



PhD Thesis

**Robust Reconstruction and Efficient
Localization for Mobile Augmented Reality**

Manfred Klopschitz

Graz University of Technology
Institute for Computer Graphics and Vision

Thesis supervisors
Prof. Dr. Gerhard Reitmayr
Prof. Dr. Marc Pollefeys

Graz, April 2012

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 1.1 | Problem Statement | 2 |
| 1.2 | Overview and Contribution | 5 |
| 2 | Methods | 9 |
| 2.1 | Introduction | 9 |
| 2.2 | Camera Models | 12 |
| 2.2.1 | Pinhole Camera Model | 12 |
| 2.2.2 | Geometric Camera Calibration | 15 |
| 2.3 | 6DOF Localization | 15 |
| 2.3.1 | 6DOF Localization with a Calibrated Camera | 17 |
| 2.3.2 | 6DOF Localization for Panoramic Images | 18 |
| 2.4 | Structure From Motion | 20 |
| 2.4.1 | Multiple View Geometry | 20 |
| 2.4.2 | Bundle Adjustment | 21 |
| 2.4.3 | Non-Linear Least Squares | 21 |
| 2.4.4 | SfM Initialization | 23 |
| 2.4.5 | Essential Matrix | 25 |
| 2.4.6 | SfM for Multiple Views | 27 |
| 2.5 | Point Correspondences | 28 |
| 2.5.1 | Interest Points | 28 |
| 2.5.2 | Feature Descriptors | 30 |
| 2.5.3 | Feature Matching | 30 |
| 2.5.4 | Large Scale Feature Matching | 30 |
| 2.6 | Robust Estimation | 34 |
| 2.6.1 | RANSAC Improvements | 35 |

| | | |
|----------|---|-----------|
| 2.6.2 | Model Computation | 35 |
| 2.6.3 | Model Verification | 35 |
| I | Robust Scene Reconstruction | 37 |
| 3 | Robust Incremental Structure from Motion | 39 |
| 3.1 | Introduction | 39 |
| 3.2 | Structure and Motion Computation | 43 |
| 3.2.1 | Epipolar Graph $G_{\mathcal{E}}$ | 43 |
| 3.2.2 | Trifocal Graph $G_{\mathcal{T}}$ | 44 |
| 3.2.3 | Reconstruction | 46 |
| 3.3 | Results | 49 |
| 3.4 | Conclusion and Discussion | 55 |
| 4 | Feature Matching for Sequential Data | 57 |
| 4.1 | Introduction | 57 |
| 4.2 | Image-based Sequence Similarity | 58 |
| 4.2.1 | Image Matching | 59 |
| 4.2.2 | Visual Similarity Matrix | 59 |
| 4.3 | Sequence Alignment | 59 |
| 4.3.1 | Scanline Optimization Problem Formulation | 61 |
| 4.3.2 | Efficient Minimization | 62 |
| 4.3.3 | Matching Multiple Sequences | 63 |
| 4.4 | Results | 64 |
| 4.5 | Summary and Conclusions | 64 |
| 5 | Disambiguating Visual Relations Using Loop Constraints | 67 |
| 5.1 | Introduction | 67 |
| 5.2 | Related Work | 70 |
| 5.3 | Inference of False Visual Relations | 71 |
| 5.4 | Application: Homography Matching | 75 |
| 5.5 | Application: Screening the Epipolar Graph | 76 |
| 5.6 | Application: Structure and Motion | 78 |
| 5.7 | Comparison Between BP and BnB Inference | 80 |
| 5.8 | Discussion and Future Work | 80 |

| | | |
|-----------|---|------------|
| II | Efficient Localization | 85 |
| 6 | Visibility Constrained Localization | 89 |
| 6.1 | Overview | 91 |
| 6.2 | Related Work | 91 |
| 6.2.1 | Localization for Mobile Phones | 92 |
| 6.2.2 | Potentially Visible Sets | 92 |
| 6.3 | Offline Data Acquisition | 93 |
| 6.3.1 | Global Registration | 94 |
| 6.3.2 | Potentially Visible Sets | 95 |
| 6.4 | Localization | 95 |
| 6.4.1 | Run-time Memory Management | 97 |
| 6.4.2 | PVS Selection | 97 |
| 6.4.3 | Localization | 98 |
| 6.4.4 | Detection vs. Tracking | 99 |
| 6.5 | Experiments | 99 |
| 6.5.1 | Test Data Acquisition | 99 |
| 6.5.2 | Memory Consumption | 100 |
| 6.5.3 | Matching Strategies | 101 |
| 6.5.4 | Image Resolution | 101 |
| 6.5.5 | Full System Mobile Phone Evaluation | 103 |
| 6.6 | Conclusion and Outlook | 104 |
| 7 | Robust Localization from Panoramic Images | 109 |
| 7.1 | Panorama Generation | 109 |
| 7.2 | Reconstruction and Global Registration | 111 |
| 7.2.1 | Visibility Partitioning | 112 |
| 7.3 | Localization from Panoramic Images | 115 |
| 7.4 | Experiments | 115 |
| 7.4.1 | Localization Database and Panoramic Images | 115 |
| 7.4.2 | Aperture Dependent Localization Performance | 117 |
| 7.4.3 | Pose Accuracy | 120 |
| 7.4.4 | Runtime Estimation | 121 |
| 7.4.5 | Panoramas captured under Realistic Conditions | 124 |
| 7.4.6 | Live Augmentation | 125 |
| 7.4.7 | Discussion | 126 |
| 7.5 | Conclusion | 126 |

| | |
|-----------------------|------------|
| 8 Conclusion | 129 |
| 8.1 Outlook | 129 |

Abstract

This thesis introduces robust reconstruction and efficient localization methods for mobile Augmented Reality (AR). Robust Structure from Motion (SfM) methods are necessary to create sparse reconstructions of specific areas where localization and therefore AR should be possible. The resulting reconstructions serve as large, general tracking targets. Efficient localization methods are required to solve the localization problem on computationally limited mobile devices. Apart from specific algorithmic improvements, this thesis demonstrates that image-based localization and SfM are highly interconnected problems. The 3D data that is needed for pose computation is exactly what SfM methods compute. Furthermore, image-based pose estimation algorithms can minimize a reprojection error and not an object-space error. This reprojection error is exactly what matters for superimposing information, which is the topic of AR. The user notices localization deviations as pixel offsets, the absolute position error is usually irrelevant.

We present a robust incremental hierarchical SfM approach that allows to create large data sets of specific areas for localization applications. Image matches are the input of basically all SfM methods. Because image matching itself is a hard task we increase the matching robustness further by using the additional sequential information of image sequences when possible and by screening the epipolar graph with novel cycle constraints in graphs of pair-wise visual relations.

Highly efficient localization methods are presented that use visibility information to partition the SfM point cloud results. This addresses two limitations of current mobile devices for image-based localization: (i) The amount of main memory that is necessary for search data structures and (ii) computational requirements. We show that localization quality is highly correlated with the field of view. Since panoramic images can be created online on mobile devices, the user can contribute to the localization performance by increasing the field of view. The robustness and performance of the presented reconstruction and localization methods is finally demonstrated and evaluated in a large-scale outdoor localization experiment that uses a combination of all presented techniques.

