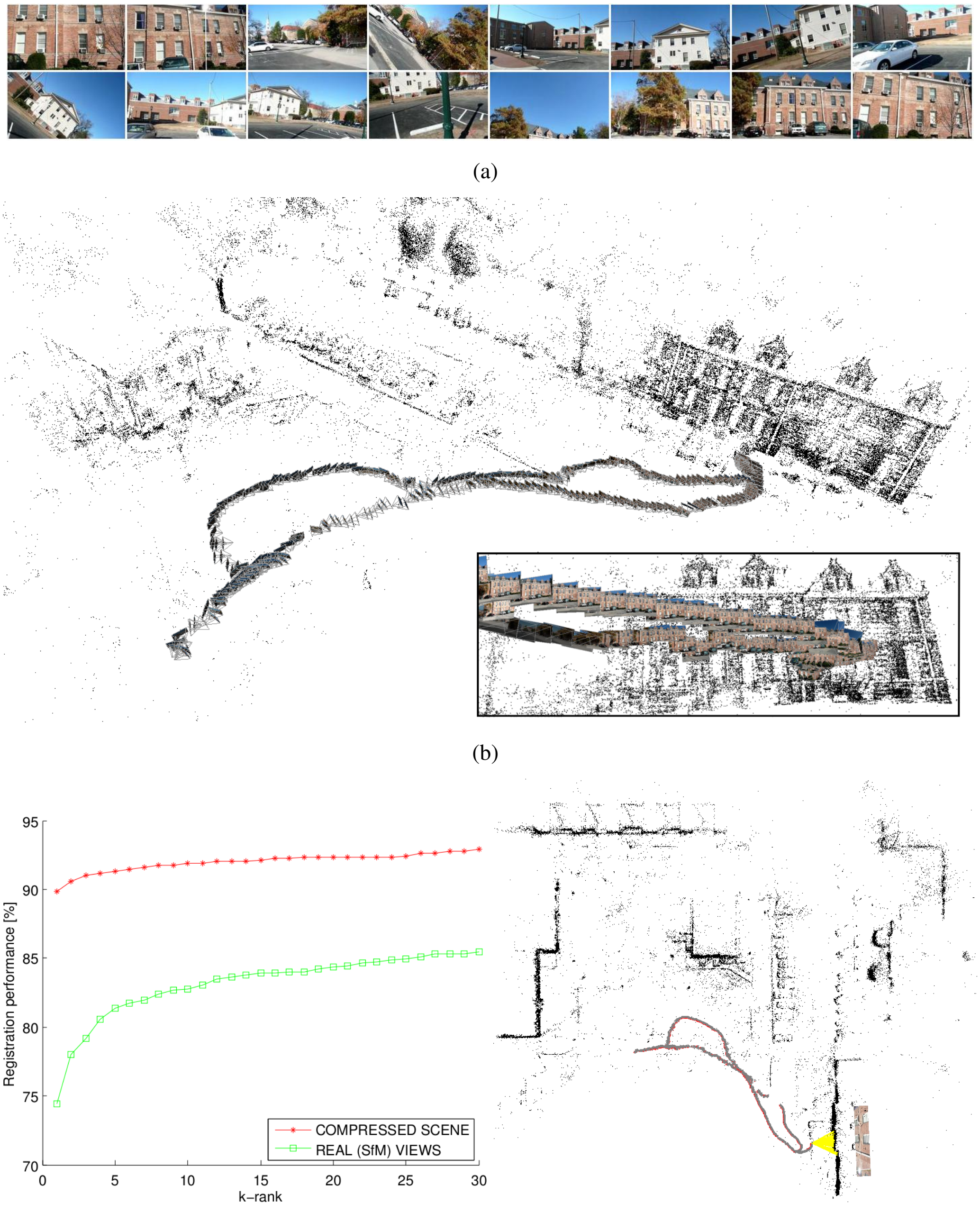
(e) M5

(f) M6

(g) M7

Figure 5.9: Structure from Motion reconstructions of seven scenes used through our hand-held camera tracking experiments.

istration performance than approaches that are only based on original images. This is especially true for the localization result shown in Figure 5.11. While the compressed scene representation achieves a recognition rate of 100% by only testing the first ranked synthetic view, the approach based on original views only achieves 90% after testing up to 30 original images.



(a)

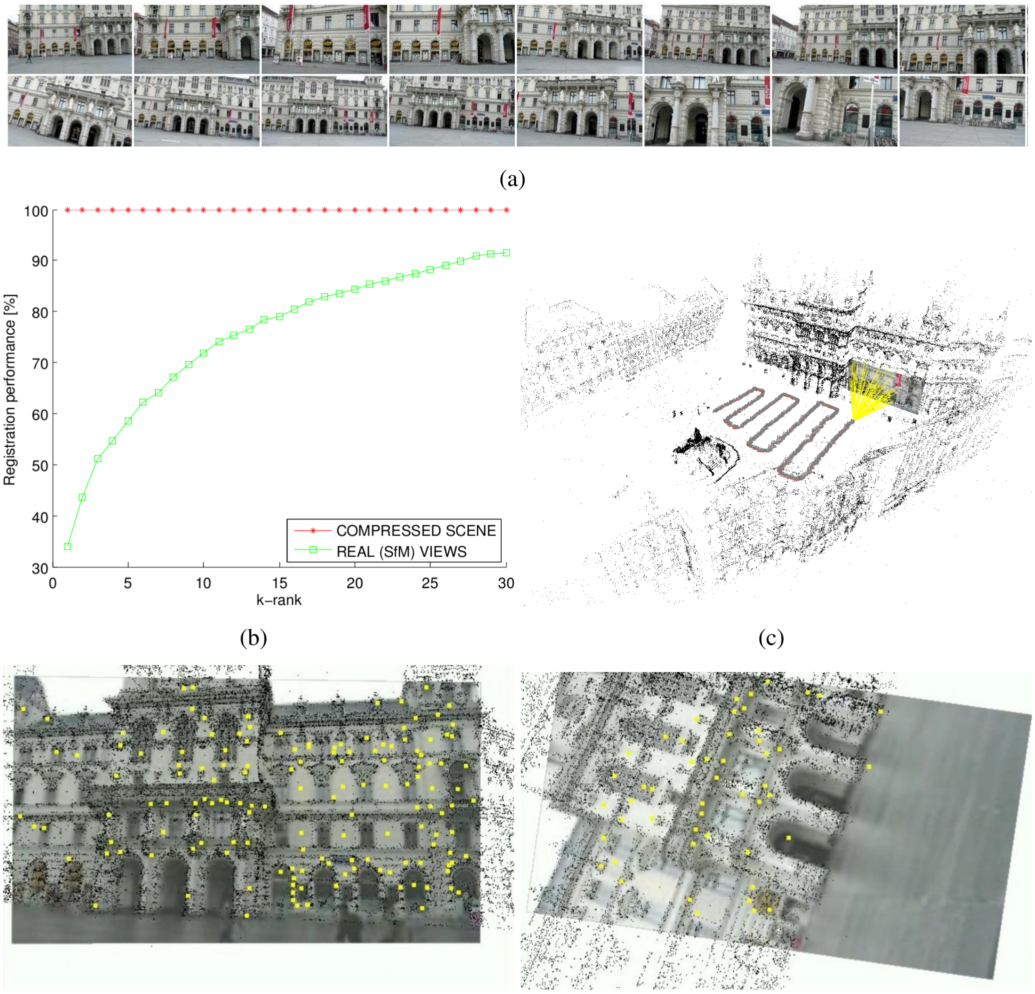
(b)

(c)

(d)

Figure 5.10: (a) Some sample frames from a hand-held video used for evaluation. (b) Registration result of a 2000 frame video with respect to model M3. (c) Location recognition performance with respect to compressed scene representation (red) and real/original images (green) used for structure from motion computation. (d) Sparse point cloud and registered camera locations.

Figures 5.12(c) and 5.12(d) show registration performance for two hand-held video sequences (V1,V2) with respect to different 3D document strategies. Our evaluation includes also a com-



(a)

(b)

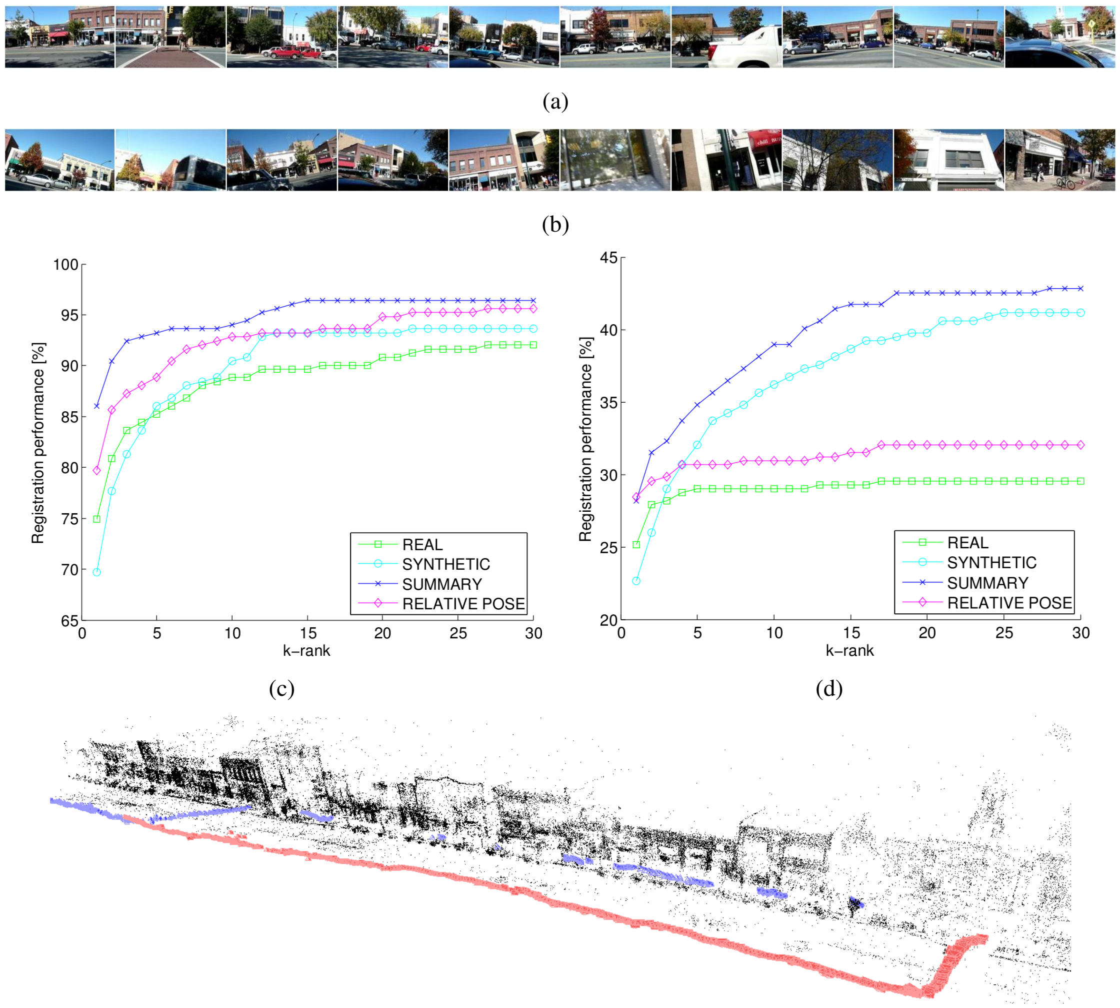
(c)

(d)

(e)

Figure 5.11: (a) Video frames used for evaluation. (b) Location recognition performanc. A recognition rate of 100% is achieved by using our synthetic view approach with the compressed 3D document set. (c) Sparse point cloud and registered camera locations. (d),(e) Registration results showing inliers (yellow) and 3D scene points.

parison to a pure image based method, with the Five-Point algorithm [Nistér, 2004] used for pose verification (relevant parameters are adjusted to get comparable timings to the Three-Point method). Note, V1 was taken close to the camera positions which were used for model reconstruction, but at different resolution. Overall for sequence V1 a higher recognition rate is achieved than for the more challenging sequence V2, that follows a different path approaching the facades. For both cases, the reduced document set based on synthetic and real views gives the best registration performance.



(a)

(b)

(c)

(d)

(e)

Figure 5.12: (a),(b) Some sample frames of two video streams V1, V2 acquired with a hand-held camera. V1 was taken close to original camera position of real views (images from model reconstruction), whereas V2 follows a different path. (c) and (d) show registration performance measured in terms of percentage of registered views after considering the k-top ranked images from the vocabulary tree scoring for V1 and V2, respectively. Each graph shows: *REAL*, set of 3D documents formed by views from model reconstruction; *SYNTHETIC*, synthetic views; *SUMMARY*, reduced set of 3D documents computed by scene compression; *RELATIVE POSE*, image based retrieval with Five-Point relative pose verification. (e) Side view showing registered views from video stream V1(red) and V2(blue), respectively.

5.3.2 Timings

Critical thresholds in terms of timings are the maximal number of RANSAC iterations N_{max} and the number of extracted features $|\mathcal{Q}|$ in the input image and 3D document $|\mathcal{D}|$. We set $N_{max} = 500$

(corresponding to a maximal outlier fraction of $\epsilon \approx 0.8$ at a 95% confidence level), $|\mathcal{Q}| = 1600$ and $|\mathcal{D}| = 2500$, which results in execution times of $25ms$ to test a single 3D document on average. By using the publicly available SiftGPU[1] software and only testing the first-ranked 3D document from the vocabulary tree scoring, view registration can be done in real time. Average timings are listed in Table 5.2.

Operation	time $[ms]$
SiftGPU 848 × 480	33
Vocabulary Tree Traversal K=50 D=3	4
Inverted File Scoring	15
Matching 1600 × 2500 SIFT key's	$10 \times k$
RANSAC Three-Point (up to 500 samples)	$15 \times k$

Table 5.2: Average timings of our system on an Intel Pentium D 3.2Ghz and a GeForce GTX 280. k is the number of top-ranked documents geometric verification is applied on.