Chapter 1

Introduction

AR superimposes registered 3D graphics on the users view of the real world, allowing the user to perceive overlaid information that is spatially registered to the environment, see Figure 1.1. In this context, tracking provides a user camera localization to correctly register digital information, e.g. navigational hints, to the user. Computer Vision-based localization techniques offer great advantages over other localization methods based on infrared, WLAN/Wi-Fi or ultra-wide-band. Image-based measurements allow very high precision in the pose estimation and self-contained operation without complex and expensive infrastructure. In computer vision, the problem of location recognition has been addressed in the past by a variety of approaches. 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, planar surfaces or 3D models. Assuming a static scene, geometric verification can be used to determine the actual pose of the camera with respect to the exemplary database.

There are multiple reasons why image-based localization methods is appropriate for solving the tracking problem: Current off-the-shelf mobile devices offer integrated video cameras and therefore a potential tracking sensor. These devices can be used as a complete user AR device, equipped with a display, computational power and at least one camera to observe the world. Mobile phones are a particular successful example of this class of devices. Image-based pose estimation algorithms minimize the reprojection error and not an object space error. Furthermore, this reprojection error is exactly what matters for superimposing information, the user notices registration deviations as pixel offsets, the absolute error is usually irrelevant. To compute a 3D position and orientation of an image, corresponding points to 3D structure is a direct way to obtain a position. SfM methods are concerned with obtaining 3D structure from image data, so this is perfectly suitable for re-detection in tracking images. Figure 1.2 shows a reconstruction of a well



Figure 1.1: *Augmented Reality User Interface for a Map.* The user is equipped with a mobile device that renders 3D structures and information directly on top of the orthographic photo. The left image shows the screen of the mobile device, the user interaction is visible in the right image.

known scene, with relative camera positions and the resulting 3D structure. The topic of this thesis is to investigate the properties of this data as tracking infrastructure.

1.1 Problem Statement

Using 3D reconstructions as image-based tracking infrastructure poses two major challenges: Firstly, 3D point data has to be obtained. Secondly, efficient solutions for the correspondence problem from query images into this 3D database is necessary.

While it is clear that 6DOF localization is possible from 2D-3D point correspondences, it is not obvious how to obtain the 3D point data for real world situations. For small scale applications it would be possible to measure points using a total station, but this is of course time consuming and error-prone manual work. The measured 3D points have to be identified in the query images again to obtain the 2D-3D correspondences. To automate both steps, the measurement of the world and the identification of possible imaged points, a suitable work flow is necessary. We use Structure from Motion (SfM) techniques to create sparse 3D reconstructions from image data to survey the world points. SfM methods compute camera positions and orientations from image to image point correspondences. 3D points are computed from these correspondences implicitly. There are multiple reasons why SfM is interesting for creating localization data:

- **Automation:** With SfM it is possible to create large quantities of 3D point data automatically from sets of input images.
- **Simple Hardware Setup:** Only commodity digital cameras are necessary to cap-

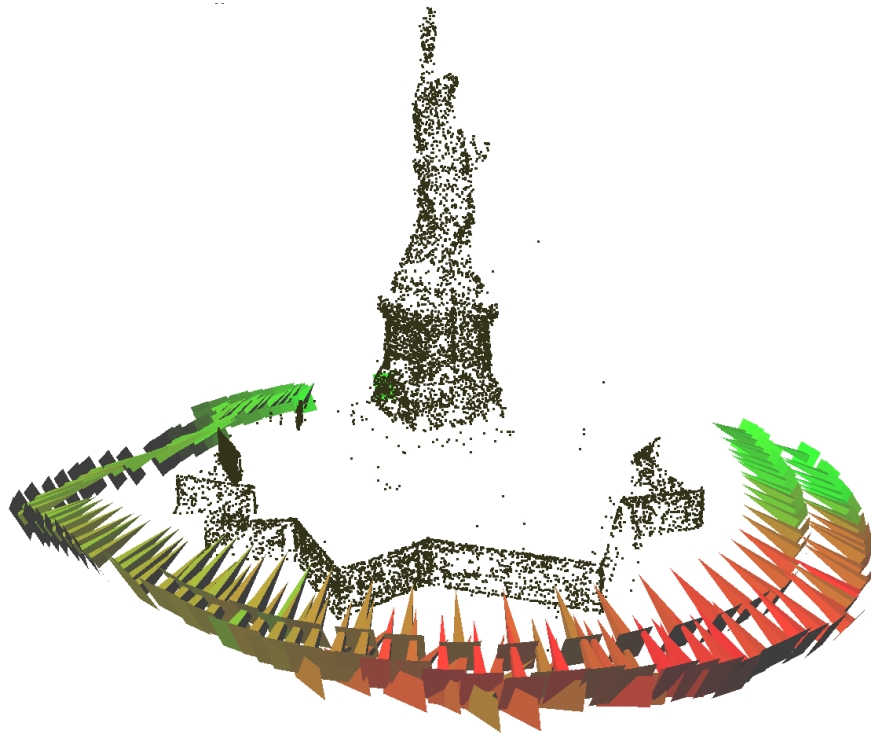


Figure 1.2: A Typical Structure from Motion Result: From a set of input images the relative camera positions (indicated by pyramids) and 3D feature point structure is computed. Each 3D point is visible in multiple images and 2D feature points with feature descriptors are associated to each 3D point.

ture the input data. No special hardware setups have to be built and commercial cameras are easy and convenient to use.

- **Suitable Modality:** SfM input data is basically the same as image-based localization input data, namely camera images. This makes it possible to identify 2D-3D correspondences automatically by applying the same feature point detectors to the SfM data sets and the localization query images.

The data acquisition stage has additional requirements for our use case than in typical SfM scenarios. It is not enough to obtain visually pleasing results of some famous place that is already documented by thousands of images. For AR tracking we usually have a target work space in mind that has to be covered. Structure from motion is a hard problem by itself and our additional requirements make 3D reconstruction even more difficult. This is therefore addressed in the first part of the thesis.

For the tracking part, the focus in this work is on efficiency and robustness aspects. Figure 1.3 shows a 3D reconstruction with millions of 3D points. Efficient search algo-

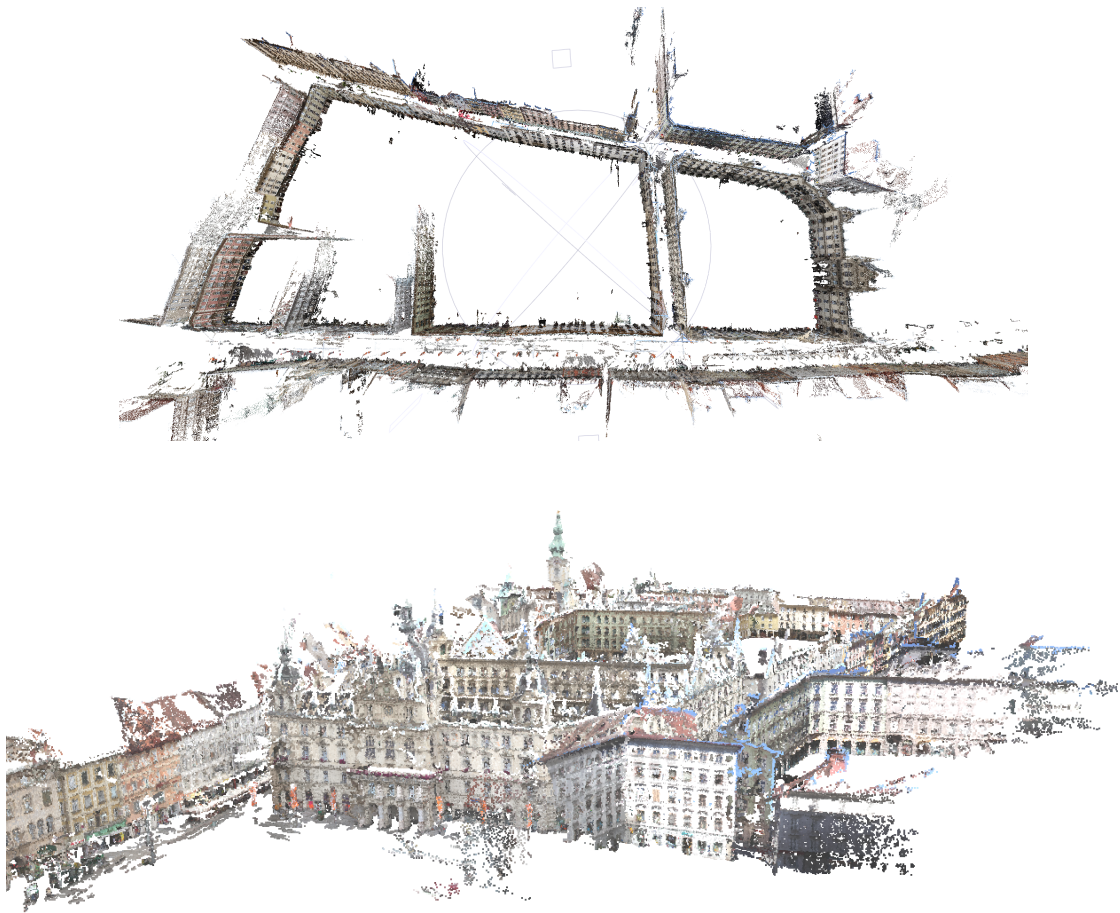


Figure 1.3: Large 3D point cloud reconstruction result created with methods presented in this thesis and used for extensive outdoor localization experiments. The data set size illustrates that correspondence search on this scale is computationally demanding.

gorithms are clearly important in this case. Furthermore, mobile devices are perfect AR end user devices. These devices have limited memory and computational power compared to desktop computers or computing clusters. We try to address these limitations for example by limiting the search space using visibility considerations. The camera quality of these mobile devices is another limitation, image-based localization benefits from camera quality and a large field of view. We are investigating user guided image capturing methods to mitigate these limitations. Furthermore, 3D reconstructions are prone to changes in the observed environment. These effects can also be mitigated by an increasing field of view.

1.2 Overview and Contribution

Image-based 6DOF localization and structure from motion are highly interconnected problems. The 3D data that is needed for an absolute pose computation is exactly what structure from motion methods compute. To create 3D data sets that are interesting for localization experiments and cover predefined locations, structure from motion methods had to be improved. Chapter 3 presents a robust incremental reconstruction approach and was used to create all the 3D data for testing the localization algorithms in this thesis. This reconstruction approach is further improved in Chapter 4 and 5 for special scenarios. Chapter 6 uses this data for indoor localization and introduces point based visibility concepts to improve image-based matching computation times. Chapter 7 builds upon the aperture angle results of the indoor localization experiments and extends the localization to panoramic images. The influence of the field of view is extensively evaluated on a large-scale (in space and time) outdoor data set.

- We show that image-based 6DOF localization and structure from motion are highly interconnected problems. The 3D data that is needed for an absolute pose computation is exactly what structure from motion methods compute. Efficient and robust feature detection, description and matching techniques are also very similar.
- A robust incremental hierarchical structure from motion approach is presented that allows to create suitable data sets for further localization experiments. The method is incremental for scalability reasons and is able to create reconstructions in difficult low-redundancy scenarios.
- Robust sequential image matching techniques are introduced that are suitable for our reconstruction problems. This makes it possible to detect robust overlaps of individual 3D reconstructions in ambiguous environments.
- A method to filter the epipolar image matching graph in difficult ambiguous environments is presented. This makes structure from motion even more robust. The proposed method can be generalized to visual relations where cycles in a graph can be expressed as an invertible transformation.
- State of the art computer vision techniques are combined with visibility reasoning to make 6DOF localization more efficient and robust. This is demonstrated in an indoor scenario. This is a general principle that can be used to speed up different matching techniques.
- Image-based localization is extended to panoramic images. Panoramic images can be created online on mobile devices, this gives us the opportunity to let the user

contribute to the localization performance by increasing the field of view. The recognition rate and localization accuracy are improved. The effects of these field of view extensions are evaluated on a large-scale data set. This is again a general concept that does not depend directly on a particular feature detector, descriptor or matching technique.

This thesis is based on work documented in a set of thesis-related publications. Some other aspects of the topic, for example the connection to photogrammetry and absolute reconstruction accuracy, are documented in topic-related publications.

Thesis-Related

[1] *Generalized Detection and Merging of Loop Closures for Video Sequences* Manfred Klopschitz, Christopher Zach, Arnold Irschara, and Dieter Schmalstieg. 3DPVT 2008.

[2] *Robust Incremental Structure from Motion* Manfred Klopschitz, Arnold Irschara, Gerhard Reitmayr, Dieter Schmalstieg 3DPVT 2010.

[3] *Towards Wide Area Localization on Mobile Phones* Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara and Dieter Schmalstieg . ISMAR 2009.

[4] *Disambiguating Visual Relations Using Loop Constraints*. Christopher Zach, Manfred Klopschitz, Marc Pollefeys, CVPR 2010

[5] *Real-Time Self-Localization from Panoramic Images on Mobile Devices* Clemens Arth, Manfred Klopschitz, Gerhard Reitmayr , Dieter Schmalstieg , ISMAR 2011

Topic-Related

[1] *Visual Tracking for Augmented Reality* Manfred Klopschitz, Gerhard Schall, Dieter Schmalstieg, Gerhard Reitmayr IPIN 2010 International Conference on Indoor Positioning and Indoor Navigation, ETH Zurich, Switzerland

[2] *Automatic Reconstruction of Wide-Area Fiducial Marker Models* Manfred Klopschitz, and Dieter Schmalstieg. ISMAR 2007.

[3] *Large Scale, Dense City Reconstruction from User-Contributed Photos*. Arnold Irschara,

Christopher Zach, Manfred Klopschitz, and Horst Bischof. *Journal Computer Vision and Image Understanding*, 2011.

[4] *Towards Fully Automatic Photogrammetric Reconstruction Using Digital Images Taken From UAVs* Arnold Irschara , Viktor Kaufmann, Manfred Klopschitz , Horst Bischof , Leberl Franz. ISPRS 2010

Chapter 2

Methods

This chapter gives an overview of localization methods for AR, a brief introduction to image-based localization and 3D reconstruction fundamentals. Image-based localization and structure from motion have a lot of basic techniques in common.

For both problems, minimal point cases are important. Solutions for the minimal cases are usually perfectly satisfied geometry problems that contain no measurement error. Actual image measurements are of course not perfect and robust statistical methods are used to integrate all available measurements.

Furthermore, solving the correspondence problem is essential for localization and reconstruction. For localization, points in the image have to be matched to 3D structure points. Reconstruction uses image to image (2D-2D) point correspondences to infer structure and camera positions. Similar techniques to extract suitable feature points, create a robust encoding of them and match to large databases are necessary for both problems.