5.4 Limitations of Natural Features for Localization

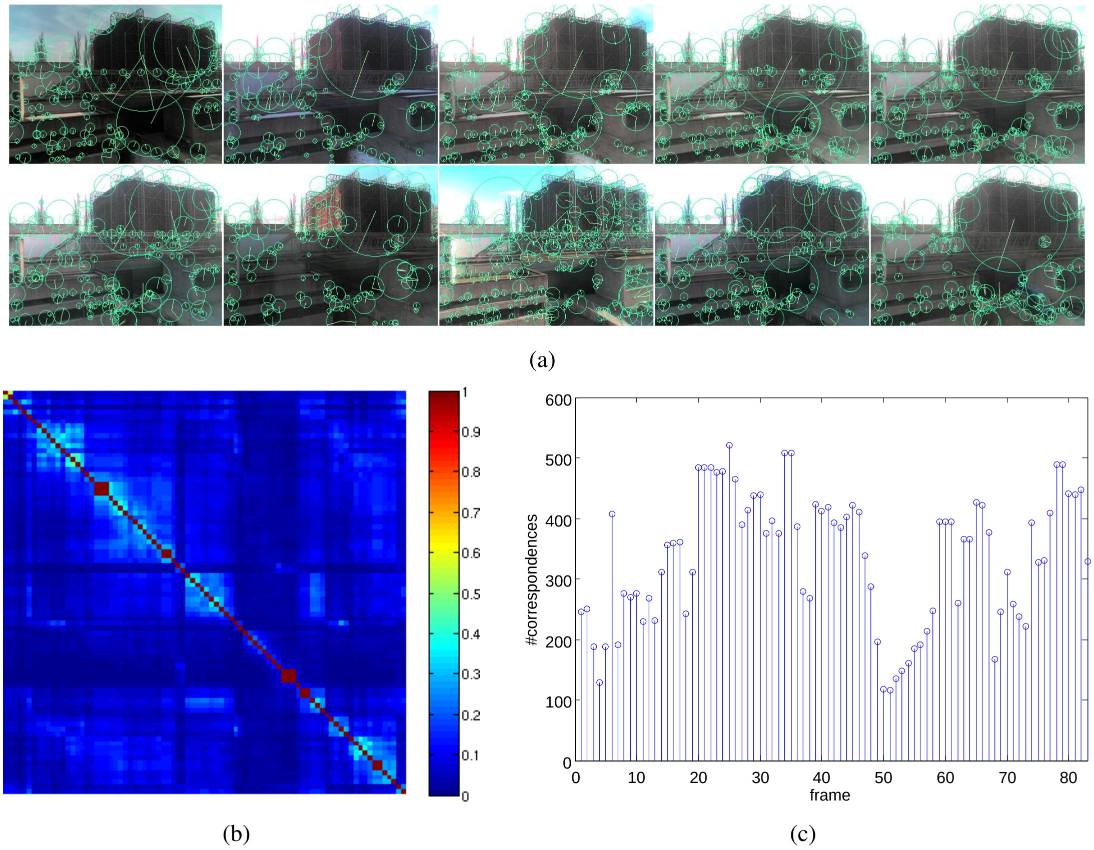
The evaluation framework of [Mikolajczyk and Schmid, 2005] includes rotation, scale change, viewpoint variation, image blur, JPEG compression and light changes to test the repeatability of image features. While current state-of-the-art feature detector and descriptors can widely handle correspondence computation of these images, matching images from real world scenes is often more challenging. This is mainly based on two reasons. First, large viewpoint changes often introduce large distortions on the geometry. Second, illumination variations and shadows occur that introduce large variations on the scene texture. The combination of those two effects makes unconstrained view registration based on local descriptors a challenging problem. For instances, correspondence computation between images taken at sunshine conditions (comprising many shadows and light spots) and images acquired during cloudy days (i.e. ambient light) is a challenging task. Changing weather/season conditions strongly effect the appearance of a scene and extracted number/locations of features and descriptors. We perform matching experiments on two datasets. The first experiments considers scene variations that are due to illumination changes within one day. The second experiments considers seasonal variations.

5.4.1 Day Variations

In our first experiment we captured 83 frames from one and the same viewpoint over a time period of approximately ten hours (see Figure 5.13(a)). SIFT keys are extracted from each image and exhaustively matched between all possible image pairs. We use the Lowe distance ratio test [Lowe, 2004] with $r = 0.8$ to decide weather two features match. Quantitative matching results are shown

[1] `http://cs.unc.edu/~ccwu/siftgpu`

in Figure 5.13. On average 1500 features are extracted from each image, however only a small fraction of about 10% can be successfully matched over time. This number will further be reduced by about 50% for moderate view point changes.



(a)

(b)

(c)

Figure 5.13: (a) Sample images captured during a time period of ten hours and extracted SIFT keys (visualized as circles, only a subset of features at a coarse scale are shown). (b) Matching matrix showing the fraction of successfully matched features between all image pairs. (c) Average number of successfully matched SIFT keys between view i and the set of views $I \backslash i$.

5.4.2 Seasonal Variations

In our second experiment we study how the SIFT key detection and matching performs with respect to seasonal changes. Our experiments are based on an image sequence acquired by a webcam over a long time period of a whole year. Images are taken every first day of each month from January to December. During one day eleven images were captured on a hourly basis between 7 a.m. to 5 p.m.. Hence, the image database is very diverse in terms of illumination (weather conditions) and seasonal appearance (snow, vegetation changes, occlusion). Again exhaustive nearest neighbor matching is used to estimate matching performance of SIFT features with respect to all possible image pairs. The outcome of our experiments are summarized in Figure 5.14. Whereas, matching of image pairs taken within one day is quite stable (consider Figure 5.13(b) e.g. February, Mai, September, December), only a small fraction of features can be matched between images acquired at different seasons (e.g. summer / winter). From our experiments we conclude that a typical outdoor scene might undergo large variations during a day/season that cannot be robustly handled by a single SIFT representation. For instance, shadows and vegetation changes introduce new structures and occlusions which result into different keypoints and distorted descriptors. Even though, SIFT is basically designed to cope with some variations in illumination and scene appearance, a monolithic representation does not always suffice for location recognition in real world scenes.

5.5 Conclusion and Discussion

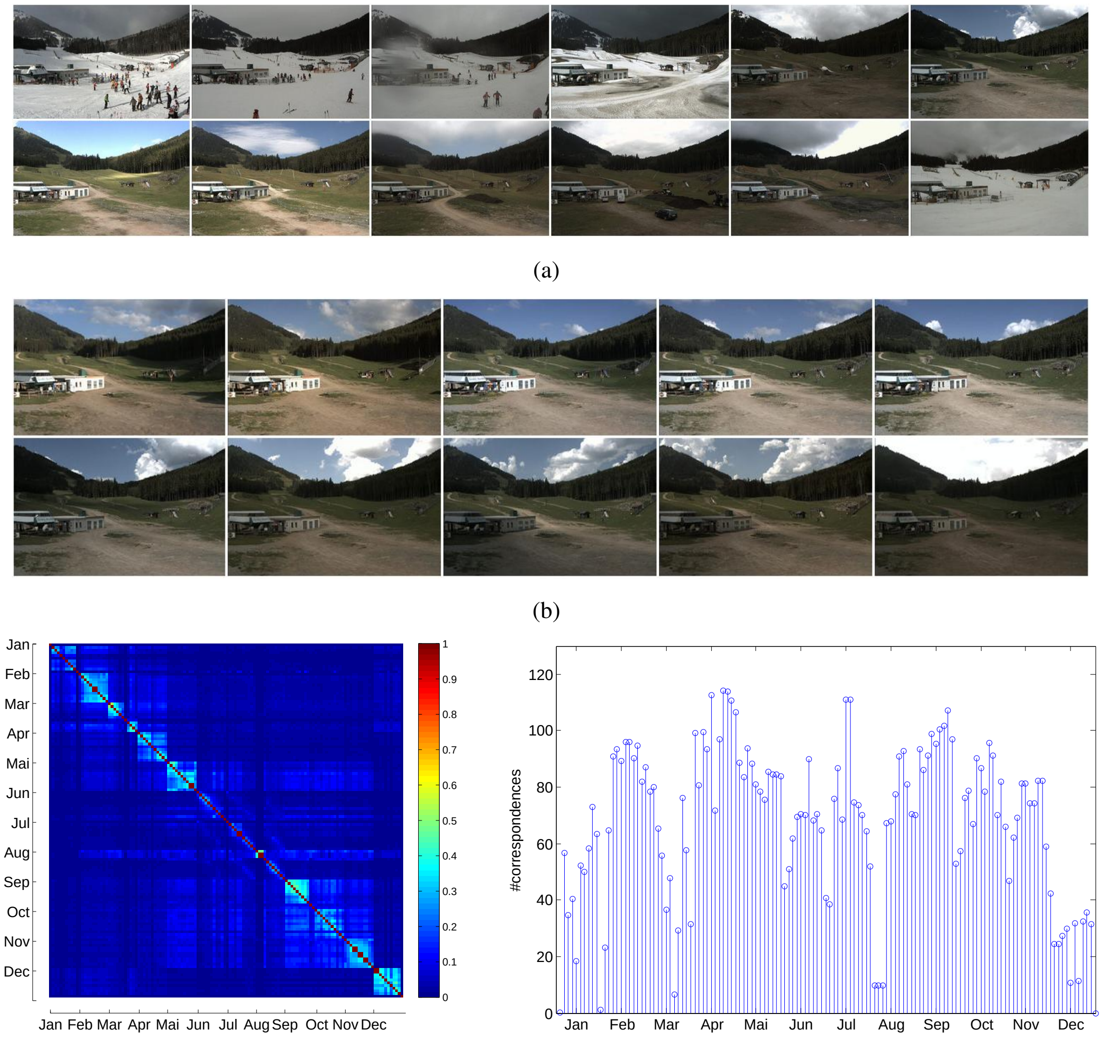
We introduced a novel method for image based real-time view registration and localization in large out and indoor environments. The main contributions of the proposed method are,

(i) the introduction of synthetic views to allow better registration of images taken from novel viewpoints,

(ii) an effective document compression procedure for provided real imagery and the synthetic ones in order to reduce the database size, and

(iii) a novel scoring function to rank the documents returned by vocabulary tree queries.

Video-based inside-out tracking for large outdoor environments can be achieved with real-time performance. We demonstrated localization results on large structure from motion point clouds comprising more than 1.5 Million 3D points. View registration of video data can be done at 15 fps with a recognition rate of more than 90%. The algorithm was tested on a variety of data and showed superior results compared to existing methods. Furthermore, we studied basic limitations of our approach with respect to changing illumination and appearance conditions of a scene. We conclude that a monolithic 3D representation based on SIFT features in general may not be sufficient for image based localization, since a typical scene undergoes variations that are beyond the variations SIFT is robust against. A simple solution to address this problem would be to enrich the 3D representation with more images, which could be naturally incorporated into our system. However, being a feature based approach there will still exist scenes where our approach fails due to missing texture information.



(a)

(b)

(c)

(d)

Figure 5.14: (a) Images taken by a web-cam over a time period of one year. Each image is captured at 12 p.m. on every first day of the month from January (top left) to December (bottom right). (b) shows ten images taken hourly on June 1th between 7 a.m. an 4 p.m., respectively. (c) Matching matrix depicting the fraction of successfully matched features between each image pair. (d) Average number of successfully matched SIFT keys between view i and the set of views $I \backslash i$.

Chapter 6

Localization Applications

In this chapter three different applications and variants of our proposed localization framework are presented. The first application is the registration of community photo collections with respect to sparse city models reconstructed by structure from motion techniques as described in Section 3.1. In Section 6.2 we apply our view registration framework for localizing a Micro Aerial Vehicle (MAV) in GPS-denied outdoor environments. Our approach significantly outperforms current state of the art Simultaneous Localization and Mapping (SLAM) approaches. Finally, in Section 6.3 we demonstrate a localization framework that runs on mobile phones. The restricted processing power and limited memory of the target platform requires several algorithmic modifications.

6.1 Registration of Community Photo Collections

Today, there exist an ever increasing amount of photographs from places of interest on earth. For instance, solely on *flicker.com* there are more than one million of photographs related to "Vienna". Popular sites of a city like historic buildings, facades, fountains, sculpture and paintings are captured from hundreds or thousands of viewpoints and under varying illumination conditions. Current advances in structure from motion for unordered image sets [Snavely et al., 2006, Agarwal et al., 2009, Frahm et al., 2010] have demonstrated that large scale 3D reconstruction based on huge Internet photo collections is feasible. While reconstruction systems solve for camera orientations and 3D geometry simultaneously, computing structure from motion on the whole dataset is not always necessary. It has been shown by Snavely et al. [Snavely et al., 2008b] that a skeletal subset of images from redundant community photo collections is often sufficient for 3D model reconstruction. Once a skeletal set of images is reconstructed, new images can be inserted by view registration (localization) as described in this section.

6.1.1 Registration Algorithm

In order to apply the localization approach described in Chapter 5 to images from the web, some algorithmic modifications are necessary. First of all, the resolution and field of view of synthetic cameras has to be chosen in order to optimize registration performance. Second, images from community photo collections leak intrinsic calibration, hence the efficient Three-Point method cannot be directly applied. We address the first issue by analysing image statistics of Internet photo collections. In particular we download a set of images related to Vienna from the panoramio homepage (see Figure 6.2(a)). The dataset comprises 17282 images, where a subset of 4833 images have EXIF information associated. For images with EXIF we can extract an approximate value for the focal length, which we use as a prior to determine a mean field of view. Figure 6.1 shows statistics about the distribution of the field of view (FOV) and the image resolution. We estimate a mean focal length and field of view of synthetic cameras $\tilde{\theta} = 50°$ and an average image resolution of one megapixel. The estimated values are used to determine a suitable focal length and image resolution for respective synthetic views. In practice we use a focal length of $\theta = 60°$ (the slightly increased field of view accounts for boundary effects), and set the image resolution to 1024×1024 pixel to simultaneously account for portrait and landscape image formats. Synthetic images are depicted in Figure 6.3. In order to determine the pose of the current camera, we cannot directly apply the Three-Point method because accurate intrinsic calibration of Internet photos is normally not available. Without knowledge of camera parameters, the direct linear transform (DLT) algorithm [Hartley and Zisserman, 2000] can be used to solve for the camera projection matrix from a minimal number of six 2D to 3D point correspondences. The projection matrix $P = [\mathbf{p}_1,\ \mathbf{p}_2,\ \mathbf{p}_3,\ \mathbf{p}_4]$ can then be decomposed into external and internal orientation from,

$$P = [M| - M\mathbf{C}] = K[R| - R\mathbf{C}] \tag{6.1}$$

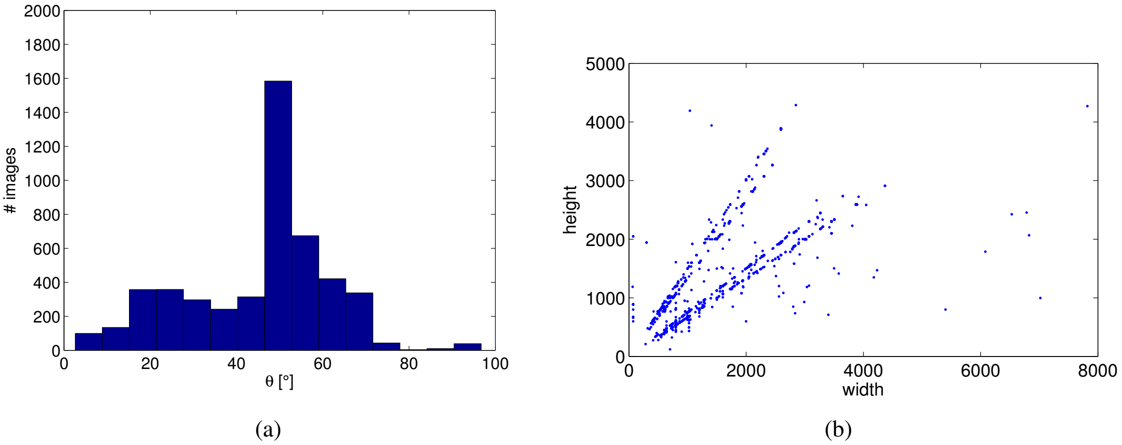
using the RQ-Decomposition $M = KR$, where $\mathbf{C}$ is the camera center $\mathbf{C} = -K^{-1}R^{\top}\mathbf{p}_4$. While the 6-point DLT approach works well for scenes without dominant planar structure and low outlier fractions, it fails in situations where mismatches dominate. We therefore take a different approach as proposed in [Li et al., 2010] and use the Three-Point pose method on weakly calibrated images. We assume that the principal point is at the image center and the focal length is taken from the EXIF [1] tags of an image. If EXIF information is not available, we discretize a reasonable range for the focal length and apply the Three-Point algorithm for a set of potential focal length values. The pose/focal length pair with the highest number of inliers is reported.

6.1.2 Experimental Evaluation

We apply our view registration technique to images downloaded from the web. A calibrated camera was used to reconstruct three landmarks of Vienna (Graben street, Michaeler square and Josef square) from 117, 128 and 622 images, respectively. Then synthetic views are placed in the scene as depicted in Figure 6.3. For these particular landmarks we gathered a set of images from the Panoramio webpage, geographically associated with these places of interest (see Figure 6.2(a)).