2.1 Introduction

Localization in Augmented Reality Among the first dedicated wearable location systems was the Active Badge system [Want et al., 1992], which consisted of infrared (IR) badges sending location information signals to a server. Its successor, the Bat system [Ardlesee et al., 2001], used ultrasonic location estimation to provide more accurate position data. PlaceLab [Otsason et al., 2005] is a system that relies on signal strength of existing infrastructure, such as GSM, Bluetooth and WiFi, for indoor and outdoor location tracking. Accuracy strongly depends on the number of senders in the environment and has been reported in the range of 3-6 meters for indoor usage.

Markers have a strong tradition in Augmented Reality due to their robustness and low computational requirements. Hence, marker based approaches have also been applied for indoor localization. Kalkusch *et al.* [Kalkusch et al., 2002] employed localization

by tracking 2D barcodes (fiducial markers) that were installed in the environment. The IS-1200 tracker [Naimark and Foxlin, 2002] is a commercial variant of this approach, combining tracking of circular fiducials with an inertial sensor for increased robustness. Although there is a large body of publications for indoor and outdoor tracking for Augmented Reality, little work has gone into localization. Reitmayr proposed an accurate localization technique [Reitmayr and Drummond, 2007] using GPS to initialize tracking. The GPS accuracy makes it necessary to test multiple locations around the GPS position. A Gaussian process is used to model the GPS error and make re-initialization more efficient by reducing this search space around the GPS position.

Markerless registration for Augmented Reality on mobile phones has only recently become possible due to increased processing capabilities of modern smart phones. First approaches were based on optical flow based methods such as TinyMotion [Wang et al., 2006]. Our own previous work [Wagner et al., 2008] marks the first real-time 6DOF pose estimation from natural features on a mobile phone. However, none of these works addresses wide area tracking.

Vision based Localization 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, 2006a]. The most successful methods rely on bag of words based matching techniques of 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., 2007b], planar surfaces [Robertson and Cipolla, 2004] or 3D models [Irschara et al., 2009, 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., 2006] that act as descriptors of local image patches. These features have been found to be highly distinctive and repeatable in performance evaluation [Mikolajczyk and Schmid, 2003].

Some recent work formulates the city-scale localization as an image retrieval problem: assuming a database created from GPS-tagged images, find the closest matching images to the current view and return the GPS information or a quantity derived from it; for example, consider the work by Zhang and Kosecka [Zhang and Kosecka, 2006b]. A similar approach operating in near-real time is described by Schindler *et al.* [Schindler et al., 2007b]. A vocabulary tree concept and inverted file scoring as described in [Nister and Stewenius, 2006] is used to allow sub-linear search of large descriptor databases requiring low storage space. In contrast, Lepetit *et al.* [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 run time. The paper by Zhu *et al.* [Zhu et al., 2008] computes more accurate 2D coordinates in real time, but relies on a dual stereo camera setup, while the other systems use a conventional single camera.

In contrast to these approaches, which perform essentially 2D localization, AR requires full 6DOF. Outdoor self-localization with 6DOF was shown by Reitmayr and Drummond [Reitmayr and Drummond, 2006], but this work relies on a textured polygonal model of the environment and does not necessarily scale to large environments. Likewise, the seminal mapping and tracking work by Klein and Murray [Klein and Murray, 2007] is intended for small workspaces. In contrast, Irschara *et al.* presented a 6DOF localization method for large environments using vocabulary trees and a sparse point-cloud reconstruction [Irschara et al., 2009]. By inserting synthetic views during database creation, visibility of features from given viewpoints is inferred, which is later used to compress the database size and to improve the localization results. Real-time performance is achieved through the use of a GPU in order to deal with the high computational cost of the method. Li *et al.* [Li et al., 2010] improved accuracy over this work, but they do not report real-time frame rates.

Existing 6DOF localization systems have in common that they are computationally intensive and not directly suitable for mobile devices. Recent work has therefore examined how localization can be enabled using limited computational and storage resources. Takacs *et al.* [Takacs et al., 2008] perform keypoint detection and matching directly on the mobile phone. Features are clustered in a regular 2D grid and pre-fetched by proximity. Each grid element contains a set of clustered meta-features that represent the most repeatable and stable features of the scene. While this technique operates in real time, it does not provide full 6DOF. Klein's and Murray's [Klein and Murray, 2009] study showed how parallel mapping and tracking with 6DOF can be computed on a mobile device, they accepted a limitation to rather small workspaces as a trade-off, however.

Structure From Motion To make full 6DOF tracking possible without making assumptions (apart from the scene being static) about the structure of the environment, 3D points [Haralick et al., 1994] or camera centers with relative orientations [Brand et al., 2004a] relative to a query image have to be known. With more assumptions, like the presence of a dominant plane [Batz et al., 2010], the degrees of freedom of the unknown pose can be reduced. Some disadvantages are that it is not clear when the planarity assumption is not valid anymore, that matching gets more difficult with the number of planes and a geometric model of the environment has to be available or estimated. This thesis is concerned only with the general case, where 3D structure is needed and SfM is the general method used to obtain this data.

Recent literature in SfM comprises a number of approaches addressing the problem of

reconstructing a scene from unordered collections of images like [Snavely et al., 2006, Li et al., 2008, Martinec and Pajdla, 2007]. Snavely et al. describe in [Snavely et al., 2006] a system that is able to reconstruct a scene from a very diverse set of images, like image collections gathered from the web. A limiting factor regarding the scalability of this approach is mainly pair-wise image matching and large scale bundle adjustment. In [Snavely et al., 2008b] the latter problem is addressed by computing a small but representative *skeletal* set of a scene. Targeting at efficiency, Ni et al. propose in [Kai Ni Steedly, 2007] an out-of-core bundle adjustment capable of tackling larger reconstructions. Closely related is the approach described in [Li et al., 2008] where object recognition techniques are utilized to compute a small subset of *iconic images* that represent the important aspects from a scene. Again the algorithm is designed to work on large-scale image collections gathered from the Internet. Common for all the approaches is to rely on some calibration information, often directly derived from EXIF information.

In our intended use case it is not enough to obtain some models for a set of input images. We usually want to cover a specific place or area with as few SfM input images as possible. Current SfM methods have reliability problems with this kind of data. We are showing methods to improve on state of the art in this regard and these methods enable us to perform large scale localization experiments.

2.2 Camera Models

In this work we focus on central projection camera models. This is of course not the only possible camera model, for example linear pushbroom cameras [Gupta and Hartley, 1997] are used in satellite imaging.

2.2.1 Pinhole Camera Model

The pinhole camera model is a simple approximation of the imaging process in a real camera. A point in space is mapped onto the image plane where the line joining the projection center and the point in space intersect the image plane. Figure 2.1 shows the geometry of this model.

The mapping of a point in space $\mathbf{X} = (x, y, z)^T$ onto the image plane is given by:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} f \frac{x}{z} \\ f \frac{y}{z} \end{pmatrix}. \quad (2.1)$$

Here f models the focal length of the camera, the plane $z = f$ is called image plane, the point C is called camera center, the line that is perpendicular to the image plane and

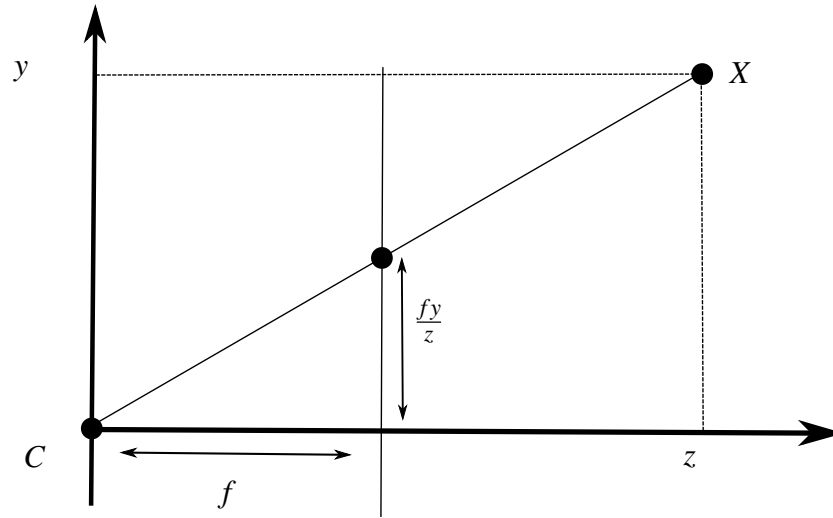


Figure 2.1: The basic pinhole camera model.

intersects C is called principal axis and the point where the principal axis and the image plane intersect is called the principal point.

2.2.1.1 Pinhole Camera Model in Homogeneous Coordinates

The non-linear mapping of Equation 2.1 can be written as a linear mapping by using homogeneous coordinates

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fx \\ fy \\ z \end{pmatrix}, \quad (2.2)$$

this can be formulated as a matrix equation:

$$\begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (2.3)$$

By representing the world point as homogeneous 4-vector $\mathbf{X} = (x, y, z, 1)^T$ and the mapped image point as homogeneous 3-vector \mathbf{x} and the 3×4 matrix in Equation 2.6 as P , Equation 2.6 can be written as

$$\mathbf{x} = P\mathbf{X}. \quad (2.4)$$

Allowing the principal point to move to $\mathbf{p} = (p_x, p_y)^T$ in the image plane an offset has to be added

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} f \frac{x}{z} + p_x \\ f \frac{y}{z} + p_y \end{pmatrix}. \quad (2.5)$$

Writing this in homogeneous coordinates as a matrix equation gives

$$\begin{pmatrix} fx + zp_x \\ fy + zp_y \\ z \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ & & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (2.6)$$

Defining the camera calibration matrix K as

$$K = \begin{bmatrix} f & p_x \\ f & p_y \\ & & 1 \end{bmatrix}, \quad (2.7)$$

the central projection model is now

$$\mathbf{x} = P\mathbf{X} = K[I|0]\mathbf{X}. \quad (2.8)$$

Until now the camera center is located at the origin, this is called camera coordinate frame. A general position of the camera in space can be modeled by rotating and translating the coordinate system (called world coordinate frame) into the camera coordinate frame. The transformation of a point $\tilde{\mathbf{X}}$ in the world coordinate frame to the point $\tilde{\mathbf{X}}_{cam}$ into the camera coordinate frame can be written as $\tilde{\mathbf{X}}_{cam} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$. R is a 3×3 rotation matrix and $\tilde{\mathbf{C}}$ the camera center in world coordinates. In homogeneous coordinates this is

$$\mathbf{X}_{cam} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}. \quad (2.9)$$

Combining this transformation with the model of the pinhole camera gives

$$\mathbf{x} = KR[I|-\tilde{\mathbf{C}}]\mathbf{X}. \quad (2.10)$$

R and $\tilde{\mathbf{C}}$ model the location of the camera in space and are called extrinsic camera parameters, the parameters of the matrix K depend only on internal camera properties and are called intrinsic parameters.

In real cameras the scales of both image axis do not have to be equal and in some cases a skew parameter of the image plane is needed. This can be modeled by extending the camera calibration matrix K

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}, \quad (2.11)$$

where $\alpha_x = fm_x$, $\alpha_y = fm_y$, $x_0 = m_x p_x$ and $y_0 = m_y p_y$, the factors m_x, m_y allow the transformation of world coordinates to pixel coordinates and s is the skew parameter.

2.2.2 Geometric Camera Calibration

The pinhole camera model is of course only an approximation of the imaging process in real cameras. Most cameras use lenses and they have a number of aberrations. This is even more complex in the case of automated zoom lenses, as [Willson, 1994] shows. The aberration errors are studied in detail in [Willson and Shafer, 1994]. This paper also defines many possible image centers. The geometric camera calibration process estimates the mapping between points in world coordinates and their corresponding image locations. The 3D points are usually assumed to have a known position in a fixed world coordinate system.

In the following example, the used camera model contains the extrinsic parameters, skew, focal length and the principal point for the intrinsic parameters and decomposes the lens distortion in the commonly used radial and tangential components. Many different distortion models are possible, the example follows [Heikkilä and Silven, 1997].

The known distorted 2D point \mathbf{x}_d and the corrected point \mathbf{x}_c are approximated by:

$$\mathbf{x}_c = \mathbf{x}_d + \mathcal{F}_D(\mathbf{x}_d, \boldsymbol{\delta}) \quad (2.12)$$

$$\mathcal{F}_D(\mathbf{x}_d, \boldsymbol{\delta}) = \begin{bmatrix} \bar{x}_{1d}(k_1 r_d^2 + k_2 r_d^4) + 2p_1 \bar{x}_{1d} \bar{x}_{2d} + p_2 (r_d^2 + 2\bar{x}_{1d}^2) \\ \bar{x}_{2d}(k_1 r_d^2 + k_2 r_d^4) + p_1 (r_d^2 + 2\bar{x}_{2d}^2) + 2p_2 \bar{x}_{1d} \bar{x}_{2d} \end{bmatrix} \quad (2.13)$$

with $\mathbf{x}_d = (x_{1d}, x_{2d})^T$, $\bar{x}_{1d} = x_{1d} - u_0$, $\bar{x}_{2d} = x_{2d} - v_0$, u_0 and v_0 are the coordinates of the principal point, $r_d = \sqrt{\bar{x}_{1d}^2 + \bar{x}_{2d}^2}$ and $\boldsymbol{\delta} = (k_1, k_2, p_1, p_2)^T$. k_1, k_2 are the radial distortion coefficients and p_1, p_2 the tangential distortion coefficients.