We select a relevant subset of 266 images that have a potential visual overlap with the reconstructed scenes. Some challenging sample images are shown in Figure 6.2. The photos are taken at different illuminations conditions, under large view point changes and partial occlusion. In order

[1] http://www.exif.org/



(a)

(b)

Figure 6.1: (a) Distribution of the angle of view θ (field of view) and (b) distribution of the image resolution for community photos from Vienna downloaded from the Panoramio website.

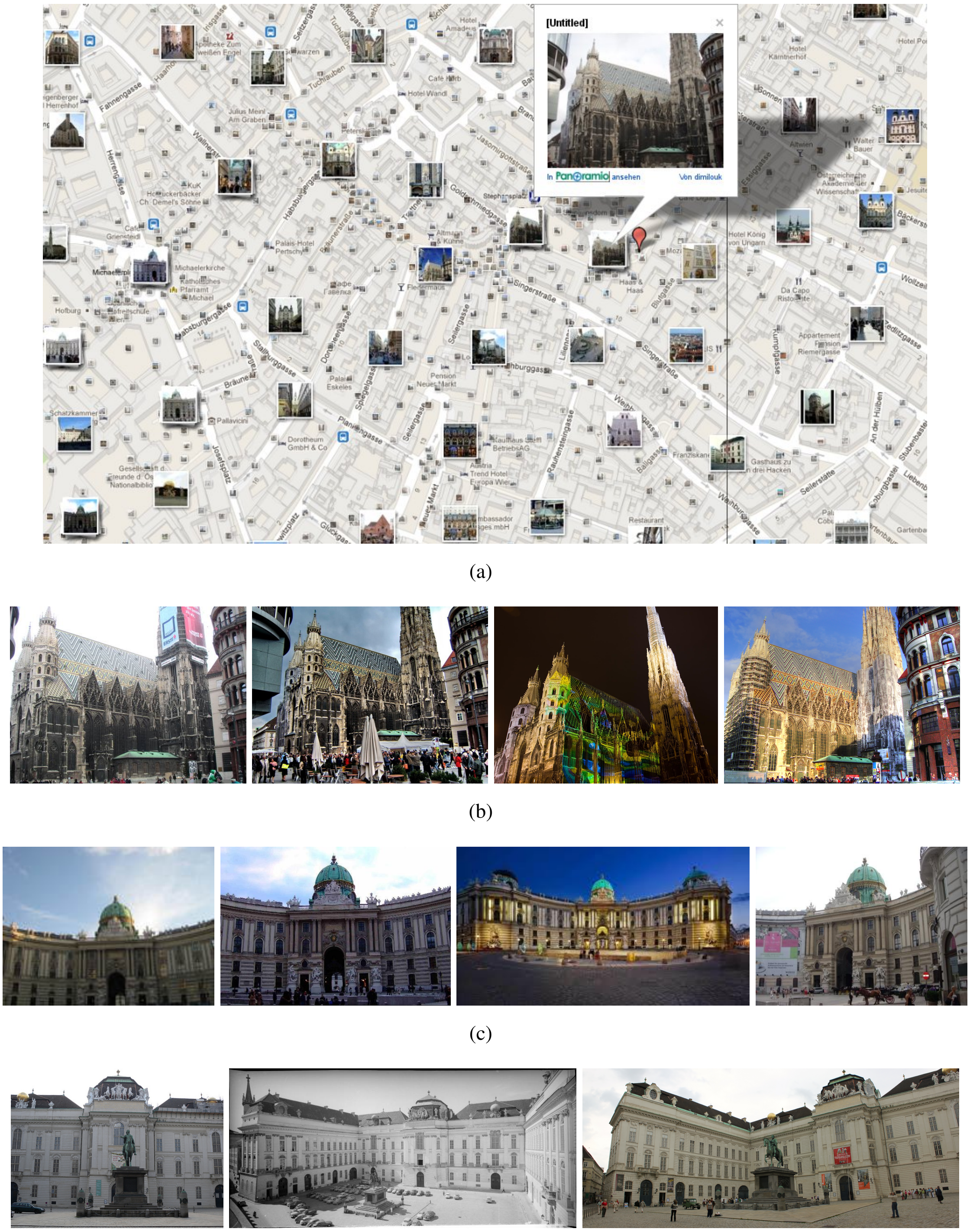
to determine the camera pose, we use the calibrated Three-Point method and exhaustively test ten focal lengths with respect to a field-of-view range $[30°..90°]$. By using our approach we are able to efficiently register 165 out of 266 images by considering up to ten top ranked 3D fragments, only. Qualitative registration results are shown in Figure 6.4. Our algorithm achieves a registration rate of 62% and requires less than 0.27 seconds to register or reject an image. Table 6.1 shows registration performance and runtimes on the Vienna dataset for different localization approaches. Current state-of-the-art view registration methods represented by [Li et al., 2010] and [Sattler et al., 2011] achieve slightly higher registration rates but our proposed algorithm is scalable and more efficient in terms of timing.

Approach	# images registered	reg. percentage	time [s]	
			registered	rejected
Ours (GPU)	164	64%	≤ 0.27	≤ 0.27
P2F [Li et al., 2010]	204	76%	0.55	1.96
2D-3D [Sattler et al., 2011]	211	79%	1.83	9.95

Table 6.1: Registration performance for the Vienna dataset compared to current state-of-the-art approaches represented by [Li et al., 2010] and [Sattler et al., 2011].

6.1.3 Conclusion and Discussion

We have presented an efficient view registration method for Internet photo collections based on large scale 3D models of urban scenes. Our approach achieves competitive registration rates than current state-of-the art but is much faster. Furthermore the proposed approach is fully scalable and prior pose information can be easily integrated.



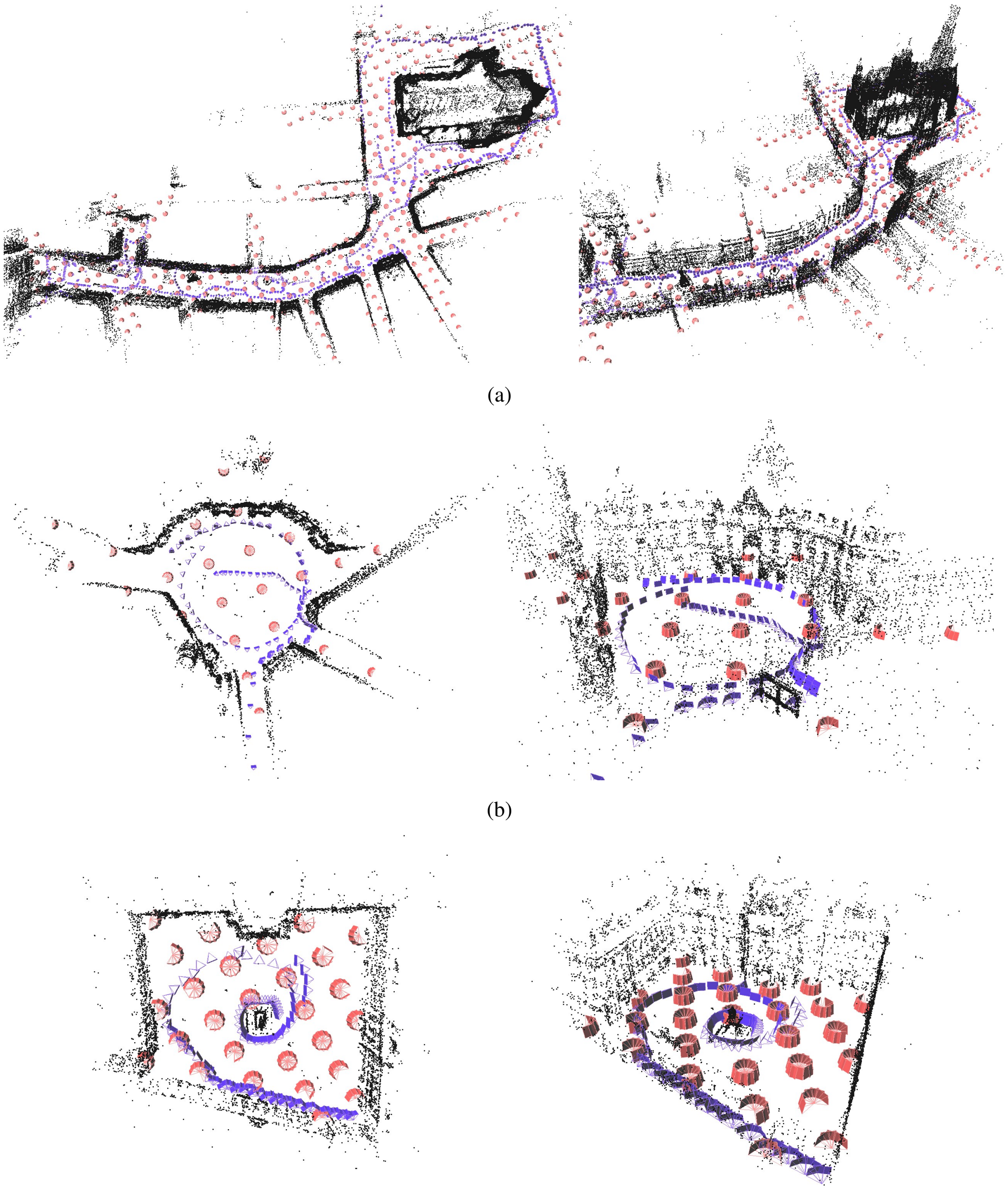
(a)

(b)

(c)

(d)

Figure 6.2: (a) Geo-tagged community photo-collections (from *www.panoramio.com*) overlaid on a map of Vienna (*maps.google.com*). (b)-(d) Challenging images from community photo collections containing different illuminations, large view point changes, occlusion and non discriminative objects.

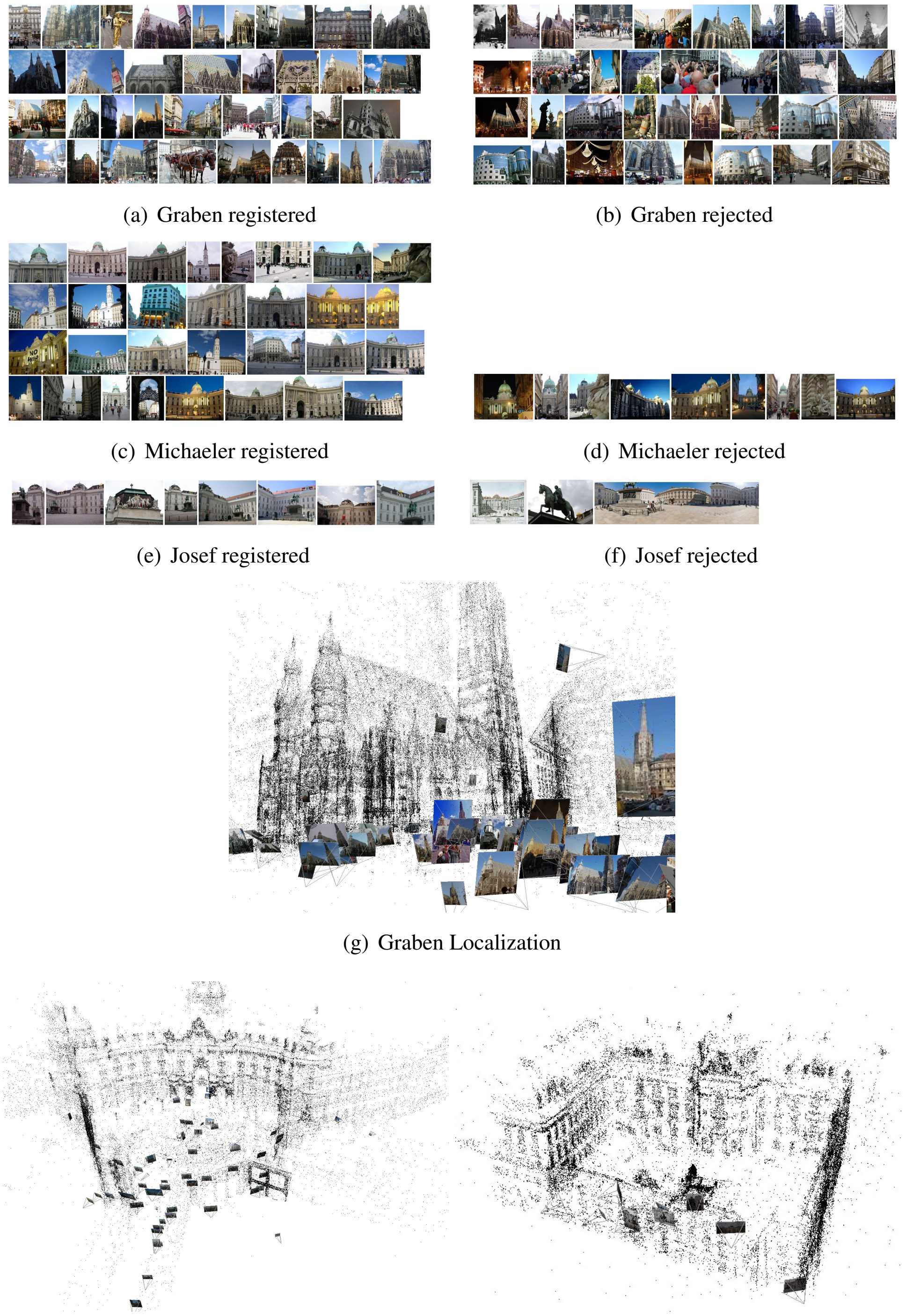


(a)

(b)

(c)

Figure 6.3: Reconstruction of three different landmarks from Vienna, (a) Graben street, (b) Michaeler square, (c) Josef square. Blue cameras are original views used for structure from motion, red cameras represent synthetic views.



(a) Graben registered

(b) Graben rejected

(c) Michaeler registered

(d) Michaeler rejected

(e) Josef registered

(f) Josef rejected

(g) Graben Localization

(h) Michaeler Localization

(i) Josef Localization

Figure 6.4: (a)(c)(e) Examples of successfully registered views and (b)(d)(f) and respective images that could not be registered (after testing up to 10 top ranked 3D point fragments) in the database. (g)(h)(i) Registered images with respect to the landmark reconstructions.

6.2 Natrual-Landmark based MAV Localization

Highly accurate localization of a Micro Aerial Vehicle (MAV) with respect to a scene is important for several applications including surveillance and inspection. For indoor navigation and trajectory planning [Mellinger et al., 2010] current state of the art MAV pose estimation is based on the commercial Vicon Motion Systems[1]. Such a system allows highly accurate outside in tracking with an achievable precision of sub-centimeters but is restricted to small workspaces of maybe 10×10m and can only be used indoors. In outdoor environments conventional methods based on Global Positioning Systems (GPS) and Inertial Measurements Units (IMU) achieve accuracies in the range of meters, but the precision is often not sufficient for target applications like navigation in urban environments. Image based computer vision methods offer a natural way to address the localization problem. A camera beeing a passive device is cost efficient, non intrusive and the accuracy of the pose estimate automatically adapts to the depth of the scene.