2.3 6DOF Localization

For all 2D-3D correspondences the following equation must hold:

$$\mathbf{x}_i = P\mathbf{X}_i \quad (2.14)$$

The problem of computing P is similar to the computation of a 2D homography. The Direct Linear Transformation (DLT) algorithm is used to solve for P . Because the points are represented by homogeneous coordinates, the vectors \mathbf{x}_i and $P\mathbf{X}_i$ representing 2D points have the same direction, but their lengths may differ by a non-zero scale factor. One way to express this is by using a cross product. Because the cross product of two vectors with the same direction is the null-vector, Equation 2.14 may be written as:

$$\mathbf{x}_i \times P\mathbf{X}_i = \mathbf{0} \quad (2.15)$$

If the 4-vector \mathbf{p}^{jT} is the j -th row of P , the product $P\mathbf{X}_i$ may be written as:

$$P\mathbf{X}_i = \begin{pmatrix} \mathbf{p}^{1T} \mathbf{X}_i \\ \mathbf{p}^{2T} \mathbf{X}_i \\ \mathbf{p}^{3T} \mathbf{X}_i \end{pmatrix} \quad (2.16)$$

If \mathbf{x}_i is written as $(x_i, y_i, w_i)^T$ the cross product is:

$$\mathbf{x}_i \times P\mathbf{X}_i = \begin{pmatrix} y_i \mathbf{p}^{3T} \mathbf{X}_i - w_i \mathbf{p}^{2T} \mathbf{X}_i \\ w_i \mathbf{p}^{1T} \mathbf{X}_i - x_i \mathbf{p}^{3T} \mathbf{X}_i \\ x_i \mathbf{p}^{2T} \mathbf{X}_i - y_i \mathbf{p}^{1T} \mathbf{X}_i \end{pmatrix} \quad (2.17)$$

Rewriting this equation as a linear homogeneous system gives:

$$\begin{bmatrix} \mathbf{0}^T & -w_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & \mathbf{0}^T & -x_i \mathbf{X}_i^T \\ -y_i \mathbf{X}_i^T & x_i \mathbf{X}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0} \quad (2.18)$$

This linear homogeneous system has the form $A_i \mathbf{p} = \mathbf{0}$. A_i is a 3×12 matrix and \mathbf{p} is a 12-vector composed of the three rows of P :

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{bmatrix}, P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \quad (2.19)$$

The three equations of 2.18 are linearly dependent. Usually the third equation is omitted:

$$\begin{bmatrix} \mathbf{0}^T & -w_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & \mathbf{0}^T & -x_i \mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0} \quad (2.20)$$

Stacking up Equation 2.20 for n point correspondences leads to a $2n \times 12$ Matrix A . P is computed by solving $A\mathbf{p} = \mathbf{0}$ for \mathbf{p} .

Solution Since P has 11 degrees of freedom at least 11 equations or rows in the matrix A are necessary to solve for $A\mathbf{p} = \mathbf{0}$ for \mathbf{p} . This means five point correspondences and the x- or y-coordinate of the sixth correspondence are necessary. In the case of an over-determined solution ($n \geq 6$) P may be estimated by minimizing an algebraic or geometric error.

Algebraic minimization The algebraic error of $A\mathbf{p} = \mathbf{0}$ is $\|A\mathbf{p}\|$, because $\mathbf{p} = \mathbf{0}$ is the trivial solution, another constraint on \mathbf{p} is necessary. The constraint usually used is $\|\mathbf{p}\| = 1$, so the over-determined solution may be easily obtained by a singular value decomposition (SVD) of A . Real implementations of this algorithm should normalize the 2D and 3D points (pre-conditioning A) to obtain greater numerical stability. Furthermore this linear solution may be used as a starting point for an iterative geometric cost function minimization.

2.3.1 6DOF Localization with a Calibrated Camera

Full 6DOF localization of an image is possible from three 2D-3D correspondences if the focal length is known. The geometry of this problem is illustrated in 2.2, the problem can be traced back to 1841 by german mathematician J.A. Grunert, a more recent treatment can be found in [Haralick et al., 1994]. Despite being a well studied problem, interesting improvements to solve the problem numerically are still published [Kneip et al., 2011].

A calibrated camera measures angles between two image points, for the absolute pose problem, three such angles θ_{ij} are observed between three image points. For three observed 3D points X_i , the pairwise 3D point distances l_{ij} can be computed. Furthermore the angles between pairs of image measurements θ_{ij} are known from the corresponding image measurements m_i . The unknowns are the distances x_i between the center of projection C and the 3D point X_i :

$$\begin{aligned} l_{ij} &= \|X_i - X_j\| \\ \theta_{ij} &= \angle(v_i, v_j) \\ x_i &= \|C - X_i\|. \end{aligned}$$

Using the law of cosines each of the three point pairs gives one equation:

$$l_{ij}^2 = x_i^2 + x_j^2 - 2x_i x_j \cos \theta_{ij}$$

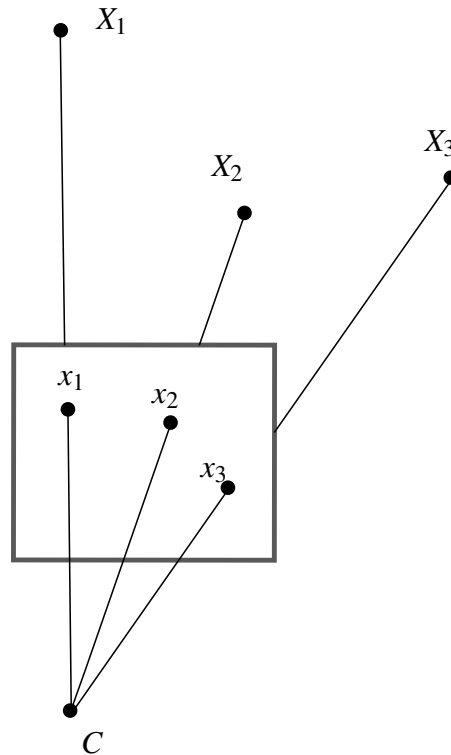


Figure 2.2: A calibrated camera measures angles between pairs of image points and therefore the angles between the imaged 3D world points as observed from the camera position C . To compute the camera position and orientation three such angles and their corresponding 3D point coordinates are necessary.

Different strategies for solving this equation system can be found in [Haralick et al., 1994]. The problem can be further simplified by assuming a known up vector of the camera. The up vector can for example be estimated by using an inertial measurement unit (IMU). The calibrated case and a projective, undistorted case are investigated in [Kukelova et al., 2011]. The minimum number of 2D-3D point correspondences for the calibrated absolute pose problem with known up vector is reduced to two.

2.3.2 6DOF Localization for Panoramic Images

The classical three point perspective pose estimation ($P3P$) problem is also applicable for localizing panoramic images. Figure 2.3 shows the geometry of the problem.

For pinhole camera models a known camera calibration means that the image measurements m_i can be converted to rays v_i and their pairwise angle $\angle(v_i, v_j)$ can be measured. In this case three known 3D points X_i and their corresponding image measurements m_i give rise to 3 pairwise angle measurements. These are sufficient to compute a finite

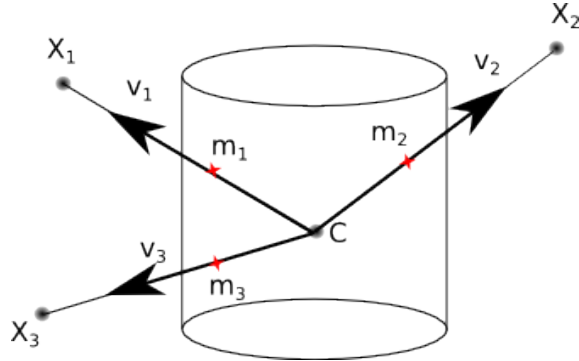


Figure 2.3: The geometry of the $P3P$ problem for panoramic camera models is the same as for pinhole models. The law of cosines relates the unknown distances of 3D points and the camera center $x_i = \|C - X_i\|$ with the pairwise angles $\angle(v_i, v_j)$ of the image measurement rays.

number of solutions for the camera location and orientation. Converting our panoramic image measurements m_i to rays v_i and thus to pairwise angle measurements $\angle(v_i, v_j)$ leads to the same equation system as in the pinhole case.