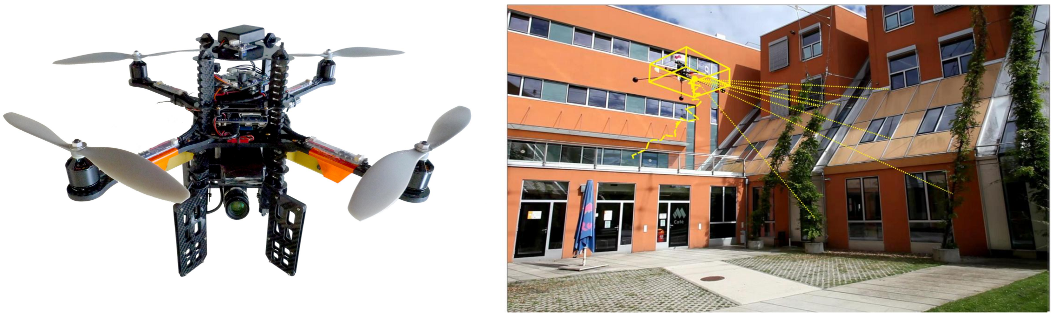
Vision based localization systems in robotics are often based on markers [Rudol et al., 2010] or rely on simultaneous Localization and Mapping (SLAM) techniques [Nistér et al., 2004, Davison et al., 2007]. Recently, [Bloesch et al., 2010] demonstrated vision based MAV navigation in unknown and unstructured environments using the Parallel Tracking and Mapping (PTAM) software [Klein and Murray, 2007]. Here map building and localization is done simultaneously. We follow a different approach and decouple mapping from localization. First, a highly accurate visual map of the environment is recorded using structure from motion techniques. Second, the pose of the MAV is determined with respect to the landmark reconstruction. Such a method has several key advantages over SLAM. First, given a geo-registered visual 3D model of the environment [Wendel et al., 2011], localization can be done in a global metric scale. This would not be possible for a monocular SLAM approach without considering additional input modalities. Moreover, once a detailed visual model of an environment is available, the localization approach is fully scalable and allows direct data fusion with other sensors such as GPS or IMU.

6.2.1 Hardware Setup

The MAV used in our experiments is a "Pelican" quad-rotor from Ascending Technologies[2], depicted in Figure 6.5. The MAV is equipped with a single, rigidly mounted consumer camera (Panasonic TZ3) that acquires video sequences at a resolution of 848×480 pixels at wide angle (65° field of view). The images required to reconstruct the scene were captured from eye-level above ground using a Canon EOS 5D. The SLR camera has a resolution of 5616×3744 pixels with a fixed $20mm$ wide angle lens. This camera is also used to acquire a ground video with a resolution of 1920×1028 pixels from an observers point of view. These image sequences depict the MAV as well as the surrounding scene, therefore we can use them to evaluate our localization performance. For both camera setups, we use the accurate and flexible calibration method described in Section 3.1.2 to simultaneously estimate the focal length, principal point, and radial

[1] http://www.vicon.com

[2] http://www.asctec.de



(a) (b)

Figure 6.5: (a) Pelican quad-rotor MAV from Ascending Technologies equipped with a single, wide angle camera used for localization. (b) Localization result of our proposed algorithm with respect to the reconstructed scene.

distortion parameters.

6.2.2 Visual Landmark

We employ structure from motion to reconstruct the Atrium scene depicted in Figure 6.6 from high resolution still images taken at ground level. The scene shows a modern building with hardly textured walls and many repetitive structures such as windows and vegetation. For 3D reconstruction of the atrium, a set of 157 images has been processed and 28.890 points were triangulated from 117.340 SIFT descriptors (an average of 747 points per image). The MAV equipped with a consumer grade video camera has been used to acquire an in–flight video stream for localization. It contains 4946 frames, recorded at a resolution of 848×480 pixels and 15 fps (frames per second), resulting in a total length of $5 : 30$ minutes. Images were taken at different altitudes, close to buildings and further away. Due to the real–world image capturing, images may be slightly blurred, have different illumination conditions, or show reflections in windows. Additionally, we have acquired a video stream showing the flight from an observer's point of view. This stream has been recorded with a resolution of 1920×1028 pixels, 25 fps, and is used as ground truth for evaluating our localization performance (see Section 6.2.5).

Overall the algorithmic processing steps are similar to the algorithm described in Chapter 5 with some modifications,

- the concept of virtual views is extended to 3D space, accounting for the 6DOF movement of a MAV,
- we impose prior knowledge about possible neighboring views, taking into account the causal movement of the MAV, and
- we perform fully 3D geometric validation of the localization results using an observer's view.

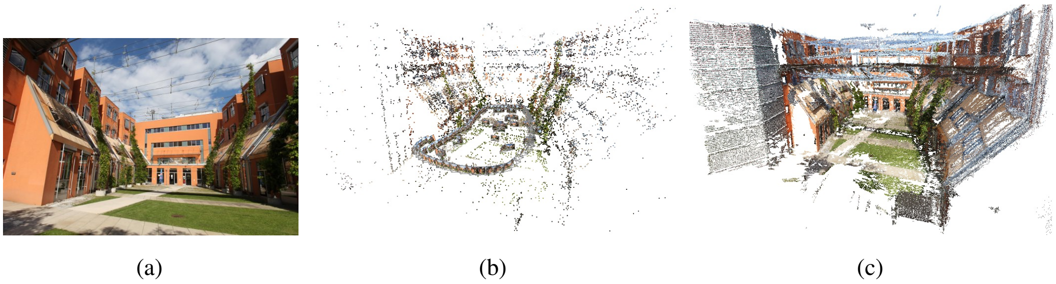


Figure 6.6: Atrium scene reconstruction from 157 views. (a) Original view of the scene. (b) Sparse model and source cameras obtained by structure from motion reconstruction. (c) Semi-Dense point model obtained by refining the sparse model with the PMVS software [Furukawa and Ponce, 2009].

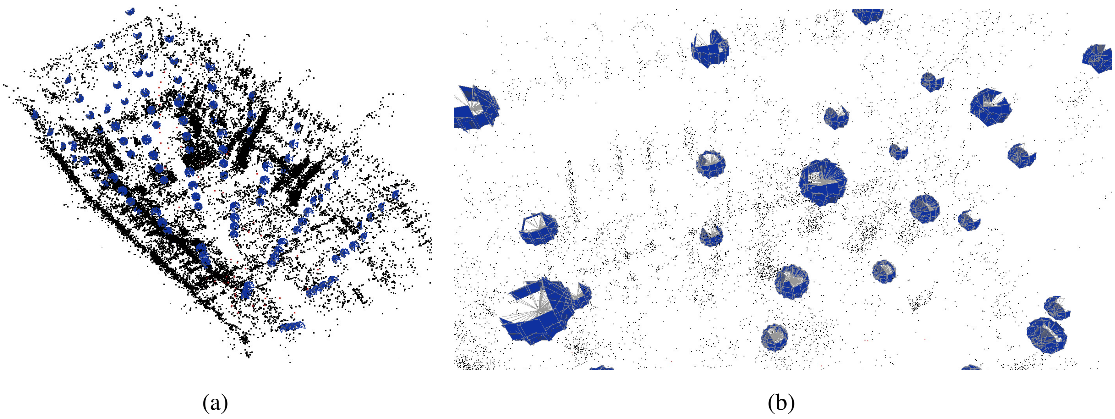


Figure 6.7: Placement of virtual cameras. Virtual cameras of a grid $\mathcal{G} = \{7, 4, 4\}$ and $\gamma = 30°$ marked in blue. (b) A detailed view of the same grid.

Given the point cloud $\mathcal{PC}$ of the atrium, we sample camera centers on a regular grid $\mathcal{G}$, and camera vectors in uniform angular steps of γ on a unit sphere. The placement of virtual cameras is visualized in Figure 6.7.

6.2.3 Localization Algorithm

For localization, we select $k = 10$ top–scoring views using a vocabulary tree in a neighborhood of $T_{max} = 1.0m$ and $R_{max} = 45°$ with respect to the previously established pose (see Figure 6.8). Finally, the top three poses with at least $|\mathcal{I}|_{min} = 10$ inliers are stored and used in the experimental evaluation. On an Intel Core2 Quad CPU with 2.83GHz and a GeForce GTX 280 we achieve an average localization speed of 4 fps. Figure 6.10 gives quantitative numbers of the registration results. With our proposed approach we are able to register 2868 out of 4946 frames (57%) of the in–flight video. The resulting flight path from take–off to landing is shown in Figure 6.9(a). Missing localizations mainly occurs due to the fact that the visual landmark is not complete, close–up and high–altitude views tend to fail more often. Another common source of error are frames distorted by motion blur and joggle, because only a subset of features can be detected. Nevertheless, our proposed approach is able to handle such local failures since the algorithm performs tracking by detection and hence automatically recovers with the next successful pose estimate. While in such a challenging scene 100% registration performance is hardly achievable, a benefit of our approach is that it never lost track for more than 60 frames, corresponding to a timespan of 4 seconds.

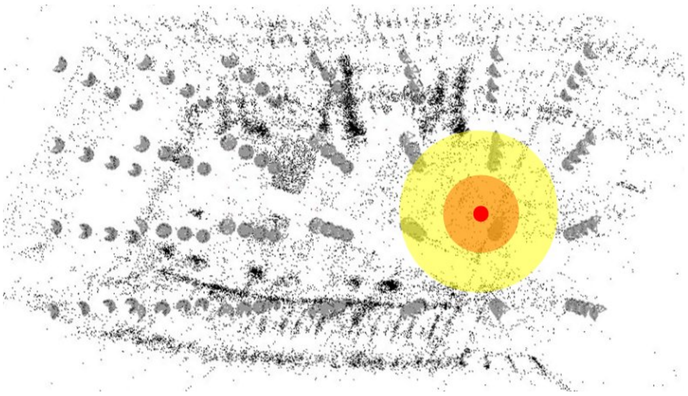
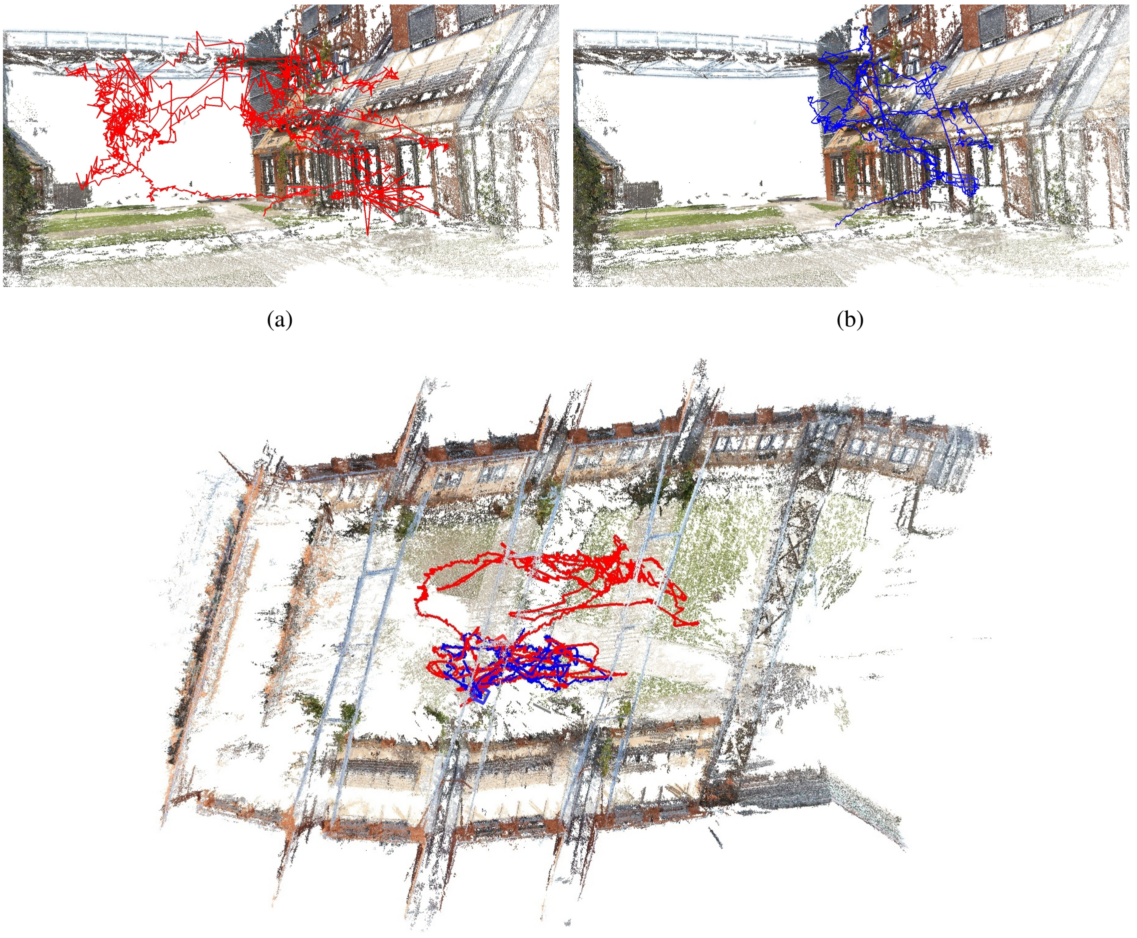


Figure 6.8: Due to the restricted and smooth motion of the MAV, after initialization, feature search can be restricted to neighboring 3D point fragments of the actual pose.

6.2.4 Comparison to PTAM

We compare our localization approach to a state-of-the-art Simultaneous Localization and Mapping software PTAM [Klein and Murray, 2007], which has recently been proposed for MAV navigation by Bloesch et al. [Bloesch et al., 2010]. Experiments show that the camera positions obtained by running the original PTAM code, which is publicly available[1], vary considerably according to the stereo initialization. Therefore, we use the recorded stream of images (with radial distortion corrected) and selected the initialization which shows best results for comparison. As the global coordinate system and the scale are not defined and differ in every execution, we align the camera centers of PTAM to those retrieved by our proposed algorithm in a least–squares sense. Figure 6.9(b) depicts the reconstructed flight path from take–off to landing. While PTAM can compete with our approach locally, i.e. in the environment it was initialized, it only works for 1828 out of 4946 frames (36%). In a direct comparison to our approach (Figure 6.9(a)), it is clearly visible that PTAM misses large parts of the flight. Another problem is the repetitiveness of the scene, which causes misleading model updates and inhibits successful relocalization within a larger space. Scene reconstruction and localization results of the PTAM framework in our scene can be seen in Figure 6.11. Even PTAMM [Castle et al., 2008], an extension of PTAM which is able to handle multiple maps, was not able to overcome this problem. As soon as the track is lost,

[1] http://www.robots.ox.ac.uk/~gk/PTAM/



(a)

(b)

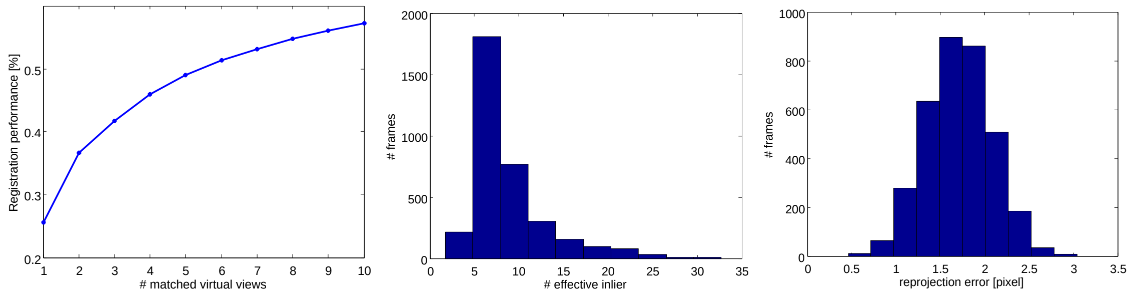
(c)

Figure 6.9: Visual localization results. (a) Flight path of our natural landmark–based approach with an error of less than $0.5m$ in 92% of all frames. (b) In comparison, a state–of–the–art visual SLAM approach (PTAM [Klein and Murray, 2007]) is not suitable for such a large–scale outdoor environment, and achieves this accuracy only in 22% of all frames. (c) The overlay of both paths shows that PTAM has lost track of its initial map at some point during the flight.

the relative pose cannot be established anymore and the localization fails.

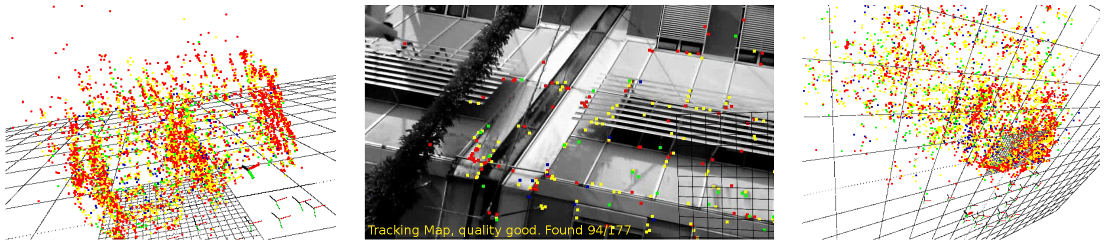
6.2.5 Comparison to Ground Truth

We take the video stream showing the quad–rotor's flight from a ground observer's viewpoint as visual ground truth. It shares the virtual cameras, so we can localize the video within our scene using our localization method as described in Chapter 5. Under the premise of synchronized video streams and a correct current pose estimate from the quad–rotor's flight and ground observing video, the backprojection of the MAV camera center must coincide with the quad–rotor position



(a) (b) (c)

Figure 6.10: (a) Registration performance showing the fraction of successful pose estimates after selecting k top–scoring views using the vocabulary tree. (b) Histogram over the number of effective inliers. (c) Distribution of the mean reprojection error after robust absolute pose estimation.



(a) (b) (c)

Figure 6.11: Results for scene reconstruction and localization using PTAM [Klein and Murray, 2007]. (a) This approach works well within a locally constrained environment such as at the beginning of our sequence. (b) However, when the track is lost wrong measurements are added to the map and hinder proper re–localization. (c) In the resulting model, the correct part can be seen to the lower right of the point cloud.

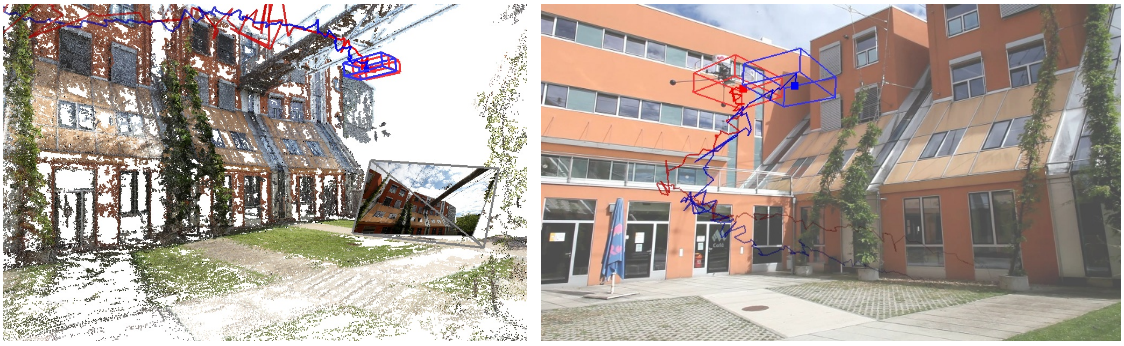
in the observer's view. This is a concept borrowed from augmented reality, that we use for the purpose of fully 3D geometric validation. Figure 6.12(a) shows the tracked 3D quad–rotor and the respective ground video frame. In particular, we use the widespread PASCAL criterion [Everingham et al., 2007] and consider the backprojected quad–rotor bounding box $\mathcal{B}_Q$ as successfully registered if the criterion,

$$\frac{\mathcal{B}_Q \bigcap \mathcal{B}_G}{\mathcal{B}_Q \bigcup \mathcal{B}_G} > 0.5 \tag{6.2}$$

is satisfied. $\mathcal{B}_G$ denotes the bounding box of the quad–rotor in the ground truth view. To this end we take every tenth frame of the registered ground video and evaluate Equation 6.2 by visual inspection. For our localization approach the 50% bounding box overlap criterion is satisfied for 92% of the frames, whereas PTAM achieves 22%, only. A subset of frames used for evaluation is shown in Figure 6.13. While the errors of 3D modeling, MAV localization, and ground truth video localization accumulate, the qualitative evaluation of the localization shows accurate localization results. Given the size of the bounding box with $1.0m \times 1.0m$, our localization approach has an error of less than $0.5m$ in 92% of all frames and can be considered more accurate than consumer grade GPS. Under the premise of exact convergence of the absolute pose algorithm, the achievable heading precision ϵ_α of the MAVs attitude can be estimated from the image width w, field of view α_{FOV} and the mean reprojection error $\epsilon_{rep.}$,

$$\epsilon_\alpha = \frac{\alpha_{FOV}}{w} \epsilon_{rep.} \ . \tag{6.3}$$

This translates into a precision of $\sim 0.1^\circ$ for a camera of resolution 848×480 with $\alpha_{FOV} = 64^\circ$ and an average reprojection error of $\epsilon_{rep.} = 1.7$ pixel. Figure 6.12 depict sample frames showing the augmented MAV position determined by the two approaches.


(a)

(b)

Figure 6.12: Visual comparison of the localization results via registered ground truth video stream. (a) Oblique view of registered ground truth showing the current camera frame and MAV positions (path and bounding box) estimated by our localization approach (red) and the aligned PTAM result (blue). (b) Back–projected bounding box from the observer's point of view.

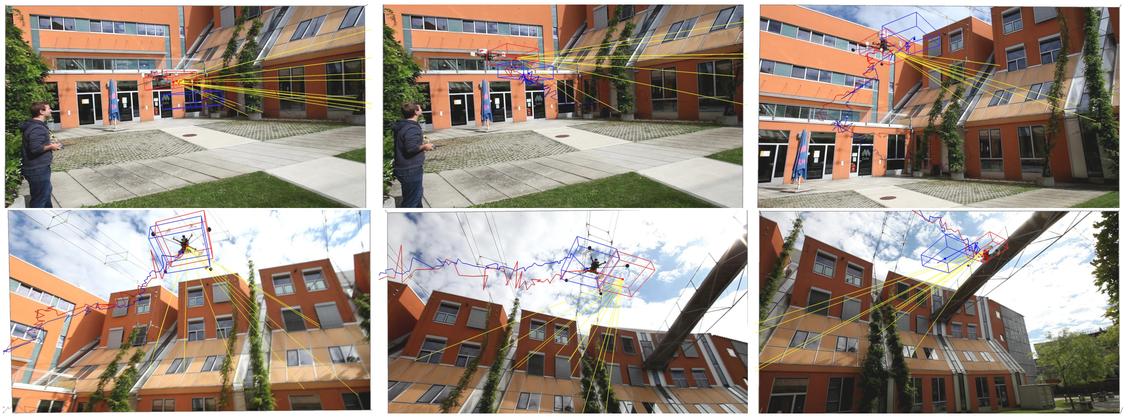


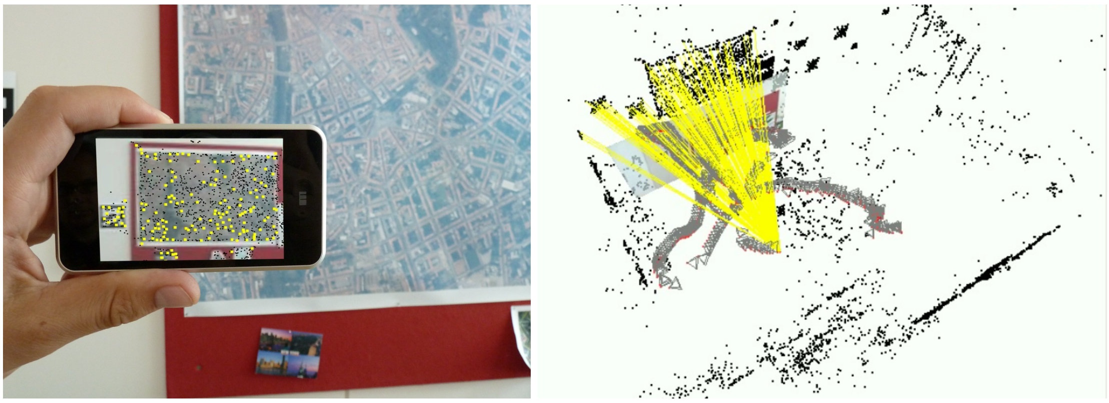
Figure 6.13: Ground truth evaluation of the MAVs pose using the PASCAL criterion.

6.2.6 Conclusion and Discussion

We introduced a novel algorithm for monocular visual localization for MAVs. Our work is based on the concept of virtual views in 3D space and it exploits the knowledge about possible neighboring views resulting from the sequence of images, which improves robustness and scalability. Under the assumption that significant parts of the scene do not alter their geometry and serve as natural landmarks, our approach outperforms conventional GPS systems in an outdoor environment. Within the atrium scene used in our experiments, we achieve an error of less than 0.5m in 92% of all frames. We significantly outperform the PTAM approach used for comparison because our localization method directly allows global registration and is neither prone to drift nor bias. This makes it well suited for long–term visual outdoor navigation. Future work should tackle the problem of close–up and high–attitude views by refining the rough model with dynamic visual data. This would also allow to generate models which are robust to weather or even seasonal changes. Additionally, it would be important to experiment with descriptors which require less computational power than SIFT, so that an implementation of the algorithm on–board an MAV becomes feasible.

6.3 Wide Area Localization on Mobile Phones

Full 6DOF pose estimation of a mobile phone with respect to a 3D model is useful and desired for location based applications in Augmented Reality [Greene, 2006]. In general the accuracy of Global Positioning Systems (GPS) and Inertial Measurement Units (IMU) is limited and does often not satisfy the accuracy needed for visual tracking [Schall et al., 2009]. Therefore, robust and accurate image registration methods are needed for a visually convincing integration of the 3D content and images of the real world. In [Arth et al., 2009] we demonstrate a modified version of the synthetic view localization approach (see Chatper 5) which is suitable to run on a mobile phone. Given a 3D reconstruction of the scene, the pose of the mobile phones camera is determined from 2D-3D feature correspondences. This information can be used to initialize a real-time pose tracker suitable for augmented reality. Figure 6.14 shows such a localization result that runs on a mobile phone in an office environment. While the 3D reconstruction is done offline using high resolution images from a SLR camera and a desktop workstation, the view registration algorithm runs fully on the mobile device. The limited computational power and memory size of a mobile device compared to a desktop computing systems requires some substantial algorithmic modifications. The core component of the system is a new and efficient feature detector/descriptor inspired by SURF [Bay et al., 2008] that runs at $380ms$ for images of resolution 640×480 on the *Meizu M8*[1] (800MHz ARMII CPU with FPI). Tests done with the framework of [Mikolajczyk and Schmid, 2005] show that it performs as well as SIFT and SURF in terms of keypoint repeatability and sometimes outperforms both. Second, a Potentially Visible Set (PVS) [Airey et al., 1990] approach is used to split the database into compact chunks that can be loaded independently into the main memory.



(a)

(b)

Figure 6.14: Camera pose tracking using the *Meizu M8* smart-phone in an indoor environment. Yellow lines depict inlier of 3D-2D point correspondences.

[1] http://meizu.com

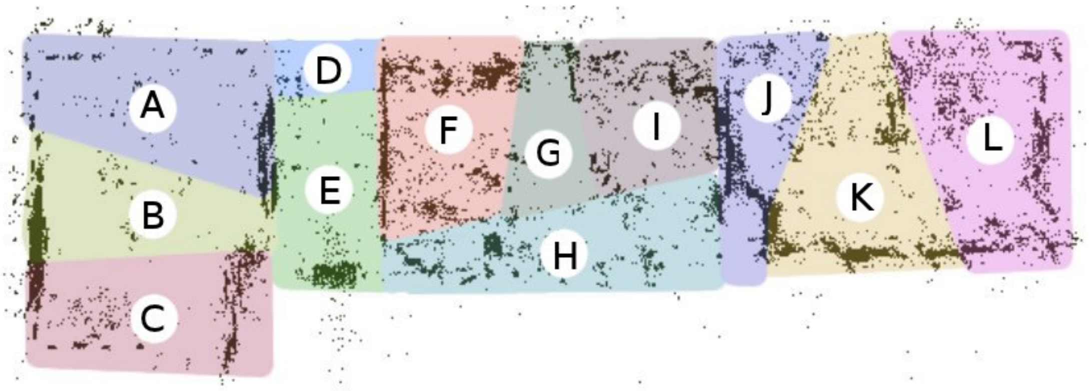
6.3.1 Potentially Visible Sets

In order to discretize the environment into manageable subsets of 3D points, we borrow a concept from computer graphics known as Potentially Visible Sets (PVS), that is often used for occlusion culling. The basic idea is to partition the environment into view cells and precompute the cell-to-cell visibility. In densely occluded environments, such as hilly regions, urban areas or building interiors, the potentially visible sets significantly reduce the amount of data that has to be processed for rendering. Indoors, the natural structure of cells (rooms) and portals (doorways) can be exploited [Teller and Séquin, 1991]. This concept is closely related to the method described Takacs et al. [Takacs et al., 2008] that suggest to partition the database into a 2D regular grid, each node holds the features sets of the closest 3x3 cells in memory. In contrast, we split the database by visibility, using a PVS structure. Every potentially visible set consists of a number of cells. Figure 6.15(a) shows a cell partitioning for an indoor reconstruction and depicts the respective precomputed cell-to-cell visibility as a graph representation.

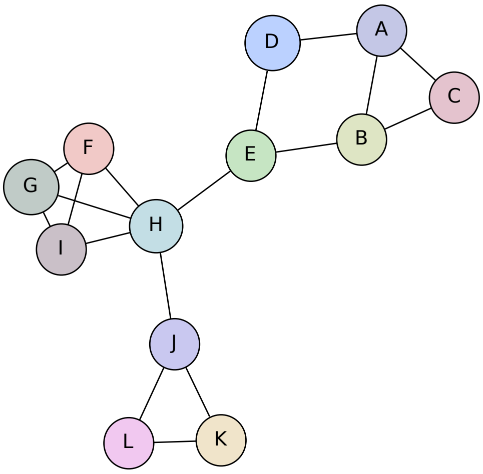
For localization, only the features of the current PVS have to be in memory, hence the amount of memory is considerably smaller than the amount needed for the entire area. Furthermore, as can be seen from Table 6.15(c) some PVS are redundant (e.g. PVS 2 and PVS 3) and several cells are shared between adjacent PVS. Hence, only a fraction of cells has to be loaded from memory if a transition from one location to the next occurs. A custom memory management loads and discards feature blocks associated to the current PVS structure. Feature blocks (corresponding to cells) are loaded on demand when a PVS requests them. On the other hand, when a cell is no longer required by any PVS, the memory manager discards it to free. This concept significantly reduces the memory footprint that is necessary for localization. In our experiments a feature block typically has a memory footprint of 1-2MB and a PVS is build from 2-5 cells. This translates into an overall memory footprint of about $5MB$, which is small enough to fit into a mobile phone's application memory.