This is the same polynomial system as in the case of the more commonly used pinhole camera model and can be solved with the same techniques. The main difference is that in the pinhole case the camera calibration matrix K is used to convert image measurements to vectors and therefore pairwise Euclidean angle measurements, while in our case, the rays are defined by the geometry of the cylindrical projection.

Optimization The three point pose estimation is used in a RANSAC scheme to generate hypotheses for the pose of the camera. After selecting inlier measurements and obtaining a maximal inlier set, a non-linear optimization for the pose is applied to minimize the reprojection error between all inlier measurements m_i and their corresponding 3D points X_i .

To avoid increasing error distortions towards the top and bottom of the panoramic image we defined a meaningful reprojection error that is independent of the location of the measurement m_i on the cylinder. We approximated the projection locally around the measurement direction with a pinhole model and applied a constant rotation R_i to both the measurement ray v_i and the camera pose to move the measurement ray into the optical axis. The rotation R_i is defined such as

$$R_i v_i = (0 \ 0 \ 1)^T.$$

The remaining degree of freedom can be chosen arbitrarily. This rotation R_i is constant for each measurement ray v_i and therefore not subject to the optimization.

The imaging model for the corresponding 3D point X_i is then given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \text{proj}(R_i T X_i), \quad \text{where}$$

$$\text{proj}((x \ y \ z)^T) = \begin{pmatrix} x \\ y \\ z \end{pmatrix},$$

and T is the camera pose matrix representing a rigid transformation.

The optimization minimizes the sum of all squared reprojection errors as a function of the camera pose matrix T :

$$E(T) = \sum_i \|\text{proj}(R_i v_i) - \text{proj}(R_i T X_i)\|^2 = \sum_i \|\text{proj}(R_i T X_i)\|^2.$$

Note that the rotation R_i rotates the measurement into the optical axis of the local pinhole camera and therefore the projection $\text{proj}(R_i v_i) = (0 \ 0)^T$. The camera pose matrix T is parameterized as an element of $SE(3)$, the group of rigid body transformations in 3D. The solution T_{min}

$$T_{min} = \arg \min_T E(T)$$

2.4 Structure From Motion

Structure from motion methods compute camera positions and orientations from image to image point correspondences. 3D points are computed from these correspondences implicitly. To solve SfM for an image data set, feature matching methods are necessary to find 2D-2D image point correspondences and multiple view geometry methods are used to infer the scene geometry.

2.4.1 Multiple View Geometry

SfM finds a solution to the reconstruction problem

$$\mathbf{x}_{ij} = P_i \mathbf{X}_j$$

where \mathbf{x}_{ij} are points in image i that correspond to the 3D point \mathbf{X}_j and is projected by the 3×4 camera matrix P_i . The structure of P_i depends on the used projection model, 2D and 3D points are represented as homogeneous vectors. A set of 2D points that are projections of the same 3D point \mathbf{X}_j are called correspondences and usually obtained by feature point matching methods. One possible way of interpreting the point correspondences is to think of correspondences as edges in a bipartite hypergraph $H = (V, E)$ that

connect cameras and structure points. The vertices V consist of all images and 3D points $V = \{P_1..P_n, X_1..X_m\}$ and are connected by 2D point correspondences. These 2D point correspondences $E = \{\{x_{i1}, \dots, x_{j1}\}, \dots, \{x_{km}, \dots, x_{lm}\}\}$ define the visibility of a 3D point in an image.

The full hypergraph formulation is usually not used by SfM algorithms to represent connectivity. Popular approximations of correspondence graphs only contain cameras in the set of vertices $V = \{P_1..P_n\}$ and connect two images if enough point correspondences are found between an image pair $P_i P_j$.

2.4.2 Bundle Adjustment

Real image measurements never lead to perfect solutions of $\mathbf{x}_{ij} = P_i \mathbf{X}_j$. Imperfections come for example from feature detector noise, correspondence outliers and deviations from the pinhole camera model. It is well known [Triggs et al., 2000] that bundle adjustment (BA) with a square cost function is the Maximum Likelihood solution for reconstruction problems that are contaminated by Gaussian noise. Bundle adjustment minimizes the sum of the geometric distances of all image measurements \mathbf{x}_{ij} and their corresponding projected 3D points $P_i \mathbf{X}_j$ in image space:

$$\min_{P_i, \mathbf{X}_j} \sum C(d(\mathbf{x}_{ij}, P_i \mathbf{X}_j))$$

$$C_s(\delta) = \delta^2,$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance of the measurement and the projected point, C is the cost function C_s , for example a quadratic cost function for classical bundle adjustment. Bundle adjustment is a non-linear optimization problem that needs an initial estimation and is only a refinement step. The optimization problem can be solved using standard techniques, [Nocedal and Wright, 2006] gives an overview. Because of the sum of squares structure, the Gauss-Newton simplification for non-linear problems is usually applied and the normal equations are solved using Levenberg-Marquardt regularization.

2.4.3 Non-Linear Least Squares

The bundle adjustment cost function

$$f(P_i, \mathbf{X}_j) = \sum d(\mathbf{x}_{ij}, P_i \mathbf{X}_j)^2$$

can be seen as generic least squares problem

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2$$

where all m projections are summed up as residual r_j . If we write the residuals r_j as a vector valued function

$$r(x) = (r_1(x), \dots, r_m(x))^T,$$

the cost function $f(x)$ can be written as the norm of the vector valued function of individual residuals, $f(x) = \frac{1}{2} \|r(x)\|_2^2$. The Jacobian of the cost function $J(x)$ is defined as

$$J(x) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \frac{\partial r_m}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \Delta r_1(x)^T \\ \Delta r_2(x)^T \\ \vdots \\ \Delta r_m(x)^T \end{bmatrix}.$$

The gradient of the cost function can now be written as

$$\Delta f(x) = \frac{1}{2} \Delta \|r(x)\|_2^2 = \frac{1}{2} \Delta (r_1(x)^2 + r_2(x)^2 + \dots + r_m(x)^2) = \frac{1}{2} \Delta \sum_{j=1}^m r_j(x)^2,$$

and by applying the chain rule

$$\Delta f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial \sum r_j^2}{\partial x_1} \\ \frac{\partial \sum r_j^2}{\partial x_2} \\ \vdots \\ \frac{\partial \sum r_j^2}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \sum r_j(x) \frac{\partial r_j}{\partial x_1} \\ \sum r_j(x) \frac{\partial r_j}{\partial x_2} \\ \vdots \\ \sum r_j(x) \frac{\partial r_j}{\partial x_n} \end{pmatrix} = \sum_{j=1}^m r_j(x) \Delta r_j(x) = J(x)^T r(x).$$

The Hessian matrix can be written in a similar way as a function of $J(x)$:

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \Delta^2 r_j(x),$$

which gives the first order Hessian approximation $H(x) \approx J(x)^T J(x)$.

The general Newton method $\Delta^2 f(x)p = -\Delta f(x)$, where p is the search direction, becomes with this approximation $J^T J p = -\Delta f$ and is known as Gauss-Newton approach. Levenberg-Marquardt is a popular extension to solve this equation more robustly when J is (nearly) singular. The idea is to add a scaled identity matrix, $(J^T J + \lambda I)p = -\Delta f$.

When λ is decreased, the equations approach the original Gauss-Newton direction, when λ is increased the system is forced to be non-singular. This can be also interpreted as a simple trust region method, see [Nocedal and Wright, 2006].