6.3.2 Localization

Localizing a mobile users position involves the following steps: feature and descriptor extraction, feature matching, outlier removal, and finally pose estimation and refinement. Feature extraction uses a scale space search to find keypoints in the 2D image, including a size estimation. For each keypoint, we estimate a single dominant orientation and create one descriptor. The scale space search step dominates the resource requirements, taking about 80% of the computation time and requires roughly 12 bytes per camera image pixel. The memory overhead for creating descriptors is relatively low with about 0.3 bytes per camera image pixel and ~80 bytes per feature. We implemented two alternative methods for feature matching: in matching-friendly scenarios, we directly match all camera image features against all features in the current PVS. Alternatively, we use a vocabulary tree voting scheme: We first define subsets by finding those images from the reconstruction step that contain enough features that match the current camera image. We then match against the top ranked subsets separately. For each subset we then try to estimate a pose. The advantage of this two step approach is that we largely reduce the number of features to match



(a)



loc	*name*	*cells*
A	PVS 1	$\{A, B, C, D\}$
B	PVS 2	$\{A, B, C, E\}$
C	PVS 3	$\{A, B, C, E\}$
D	PVS 4	$\{A, D, E\}$
E	PVS 5	$\{B, D, E, H\}$
F	PVS 6	$\{F, G, H, I\}$
G	PVS 7	$\{F, G, H, I\}$
H	PVS 8	$\{E, F, G, H, I, J\}$
I	PVS 9	$\{F, G, H, I\}$
J	PVS 10	$\{H, J, K, L\}$
K	PVS 11	$\{H, J, K, L\}$
L	PVS 12	$\{J, K, L\}$

(b) (c)

Figure 6.15: (a) Representation of a corridor reconstruction as 12 separate PVS cells. The assignment of features to cells is color coded.(b) Potentially Visible Set representation of the scene, dependent on the current location *loc*.

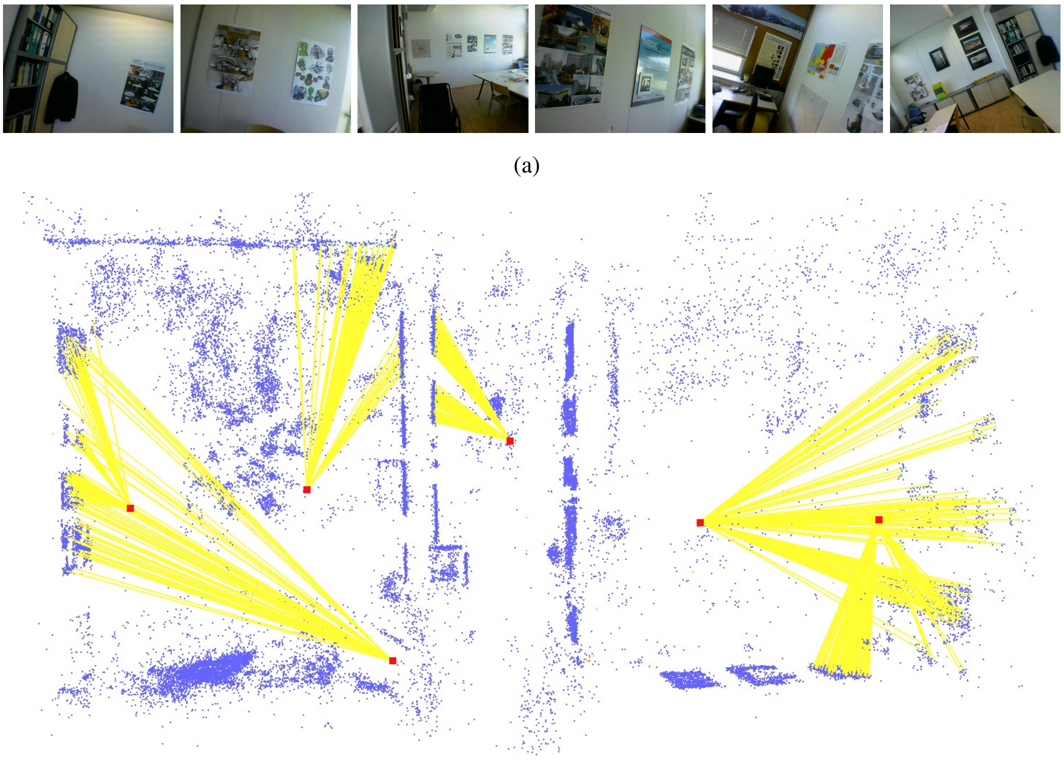
against, which makes the matching itself more robust. However, it has higher computational requirements. In both cases, matching the camera image against the dataset gives a set of 2D-3D correspondences that still includes outliers. A robust pose estimation procedure is therefore required to deal with these outliers. We therefore apply a RANSAC scheme with a Three-Point pose [Haralick et al., 1991] as hypothesis and use a subset of up to 50 correspondences for validation. The Three-Point pose estimation is based on a fixed-point implementation of the method in [Fischler and Bolles, 1981]. The hypothesis with the largest number of inliers is selected as a starting point for a non-linear refinement. Based on the inlier set of the best hypothesis, we apply an M-estimator in a Gauss-Newton iteration to refine the pose and find more inliers. This step is repeated until the inlier set does not grow anymore. In theory, four points are enough to calculate a 6DOF pose from known 2D-3D correspondences. However, given a large enough number of outliers, it is likely to find an invalid pose from a small number of correspondences only. We therefore treat a pose only as valid if at least 20 inliers were found.

6.3.3 Experiments

Our experimental setup consists of a corridor reconstruction from high resolution SLR images. We use the *Meizu M8* smartphone for localization. The mobile phones internal camera delivers still images with a maximal resolution of 1920×1440 pixels. The camera is calibrated and the lens distortion is adjusted to meet a pinhole camera model. Operating on full resolution images provides the most accurate localization results but is computationally demanding. For augmented reality on mobile phones meeting the target resolution of the display is sufficient for pixel accurate alignment. Hence, we can operate on downsampled images (i.e.720×480). Figure 6.16 shows sample images from our test scene and respective localization results. A quantitative evaluation of the reprojection error is depicted in Figure 6.17. One can observe that almost 80% of all inliers used for calculating the pose have a reprojection error smaller than 4 pixel.

6.3.4 Conclusion and Discussion

We presented an approach for wide-area 6DOF pose estimation that runs on a current smartphone at about 2-3Hz. It relies on a previously acquired 3D feature model, which can be generated from image collections, and can therefore tap into the rapidly increasing amount of real world imagery acquired for digital globe projects and similar ventures. To make the approach scalable, a representation inspired by potentially visible set techniques was adopted together with a feature representation that is suitable to work in real time on a mobile phone.



(a)

(b)

Figure 6.16: (a) Sample images from the test set and (b) respective localization results. The yellow lines show the view rays of inlier used for localization.

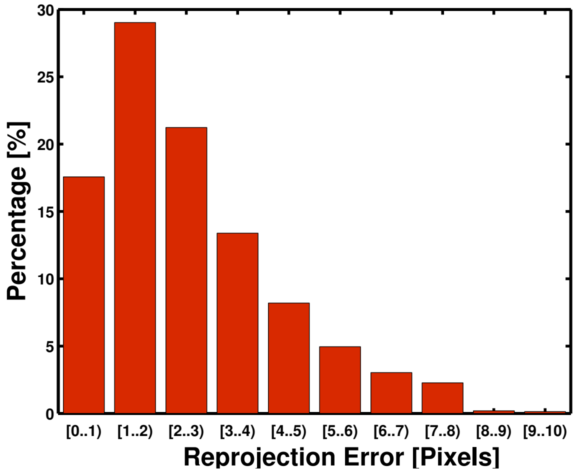


Figure 6.17: Reprojection error for inliers with respect to a distance threshold of 10 pixels.

Chapter 7

Conclusions

In this thesis two main problems in computer vision have been addressed: robust and scalable structure from motion and efficient localization from images. The main contribution of this thesis is in building a reconstruction and localization system that can be applied to large scale real world problems. Different algorithms and methods were presented that allow fully automatic scene reconstruction from unordered image collections. Contributions were made to several system components including image matching, geometric verification, structure from motion estimation and dense matching. We introduced an algorithm for non-monotone reasoning about view triplets which enables to identify wrong epipolar geometries from the matching graph. Our algorithm is able to infer and remove such erroneous view pairs. Furthermore, our method can be easily integrated into existing incremental structure from motion pipelines and is able to handle duplicate scene structures up to some degree.

In our reconstruction system we successfully leverage the highly parallel computing power of current graphic processing units to sped up various processing steps. This includes GPU based vocabulary tree traversal, feature matching and geometric verification. Most of the methods are generic and can be used for different application domains. In particular we presented a Wiki-based approach for user contributed dense city modeling. A novel calibration method based on planar markers allows accurate camera calibration and is easy to apply for end-users. Furthermore, we introduced a guided view selection approach based on external pose priors. Our system advances vocabulary tree based coarse matching by imposing additional constraints on the geometry. This approach is especially useful for 3D modeling from images taken by micro aerial vehicles that provide GPS/IMU support. In addition an end-to-end workflow for aerial dense matching was presented. Instead of pairwise stereo fusion, our method is based on multi-view plane sweep with global optimization on a 3D voxel-space. The algorithm delivers globally optimal solutions with respect to the minimized energy but is computationally very intensive.

Given a sparse 3D reconstruction of a scene, we introduced a novel method for image based real-time scene recognition. Our proposed approach is able to efficiently register images/videos to large 3D point clouds as computed by structure from motion techniques. The method performs real-time tracking by detection. Since every frame is individually matched and compared to a global 3D model database, the system is able to automatically recover from tracking failures. The core component of our system is a fast indexing method based on a compressed set of synthetic views. These synthetic views correspond to sampled 3D point fragments that are globally indexed through a vocabulary tree and inverted file structure. Our framework achieves excellent registration rates and is currently among the most efficient approaches for 6DOF location recognition. We introduced a framework for fast registration of community photo collections to known landmark reconstructions. The algorithm is suitable for outdoor robot localization, outperforming state of the art SLAM approaches. Furthermore, a variation of our localization framework is capable to run on modest hardware such as smart-phones that allows hand-held augmented reality. The employed algorithms are generic and extensively evaluated on a variety of different datasets. Our experiments demonstrated robustness, scalability and high geometric accuracy of the proposed algorithms.

7.1 Directions for Future Work

In this thesis various algorithms and methods for efficient and scalable image based 3D reconstruction have been presented. The ultimate goal would be a system that is able to determine the exact 3D location in space of each captured pixel in each image/video and the orientation of every image taken on Earth. Such a system would require fusing all available image data into one detailed and consistent global 3D model of the human habitat in three-dimensions and in time. Of course, this can only be achieved if a certain degree of texture information about the scene and some prior information about the position of each image is available. Despite these prerequisites, there are still unresolved problems in image based 3D modeling and localization.

Scalability True scalability of image based 3D reconstruction and localization methods is still not achievable. Although, we observe an ever increasing amount of computing power due to the emerge of multi-core CPUs and graphic processing units (GPUs), real-time performance for the task of unordered structure from motion and dense matching of large aerial images is currently out of reach. The minimization of the reprojection error of all images taken at the scale of the whole world is practically impossible. Additional external orientation information is necessary to partition the data into manageable subsets. While this problem can be solved outdoors by using GPS sensors, we are not aware of any adequate indoor localization approach that can be easily deployed. This is also true for large scale image based localization systems.

Robustness Beside scalability, the robustness of image based 3D reconstruction and localization is another challenge. Even though current structure from motion algorithms work reasonably well on a large variety of input data, there are still some scenes or special camera configurations where these algorithms fail. In Chapter 4 we have presented an algorithm that is able to detect inconsistent geometric relations between pairs of images. The employed prior probabilities on the number of detected correspondences is based on a rather simplistic model and wrong 3D reconstructions can still occur. Some assumption made in the proposed framework are strong and future work should address relaxing some of these. Furthermore, computing image correspondences between views that are widely separated in time and space is an ill-posed problem. Current feature detectors and descriptors are only invariant to a certain degree of view point change and illumination variations. We are not aware of any robust algorithm for matching aerial and terrestrial images with a view point difference close to 90°. In general the reconstruction problem (with missing data and outliers) itself is NP-hard [Nister et al., 2007]. The reconstruction of poorly textured scenes or scenes including repetitive structures is another open research problem.

Sensor Fusion The fusion of image based data with other sensory information like global positioning systems (GPS), Inertial Measurement Units (IMU), Light Detection and Ranging (LiDAR), odometry and sonar data is an important aspect for global 3D modeling. Using different sensors forms the basis for an optimal geo-spatial data fusion since different properties of objects are recorded, based on different physical principles of the sensors, bringing together complementary and often redundant information. The fusion of this heterogeneous data implies new challenges of calibration, accuracy, precision and data representation. Today, a large amount of geographic information is already freely available and can be accessed from the web. Especially collaborative online community projects such as OpenStreetMap[1] (OSM) provide detailed 2D vector data of street networks and building outlines. The massive amount of shared geo-referenced data can be exploited for world scale reconstruction and localization tasks.

Camera Network Design In order to achieve a certain accuracy in the 3D reconstructions, photogrammetric network planning is important for image based modeling. While this is a well known task in aerial photogrammetry, only recently this problem gained attention for terrestrial and close range reconstruction [Schmid et al., 2012]. In manned aerial image acquisition, the network design is normally restricted to a 2D camera grid. In Section 3.2.3 we observed that the flight height and the forward and side-ward image overlap mainly determine the accuracy measured in terms of ground sampling distance. This strategy is suitable for reconstructing the bold earth's surface, but for more complex structures like urban buildings, more sophisticated methods are necessary. In particular, camera network design becomes more and more important due to new sensor platforms like micro aerial vehicles (see Chapter 3.3). These systems are able to fly close to the object of interest and deliver high resolution images which allow high quality reconstructions. Nevertheless, this views must be planned carefully to adhere constraints like sufficient image overlap and maximum angle between images to facilitate vision based similarity computations. Furthermore, the image overlap graph should be fully connected, otherwise disjoint reconstructions are obtained. This includes minimizing a multi objective function. Finding an optimal viewpoint plan is an NP-complete problem and therefore hard to optimize. The design and optimization of such a problem is a challenging task.

[1] www.openstreetmap.org

Handling Time Reconstruction approaches from images normally assume a rigid scene that does not change during the acquisition process. However, this assumption is often violated in real world scenarios since the appearance of our environment undergoes a permanent change. For instance seasonal vegetation changes, wind and weather effects or the successive reduction of glaciers over decades leads to significant appearance changes of the environment. This is especially true for urban scenes with movable objects or the reconstruction and renovation of buildings. Dealing with such kind of time dependent data and monitoring changes in existing 3D models is an interesting field of future research.

Semantic Interpretation Image based reconstruction methods result in camera orientations and dense or semi dense point clouds. For visualization aspects, a point cloud might be a sufficient representation but often semantic information is necessary in order to extract interpretable information for humans. For instance, automatically deriving a floor plan from a 3D point cloud of a building is still a hard problem. Holistic scene understanding is one of the major problems in computer vision and photogrammetry and has recently got a lot of attention [Ladicky et al., 2010]. This includes two fundamental tasks: 3D scene reconstruction and semantic interpretation of the imaged content. The tight interaction between semantic classification and 3D reconstruction is often ignored by current reconstruction and localization systems. However, these tasks are mutually informative and should be solved jointly. Semantic information from images can be used to guide reconstruction methods and the 3D information on the other hand can help to improve the semantic interpretation.

Appendix A

Publications

The publications crated during the course of this thesis are grouped by topic and roughly sorted by date.

A.1 Structure from Motion

- *Large Scale, Dense City Reconstruction from User-Contributed Photos.* Arnold Irschara, Christopher Zach, Manfred Klopschitz, and Horst Bischof. Journal Computer Vision and Image Understanding (CVIU), 2011.

- *Efficient Structure from Motion with Weak Position and Orientation Priors.* Arnold Irschara, Christoph Hoppe, Horst Bischof, and Stefan Kluckner. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Aerial Video Processing, 2011.

- *Automatic Alignment of 3D Reconstructions using a Digital Surface Model.* Andreas Wendel, Arnold Irschara, and Horst Bischof. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Aerial Video Processing, 2011.

- *3D Vision Applications for MAVs: Localization and Reconstruction.* Andreas Wendel, Michael Maurer, Arnold Irschara, and Horst Bischof. Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), 2011.

- *Towards Fully Automatic Photogrammetric Reconstruction Using Digital Images Taken From UAVs.* Arnold Irschara, Viktor Kaufmann, Manfred Klopschitz, Horst Bischof, and Franz Leberl. Proceedings of the International Society for Photogrammetry and Remote Sensing Symposium, 100 Years ISPRS - Advancing Remote Sensing Science, 2010.

- *Robust Incremental Structure from Motion.* Manfred Klopschitz, Arnold Irschara, Gerhard Reitmayr, and Dieter Schmalstieg. Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), 2010.

- *Kollaborative 3D Rekonstruktion von urbanen Gebieten.* Arnold Irschara, Christopher Zach, Horst Bischof und Franz Leberl. 15. Internationale geodätische Woche Obergurgl, 2009.

- *What Can Missing Correspondences Tell Us About 3D Structure and Motion?* Christopher Zach, Arnold Irschara and Horst Bischof. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

- *Generalized Detection and Merging of Loop Closures for Video Sequences.* Manfred Klopschitz, Christopher Zach, Arnold Irschara and Dieter Schmalstieg. Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), 2008.

- *Towards Wiki-based Dense City Modeling.* Arnold Irschara, Christopher Zach and Horst Bischof. In Proceedings of the IEEE International Conference on Computer Vision, Workshop on Virtual Representations and Modeling of Large-scale environments (VRML), 2007.

A.2 Photogrammetry

- *Photogrammetric Camera Network Design for Micro Aerial Vehicles.* Christof Hoppe, Andreas Wendel, Stefanie Zollmann, Katrin Pirker, Arnold Irschara, Horst Bischof and Stefan Kluckner. In Proceedings of the 17th Computer Vision Winter Workshop (CVWW), 2012.

- *Rapid 3D City Model Approximation from Publicly Available Geographic Data Sources and Georeferenced Aerial Images.* Markus Rumpler, Arnold Irschara, Andreas Wendel and Horst Bischof. In Proceedings of the 17th Computer Vision Winter Workshop (CVWW), 2012.

- *Multi-View Stereo: Redundancy Benefits for 3D Reconstruction.* Markus Rumpler, Arnold Irschara, and Horst Bischof. Proceedings of the 35th Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM), 2011.

- *Aerial Computer Vision for a 3D Virtual Habitat.* Franz Leberl, Horst Bischof, Thomas Pock, Arnold Irschara, and Stefan Kluckner. IEEE Computer Society, 2010.

- *Point Clouds: Lidar versus 3D Vision.* Franz Leberl, Arnold Irschara, Thomas Pock, Philipp Meixner, M. Gruber, S. Scholz and A. Wiechert. Photogrammetric Engineering and Remote Sensing, 2010.

A.3 Image Based Localization

- *Natural Landmark-based Monocular Localization for MAVs.* Andreas Wendel, Arnold Irschara, and Horst Bischof. International Conference on Robotics and Automation (ICRA), 2011.

- *Wide Area Localization on Mobile Phones.* Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara, and Dieter Schmalstieg. IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2009.

- *From Structure-from-Motion Point Clouds to Fast Location Recognition.* Arnold Irschara, Christopher Zach, Jan-Michael Frahm, Horst Bischof. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

Bibliography

[Agarwal et al., 2010] Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle adjustment in the large. In *European Conference on Computer Vision (ECCV)*, pages 29–42.

[Agarwal et al., 2009] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building Rome in a day. In *IEEE International Conference on Computer Vision (ICCV)*.

[Airey et al., 1990] Airey, J. M., Rohlf, J. H., and Brooks Jr., F. (1990). Towards image realism with interactive update rates in complex virtual building enviroments. *Computer Graphics*, 24(2):41.

[Alcantarilla et al., 2011] Alcantarilla, P. F., Ni, K., Bergasa, L. M., and Dellaert, F. (2011). Visibility learning in large-scale urban environment. In *International Conference on Robotics and Automation (ICRA)*, pages 6205–6212. IEEE.

[Anan and Hartley, 2008] Anan, C. S. and Hartley, R. I. (2008). Optimised KD-trees for fast image descriptor matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Arth et al., 2009] Arth, C., Wagner, D., Klopschitz, M., Irschara, A., and Schmalstieg, D. (2009). Wide area localization on mobile phones. In Klinker, G., Saito, H., and Höllerer, T., editors, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 73–82. IEEE Computer Society.

[Azuma et al., 1999] Azuma, R., Hoff, B., Neely, H., and Sarfaty, R. (1999). A motion-stabilized outdoor augmented reality system. In *VR*, pages 252–259.

[Baltsavias, 1999] Baltsavias, E. P. (1999). A comparison between photogrammetry and laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):83–94.

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

[Beardsley et al., 1996] Beardsley, P. A., Torr, P. H. S., and Zisserman, A. (1996). 3D model acquisition from extended image sequences. In *European Conference on Computer Vision (ECCV)*, pages II:683–695.

[Beder and Steffen, 2006] Beder, C. and Steffen, R. (2006). Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Proc. DAGM*, pages 657–666.

[Beis and Lowe, 1997] Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1006.

[Belhumeur, 1996] Belhumeur, P. N. (1996). A bayesian-approach to binocular stereopsis. *Int. Journal of Computer Vision*, 19(3):237–260.

[Bloesch et al., 2010] Bloesch, M., Weiss, S., Scaramuzza, D., and Siegwart, R. (2010). Vision based mav navigation in unknown and unstructured environments. In *International Conference on Robotics and Automation (ICRA)*.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

[Brown and Lowe, 2005] Brown, M. and Lowe, D. G. (2005). Unsupervised 3D object recognition and reconstruction in unordered datasets. In *3DIM*, pages 56–63. IEEE Computer Society.

[Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary robust independent elementary features. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV)*, volume 6314, pages 778–792. Springer.

[Castle et al., 2008] Castle, R. O., Klein, G., and Murray, D. W. (2008). Video-rate localization in multiple maps for wearable augmented reality. In *Proc 12th IEEE Int Symp on Wearable Computers, Pittsburgh PA, Sept 28 - Oct 1, 2008*, pages 15–22.

[Chen and Li, 2004] Chen, S. Y. and Li, Y. F. (2004). Automatic sensor placement for model-based robot vision. *IEEE Trans. Systems, Man and Cybernetics*, 34(1):393–408.

[Chen et al., 2008] Chen, S. Y., Li, Y. F., Zhang, J. W., and Wang, W. L. (2008). *Active Sensor Planning for Multiview Vision Tasks*. Springer-Verlag.

[Chum et al., 2003] Chum, O., Matas, J., and Kittler, J. V. (2003). Locally optimized RANSAC. In *DAGM*, pages 236–243.

[Chum et al., 2011] Chum, O., Mikulík, A., Perdoch, M., and Matas, J. (2011). Total recall II: Query expansion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–896. IEEE.

[Chum et al., 2009] Chum, O., Perd'och, M., and Matas, J. G. (2009). Geometric min-hashing: Finding a (thick) needle in a haystack. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24.

[Chum et al., 2007] Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Colomina et al., 2008] Colomina, I., Blázquez, M., Molina, P., Parés, M., and Wis, M. (2008). Towards a new paradigm for high-resolution low-cost photogrammetry and remote sensing. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVII, Part B1, pages 1201–1206.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5).

[Cornelis and Van Gool, 2005] Cornelis, N. and Van Gool, L. (2005). Real-time connectivity constrained depth map computation using programmable graphics hardware. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1099–1104.

[Crandall et al., 2011] Crandall, D. J., Owens, A., Snavely, N., and Huttenlocher, D. (2011). Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3001–3008. IEEE.

[Curless and Levoy, 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH '96*, pages 303–312.

[Dalalyan and Keriven, 2009] Dalalyan, A. and Keriven, R. (2009). L1-penalized robust estimation for a class of inverse problems arising in multiview geometry. In *23d Annual Conference on Neural Information Processing Systems*, Vancouver, Canada.

[Davis, 2006] Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. Fundamentals of Algorithms. SIAM, Philadelphia.

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

[Devernay and Faugeras, 2001] Devernay, F. and Faugeras, O. D. (2001). Straight lines have to be straight. *Mach. Vis. Appl*, 13(1):14–24.

[Drineas et al., 2004] Drineas, P., Frieze, A. M., Kannan, R., Vempala, S., and Vinay, V. (2004). Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33.

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

[Eade and Drummond, 2008] Eade, E. D. and Drummond, T. W. (2008). Unified loop closing and recovery for real time monocular SLAM. In *British Machine Vision Conference (BMVC)*.

[Eissenbeiss et al., 2009] Eissenbeiss, H., Nackaerts, K., and Everaerts, J. (2009). UAS for mapping & monitoring applications. In *2009/2010 UAS Yearbook - UAS: The Global Perspective, 7th Edition*, pages 146–150.

[Engels et al., 2006] Engels, C., Stewenius, H., and Nister, D. (2006). Bundle adjustment rules. In *Photogrammetric Computer Vision*, pages 1–6.

[Everingham et al., 2007] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[Farr et al., 2007] Farr, T. G. et al. (2007). The shuttle radar topography mission. Technical report, Rev. Geophys., 45, RG2004.

[Faugeras and Luong, 2001] Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. The MIT Press, Massachusets Institute of Technology, Cambridge, Massachusetts 02142.

[Faugeras and Lustman, 1988] Faugeras, O. D. and Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:485–508.

[Ferrari et al., 2003] Ferrari, V., Tuytelaars, T., and Van Gool, L. (2003). Wide-baseline multiple-view correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 718–725.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communication Association and Computing Machine*, 24(6):381–395.

[Fitzgibbon and Zisserman, 1998] Fitzgibbon, A. W. and Zisserman, A. (1998). Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision (ECCV)*, page I: 311.

[Förstner and Gülch, 1987] Förstner, W. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. *Proc. of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data, Interlaken*, pages 285–301.

[Förstner and Steffen, 2007] Förstner, W. and Steffen, R. (2007). Online geocoding and evaluation of large scale imagery without GPS. In Fritsch, D., editor, *Photogrammetric Week '07, Heidelberg*, pages 243–253.

[Frahm et al., 2010] Frahm, J.-M., Georgel, P. F., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., and Lazebnik, S. (2010). Building rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, volume 6314, pages 368–381.

[Frahm and Pollefeys, 2006] Frahm, J. M. and Pollefeys, M. (2006). RANSAC for (quasi-) degenerate data (QDEGSAC). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I: 453–460.

[Frandsen et al., 1999] Frandsen, P. E., Jonasson, K., Nielsen, H. B., and Tingleff, O. (1999). Unconstrained optimization.

[Fukunaga and Narendra, 1975] Fukunaga, K. and Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers*, C-24(7):750–753.

[Furukawa and Ponce, 2009] Furukawa, Y. and Ponce, J. (2009). Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

[Gallup et al., 2008] Gallup, D., Frahm, J.-M., and Pollefeys, M. (2008). Variable baseline/resolution stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Geiger et al., 1995] Geiger, D., Ladendorf, B., and Yuille, A. L. (1995). Occlusions and binocular stereo. *Int. Journal of Computer Vision*, 14(3):211–226.

[Gherardi et al., 2010] Gherardi, R., Farenzena, M., and Fusiello, A. (2010). Improving the efficiency of hierarchical structure-and-motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1594–1600. IEEE.

[Goesele et al., 2007] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision (ICCV)*.

[Golub and van Loan, 1996] Golub, G. H. and van Loan, C. F. (1996). *Matrix computations (3. ed.)*. Johns Hopkins University Press.

[Gordon and Lowe, 2006] Gordon, I. and Lowe, D. G. (2006). What and where: 3D object recognition with accurate pose. In *CLOR06*, pages 67–82.

[Govindu, 2001] Govindu, V. M. (2001). Combining two-view constraints for motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 218–225. IEEE Computer Society.

[Govindu, 2004] Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 684–691.

[Greene, 2006] Greene, K. (2006). Hyperlinking reality via phones. *MIT Technology Review*.

[Grenzdörffer et al., 2008] Grenzdörffer, G., Engel, A., and Teichert, B. (2008). The photogrammetric potential of low-cost uavs in forestry and agriculture. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVII, Part B1, pages 1207–1214.

[Gruen and Akca, 2007] Gruen, A. and Akca, D. (2007). Mobile photogrammetry. In *DGPF Tagungsband 16 / 2007 - Dreilaendertagung SGPBF, DGPF und OVG*.

[Gruen and Akca, 2008] Gruen, A. and Akca, D. (2008). Metric accuracy testing with mobile phone cameras. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVII, Part B5, pages 729–736.

[Haralick et al., 1991] Haralick, R. M., Lee, C., Ottenberg, K., and Nölle, M. (1991). Analysis and solutions of the three point perspective pose estimation problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 592–598.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings 4th Alvey Visual Conference*, pages 189–192.

[Hartley, 1995] Hartley, R. (1995). A linear method for reconstruction from points and lines. In *IEEE International Conference on Computer Vision (ICCV)*, pages 882–887.

[Hartley and Zisserman, 2000] Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[Hartley, 1997] Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(6):580–593.

[Hartley and Schaffalitzky, 2004] Hartley, R. I. and Schaffalitzky, F. (2004). L_∞ minimization in geometric reconstruction problems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I: 504–509.

[Havlena et al., 2009] Havlena, M., Torii, A., Knopp, J., and Pajdla, T. (2009). Randomized structure from motion based on atomic 3D models from camera triplets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2881.

[Havlena et al., 2010] Havlena, M., Torii, A., and Pajdla, T. (2010). Efficient structure from motion by graph optimization. In *European Conference on Computer Vision (ECCV)*, pages 1–8. Springer.

[Heikkilä, 2000] Heikkilä, J. (2000). Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(10):1066–1077.

[Hernández, 2004] Hernández, C. (2004). *Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion*. PhD thesis, Ecole Nationale Supŕieure des Télécommunications.

[Hopcroft and Tarjan, 1973] Hopcroft, J. and Tarjan, R. (1973). Efficient algorithms for graph manipulation. *Communications of the ACM*, 16:372–378.

[Horn and Schunck, 1981] Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Hornung and Kobbelt, 2006] Hornung, A. and Kobbelt, L. (2006). Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Eurographics Symposium on Geometry Processing*, pages 41–50.

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

[Hutton and Mostafa, 2005] Hutton, J. and Mostafa, M. M. (2005). 10 years of direct goereferencing for airborne photogrammetry. In *Photogrammetric Week*.

[Irschara et al., 2011] Irschara, A., Hoppe, C., Bischof, H., and Kluckner, S. (2011). Efficient structure from motion with weak position and orientation priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Aerial Video Processing*.

[Irschara et al., 2007] Irschara, A., Zach, C., and Bischof, H. (2007). Towards wiki-based dense city modeling. In *Workshop on Virtual Representations and Modeling of Large-scale environments (VRML)*.

[Ishikawa, 2003] Ishikawa, H. (2003). Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(10):1333–1336.

[Johnson, 1974] Johnson, D. S. (1974). Approximation algorithms for combinatorial problems. *J. of Comput. System Sci.*, 9:256–278.

[Kahl, 2005] Kahl, F. (2005). Multiple view geometry and the L_∞-norm. In *IEEE International Conference on Computer Vision (ICCV)*, pages II: 1002–1009.

[Kamberov et al., 2006] Kamberov, G., Kamberova, G., Chum, O., Obdrzalek, S., Martinec, D., Kostkova, J., Pajdla, T., Matas, J., and Sara, R. (2006). 3D geometry from uncalibrated images. In *Advances in Visual Computing*, pages II: 802–813.

[Karp, 1972a] Karp, R. M. (1972a). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, NY.

[Karp, 1972b] Karp, R. M. (1972b). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Symposium on geometry processing. In *Symposium on Geometry Processing*, pages 61–70.

[Kim and Muller, 2002] Kim, J. R. and Muller, J.-P. (2002). 3d reconstruction from very high resolution satellite stereo and its application to object identification.

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

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

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145.

[Konolige, 2010] Konolige, K. (2010). Sparse sparse bundle adjustment. In Labrosse, F., Zwiggelaar, R., Liu, Y., and Tiddeman, B., editors, *British Machine Vision Conference (BMVC)*, pages 1–11. British Machine Vision Association.

[Kruppa, 1913] Kruppa, E. (1913). Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung. *Other Journal*, pages 1939–1948.

[Ladicky et al., 2010] Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W. F., and Torr, P. H. S. (2010). Joint optimisation for object class segmentation and dense stereo reconstruction. In *British Machine Vision Conference (BMVC)*, pages 1–11. British Machine Vision Association.

[Leberl and Gruber, 2003] Leberl, F. and Gruber, M. (2003). Flying the new large format digital camera ultracam-d. In *Proceedings of the Photogrammetric Week, Stuttgart University*.

[Leberl et al., 2010] Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., and Wiechert, A. (2010). Point clouds: Lidar versus 3d vision. *Photogrammetric Engineering and Remote Sensing*.

[Lepetit et al., 2005] Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II: 775–781.

[Leutenegger et al., 2011] Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Li and Hartley, 2006] Li, H. D. and Hartley, R. I. (2006). Five-point motion estimation made easy. In *International Conference on Pattern Recognition (ICPR)*, pages I: 630–633.

[Li et al., 2008] Li, X., Wu, C., Zach, C., Lazebnik, S., and Frahm, J.-M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In *European Conference on Computer Vision (ECCV)*.

[Li et al., 2010] Li, Y., Snavely, N., and Huttenlocher, D. P. (2010). Location recognition using prioritized feature matching. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV)*, volume 6312, pages 791–804. Springer.

[Lloyd, 1982] Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, IT-28(2):129–137.

[Lourakis and Argyros, 2004] Lourakis, M. and Argyros, A. (2004). The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH.

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

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

[Ma et al., 2003] Ma, Y., Soatto, S., and KoseckÃ¡, J. (2003). *An Invitation to 3-D Vision*. Springer.

[Madsen et al., 2004] Madsen, K., Nielsen, H. B., and Tingleff, O. (2004). Methods for non-linear least squares problems (2nd ed.).

[Mahon et al., 2008] Mahon, I., Williams, S. B., Pizarro, O., and Johnson-Roberson, M. (2008). Efficient view-based SLAM using visual loop closures. *IEEE Transactions on Robotics*, 24(5):1002–1014.

[Mair et al., 2010] Mair, E., Hager, G. D., Burschka, D., Suppa, M., and Hirzinger, G. (2010). Adaptive and generic corner detection based on the accelerated segment test. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV)*, volume 6312, pages 183–196. Springer.

[Malis and Cipolla, 2002] Malis, E. and Cipolla, R. (2002). Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(9):1268–1272.

[Marr, 1982] Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, San Francisco.

[Martinec and Pajdla, 2006] Martinec, D. and Pajdla, T. (2006). 3d reconstruction by gluing pairwise euclidean reconstructions, or "how to achieve a good reconstruction from bad images". In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

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

[Mellinger et al., 2010] Mellinger, D., Michael, N., and Kumar, V. (2010). Trajectory generation and control for precise aggressive maneuvers with quadrotors. In *Proceedings of the International Symposium on Experimental Robotics*.

[Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60(1):63–86.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630.

[Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *Int. Journal of Computer Vision*, 65:43–72.

[Moravec, 1980] Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. In *Stanford University, Computer Science Departent*.

[Mouragnon et al., 2006] Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., and Dhome, M. (2006). Real time localization and 3D reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I: 363–370.

[Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In Ranchordas, A. and Araújo, H., editors, *VISAPP (1)*, pages 331–340. INSTICC Press.

[Najafi et al., 2006] Najafi, H., Genc, Y., and Navab, N. (2006). Fusion of 3d and appearance models for fast object detection and pose estimation. In *Asian Conference on Computer Vision (ACCV)*, pages 415–426.

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

[Newcombe et al., 2011] Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). Live dense reconstruction with a single moving camera. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE.

[Ni et al., 2007] Ni, K., Steedly, D., and Dellaert, F. (2007). Out-of-core bundle adjustment for large-scale 3D reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Nistér, 2000] Nistér, D. (2000). Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *European Conference on Computer Vision (ECCV)*, pages I: 649–663.

[Nistér, 2001] Nistér, D. (2001). *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, Royal Institute of Technology KTH, Stockholm, Sweden.

[Nistér, 2004] Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770.

[Nistér, 2005] Nistér, D. (2005). Preemptive RANSAC for live structure and motion estimation. *Mach. Vis. App.*

[Nister et al., 2007] Nister, D., Kahl, F., and Stewenius, H. (2007). Structure from motion with missing data is NP-hard. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–7.

[Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–659.

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

[Okutomi and Kanade, 1993] Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(4):353–363.

[Pajdla et al., 1997] Pajdla, T., Werner, T., and Hlaváč, V. (1997). Correcting radial lens distortion without knowledge of 3-D structure. Technical report, Center for Machine Perception, Czech Technical University.

[Peucker et al., 1978] Peucker, T. K., Fowler, R. J., Little, J. J., and Mark, D. M. (1978). The triangulated irregular network. *Amer Soc Photogrammetry Proc Digital Terrain Models Symposium*, 516:96–103.

[Pock et al., 2010] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2010). Global solutions of variational models with convex regularization. *SIAM J. Imaging Sciences*, 3(4):1122–1145.

[Pock et al., 2008] Pock, T., Schoenemann, T., Graber, G., Bischof, H., and Cremers, D. (2008). A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision (ECCV)*, Marseille, France.

[Pollefeys et al., 2004] Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *Int. Journal of Computer Vision*, 59(3):207–232.

[Pollefeys et al., 2002] Pollefeys, M., Verbiest, F., and Van Gool, L. (2002). Surviving dominant planes in uncalibrated structure and motion recovery. In *European Conference on Computer Vision (ECCV)*, pages 837–851.

[Pollefeys et al., 2008] Pollefeys et al., M. (2008). Detailed real-time urban 3d reconstruction from video. *Int. Journal of Computer Vision*, 78(2-3).

[Pritchett and Zisserman, 1998] Pritchett, P. and Zisserman, A. (1998). Wide baseline stereo matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 754–760.

[Raguram et al., 2008] Raguram, R., Frahm, J.-M., and Pollefeys, M. (2008). A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *European Conference on Computer Vision (ECCV)*.

[Raguram et al., 2009] Raguram, R., Frahm, J.-M., and Pollefeys, M. (2009). Exploiting uncertainty in random sample consensus. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2074–2081. IEEE.

[Raguram et al., 2011] Raguram, R., Wu, C., Frahm, J.-M., and Lazebnik, S. (2011). Modeling and recognition of landmark image collections using iconic scene graphs. *Int. Journal of Computer Vision*, 95(3):213–239.

[Reitmayr and Drummond, 2006] Reitmayr, G. and Drummond, T. (2006). Going out: robust model-based tracking for outdoor augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 109–118. IEEE.

[Reitmayr and Drummond, 2007] Reitmayr, G. and Drummond, T. (2007). Initialisation for visual tracking in urban environments. In *Proc. ISMAR 2007*, pages 161–160.

[Roberts et al., 2011] Roberts, R., Sinha, S. N., Szeliski, R., and Steedly, D. (2011). Structure from motion for scenes with large duplicate structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3137–3144. IEEE.

[Robertson and Cipolla, 2004] Robertson, D. and Cipolla, R. (2004). An image-based system for urban navigation. In *British Machine Vision Conference (BMVC)*.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. W. (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages I: 430–443.

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradskit, G. (2011). ORB: an efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Rudol et al., 2010] Rudol, P., Wzorek, M., and Doherty, P. (2010). Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems. In *ICRA*, pages 1913–1920. IEEE.

[Sattler et al., 2011] Sattler, T., Leibe, B., and Kobbelt, L. (2011). Fast image-based localization using direct 2d-to-3d matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Saxena et al., 2005] Saxena, A., Chung, S. H., and Ng, A. Y. (2005). Learning depth from single monocular images. *NIPS*, 18.

[Scaioni et al., 2009] Scaioni, M., Barazzetti, L., Brumana, R., Cuca, B., Fassi, F., and Prandi, F. (2009). RC-heli and structure and motion techniques for the 3-D reconstruction of a milan dome spire. In *3DARCH09*.

[Schall et al., 2009] Schall, G., Wagner, D., Reitmayr, G., Taichmann, E., Wieser, M., Schmalstieg, D., and Hofmann-Wellenhof, B. (2009). Global pose estimation using multi-sensor fusion for outdoor augmented reality. In Klinker, G., Saito, H., and Höllerer, T., editors, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 153–162. IEEE Computer Society.

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

[Schindler and Dellaert, 2010] Schindler, G. and Dellaert, F. (2010). Probabilistic temporal inference on reconstructed 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1417. IEEE.

[Schindler et al., 2007b] Schindler, G., Dellaert, F., and Kang, S. B. (2007b). Inferring temporal order of images from 3d structure. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[Se et al., 2002] Se, S., Lowe, D. G., and Little, J. J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *I. J. Robotic Res*, 21(8):735–760.

[Seitz et al., 2006] Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 519–526.

[Shannon, 1949] Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37:10–21.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle.

[Simon et al., 2007] Simon, I., Snavely, N., and Seitz, S. M. (2007). Scene summarization for online image collections. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8.

[Singhal, 2001] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43.

[Sinha et al., 2010] Sinha, S., Seitz, S., and Szeliski, R. (2010). A multi-stage linear approach to structure from motion. In *ECCV 2010 Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477.

[Skrypnyk and Lowe, 2004a] Skrypnyk, I. and Lowe, D. G. (2004a). Scene modelling, recognition and tracking with invariant image features. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 110–119. IEEE Computer Society.

[Skrypnyk and Lowe, 2004b] Skrypnyk, I. and Lowe, D. G. (2004b). Scene modelling, recognition and tracking with invariant image features. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 110–119.

[Snavely, 2008] Snavely, N. (2008). *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD thesis, University of Washington.

[Snavely et al., 2006] Snavely, N., Seitz, S., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. In *Proceedings of SIGGRAPH 2006*, pages 835–846.

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

[Snavely et al., 2008b] Snavely, N., Seitz, S. M., and Szeliski, R. S. (2008b). Skeletal graphs for efficient structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Steele and Egbert, 2006] Steele, K. L. and Egbert, P. K. (2006). Minimum spanning tree pose estimation. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 440–447.

[Strecha et al., 2010] Strecha, C., Pylvänäinen, T., and Fua, P. (2010). Dynamic and scalable large scale image reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 406–413. IEEE.

[Sun et al., 2003] Sun, J., Shum, H. Y., and Zheng, N. N. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(5):1226–1238.

[Takacs et al., 2008] Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.-C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., and Girod, B. (2008). Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Multimedia Information Retrieval*, pages 427–434.

[Teller and Séquin, 1991] Teller, S. J. and Séquin, C. H. (1991). Visibility preprocessing for interactive walktroughs. *Proceedings of SIGGRAPH'91*, 25(4):61–69.

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

[Tordoff and Murray, 2002] Tordoff, B. and Murray, D. W. (2002). Guided sampling and consensus for motion estimation. In *European Conference on Computer Vision (ECCV)*, pages 82–98.

[Torr, 2002] Torr, P. H. S. (2002). Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *Int. Journal of Computer Vision*, 50(1):35–61.

[Torr et al., 1998] Torr, P. H. S., Fitzgibbon, A. W., and Zisserman, A. (1998). Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *IEEE International Conference on Computer Vision (ICCV)*, pages 485–491.

[Torralba and Oliva, 2002] Torralba, A. and Oliva, A. (2002). Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(9):1226–1238.

[Triggs, 1998] Triggs, B. (1998). Autocalibration from planar scenes. In *European Conference on Computer Vision (ECCV)*.

[Triggs et al., 2000] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). Bundle adjustment – A modern synthesis. In Triggs, W., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag.

[Tsai, 1986] Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 364–374. IEEE.

[Tsai, 1987] Tsai, R. Y. (1987). A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344.

[Tuite et al., 2011] Tuite, K., Snavely, N., yu Hsiao, D., Tabing, N., and Popovic, Z. (2011). Photocity: training experts at large-scale image acquisition through a competitive game. In Tan, D. S., Amershi, S., Begole, B., Kellogg, W. A., and Tungare, M., editors, *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 1383–1392. ACM.

[Turcot and Lowe, 2009] Turcot, P. and Lowe, D. G. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop LAVD*.

[Vergauwen and Van Gool, 2006] Vergauwen, M. and Van Gool, L. (2006). Web-based 3D reconstruction service. *Mach. Vision Appl.*, 17(6):411–426.

[Wendel et al., 2011] Wendel, A., Irschara, A., and Bischof, H. (2011). Automatic alignment of 3d reconstructions using a digital surface mode. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Aerial Video Processing*.

[Wheeler et al., 1998] Wheeler, M., Sato, Y., and Ikeuchi, K. (1998). Consensus surfaces for modeling 3d objects from multiple range images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 917 – 924.

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

[Wu et al., 2011] Wu, C., Agarwal, S., Curless, B., and Seitz, S. M. (2011). Multicore bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064. IEEE.

[Wu et al., 2008] Wu, C., Clipp, B., Li, X., Frahm, J.-M., and Pollefeys, M. (2008). 3D Model Matching with Viewpoint Invariant Patches (VIPs). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Xiao et al., 2008] Xiao, J. X., Chen, J. N., Yeung, D. Y., and Quan, L. (2008). Structuring visual words in 3D for arbitrary-view object localization. In *European Conference on Computer Vision (ECCV)*.

[Yang and Pollefeys, 2003] Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 211–217.

[Zach et al., 2010] Zach, C., Klopschitz, M., and Pollefeys, M. (2010). Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1426–1433. IEEE.

[Zach et al., 2007] Zach, C., Pock, T., and Bischof, H. (2007). A globally optimal algorithm for robust TV-L^1 range image integration. In *IEEE International Conference on Computer Vision (ICCV)*.

[Zach and Pollefeys, 2010] Zach, C. and Pollefeys, M. (2010). Practical methods for convex multi-view reconstruction. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV)*, volume 6314 of *Lecture Notes in Computer Science*, pages 354–367. Springer.

[Zach et al., 2006a] Zach, C., Sormann, M., and Karner, K. (2006a). High-performance multi-view reconstruction. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 113–120.

[Zach et al., 2006b] Zach, C., Sormann, M., and Karner, K. (2006b). Scanline optimization for stereo on graphics hardware. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

[Zebedin, 2010] Zebedin, L. (2010). *Automatic Reconstruction of Urban Environments from Aerial Images*. PhD thesis, Graz University of Technology.

[Zechner and Granitzer, 2009] Zechner, M. and Granitzer, M. (2009). Accelerating K-means on the graphics processor via CUDA. In Boronat, F. and Dini, C., editors, *First International Conference on Intensive Applications and Services, INTENSIVE 2009, Valencia, Spain, 20-25 April 2009*, pages 7–15. IEEE.

[Zhang and Kosecka, 2006] Zhang, W. and Kosecka, J. (2006). Image based localization in urban environments. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 33–40.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334.

[Zhang and Hanson, 1996] Zhang, Z. and Hanson, A. (1996). 3d reconstruction based on homography mapping. In *Proc. ARPA96*, pages 1007–1012.

[Zhu et al., 2008] Zhu, Z. W., Oskiper, T., Samarasekera, S., Kumar, R., and Sawhney, H. S. (2008). Real-time global localization with a pre-built visual landmark database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.