For the BA problem $J(x)$ has important properties. The optimization parameters x can be grouped into camera parameters and structure parameters, so $J(x)^T J(x)$ has a special block structure and can be inverted more efficiently. Elements of $J(x)$ are only non-zero when a measurement is visible in the camera, which means J and $J(x)^T J(x)$ is sparse. The typical BA implementation optimizations that result from this properties are summarized in [Engels et al., 2006], furthermore the sparsity of the Hessian matrix for inversion can easily be exploited by using sparse linear algebra methods, for example [Davis, 2008] works well for BA problems.

2.4.4 SfM Initialization

Epipolar Geometry Initializing bundle adjustment from multiple views is an active research topic, starting with two views is well understood. Given two images of a scene the epipolar geometry describes how the two views are related. The basic primitives for this relation are corresponding image points $\mathbf{x} = P\mathbf{X}$ and $\mathbf{x}' = P'\mathbf{X}$, where the 3-space point \mathbf{X} is imaged by two cameras. The connection of the image point \mathbf{x} with \mathbf{x}' is described by a 3×3 matrix F , called the fundamental matrix.

Geometry of Two Views Given an 3-space point \mathbf{X} and its image points $\mathbf{x} = P\mathbf{X}$ and $\mathbf{x}' = P'\mathbf{X}$, these three points and the camera centers \mathbf{C}, \mathbf{C}' lie on a plane π , the epipolar plane. With the knowledge of one imaged point \mathbf{x} , the location in the second image of the corresponding point can be restricted. Figure All that is known from one image is that the line on which the point must lie. Projecting this reprojection line of the image point \mathbf{x} with the second camera, this reprojection line is again imaged as a line, the epipolar line \mathbf{l}' . The corresponding image point \mathbf{x}' must lie on this line. The line defined by the two camera centers is called baseline. Intersecting this line with the two image planes (or projecting the camera center of the other camera) gives the two points \mathbf{e}, \mathbf{e}' , called epipoles. The location of the epipoles is therefore independent of the imaged scene, but all epipolar lines intersect in these points.

Fundamental Matrix The fundamental matrix F , a 3×3 matrix of rank 2, describes algebraically the relation of two views. Geometrically, the fundamental matrix may be defined using a point transfer via a plane. This gives the connection of the image point \mathbf{x} in one image and the corresponding epipolar line \mathbf{l}' in the other,

$$\mathbf{l}' = F\mathbf{x}. \quad (2.21)$$

Because the point \mathbf{x}' lies on \mathbf{l}' , $\mathbf{x}'^T \mathbf{l}' = 0$, this leads to the correspondence condition that is valid for any pair of corresponding points

$$\mathbf{x}'^T F \mathbf{x} = 0. \quad (2.22)$$

Computation of the Fundamental Matrix From the correspondence condition $\mathbf{x}'^T F \mathbf{x} = 0$ a linear method for computing F from image point correspondences alone can be derived. With the corresponding point pair $\mathbf{x} = (x, y, 1)^T$, $\mathbf{x}' = (x', y', 1)^T$ the following equation must be satisfied

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0, \quad (2.23)$$

where f_{ij} is the element of row i and column j of the fundamental matrix F . By writing the elements of F as the 9-vector

$\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$ this can be expressed as inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = \mathbf{c} \mathbf{f} = 0. \quad (2.24)$$

When the vector \mathbf{c}_i is composed with the elements of the point pair i , the linear equations

$$A \mathbf{f} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_n \end{bmatrix} \mathbf{f} = \mathbf{0} \quad (2.25)$$

must be satisfied for n corresponding point pairs. This provides the basics for computing the fundamental matrix F . The minimum number of corresponding points is 7. To solve for the elements of F with a simple linear equation system, 8 correspondences are necessary. This is called the 8-point algorithm and originates from [Longuet-Higgins, 1987]. As [Hartley, 1997] shows, this approach has numerical problems in practice. The main reasons are a poor conditioning number of the matrix A and errors in the measurements are amplified by the products of Equation 2.24. In [Hartley, 1997] a simple image data normalization is proposed. This normalization increases the stability of the linear 8-point algorithm in many cases.

Algorithm 2.1 summarizes the linear computation of F using the normalized 8-point algorithm. When the image points are normalized as described in Algorithm 2.1, this means that the average image point has the coordinate $\mathbf{x} = (1, 1, 1)^T$. Anisotropic scaling of the image points using their principal moments would be another alternative. The singularity constraint is enforced before de-normalizing the matrix F because the smallest elements of F are the most important ones for computing epipolar lines and would be changed the most by the rank 2 enforcement in a de-normalized coordinate system. If

Algorithm 2.1 Normalized 8-Point Algorithm

- 1: Input: $n \geq 8$ point correspondences, $x_i^T F x_i = 0$ must be satisfied
- 2: Normalize: Translate image points so that center of gravity of points in each image is the origin and scale image points so that the RMS distance to the origin is $\sqrt{2}$.
- 3: Solve the linear Equation 2.25.
- 4: Enforce rank 2 condition.
- 5: De-normalize F .

the two 3×3 transformation matrices T, T' were used for the point normalization and computation of the normalized fundamental matrix \hat{F} , the de-normalized fundamental matrix F is given by $F = T'^T F T$. More motivation for the steps and choices can be found in [Hartley, 1997].

Using a projective reconstruction the computed fundamental matrix F may be further refined by minimizing a geometric distance with an iterative numerical algorithm. The projective reconstruction and the fundamental matrix are connected by the choice of a canonical camera pair, usually $P_1 = [I|\mathbf{0}]$ and $P_2 = [[\mathbf{e}']_{\times} F_{12} | \mathbf{e}']$ is used.

2.4.5 Essential Matrix

The essential matrix is a special case of a fundamental matrix where the image coordinates are normalized. Compared to the fundamental matrix, the essential matrix has fewer degrees of freedom and the two non-zero singular values are equal.

Normalized image coordinates are defined as $\hat{\mathbf{x}} = K^{-1} \mathbf{x}$, where the calibration matrix K of the camera $P = K[R|\mathbf{t}]$ is known. Using normalized coordinates $\hat{\mathbf{x}} = [R|\mathbf{t}]\mathbf{X}$ with the camera matrix written as $K^{-1}P = [R|\mathbf{t}]$ is called a normalized camera matrix. The fundamental matrix of the normalized camera matrices $P = [I|\mathbf{0}]$ and $P' = [R|\mathbf{t}]$ is called the essential matrix. Like in the case of the fundamental matrix for corresponding points $\mathbf{x}, \hat{\mathbf{x}}$ the following holds:

$$\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0 \quad (2.26)$$

Comparing $\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = \mathbf{x}'^T K'^{-T} E K^{-1} \mathbf{x} = 0$ and $\hat{\mathbf{x}}'^T F \hat{\mathbf{x}} = 0$ shows that:

$$E = K'^T F K \quad (2.27)$$

K and K' have full rank, the fundamental matrix F has rank 2, so the rank 2 property of the essential matrix is automatically fulfilled. For a given fundamental matrix F , the

equality of the two non-zero singular values is now used to define an objective function which is optimized by a numerical technique.

Computation of the Essential Matrix The geometry of the essential matrix problem is illustrated in [Horn, 1989] and are represented in the coplanarity constraints $\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0$. Current algorithms to compute the essential matrix from five correspondences all use fact that two singular values of E are equal and not zero. This property can be written as [Faugeras, 1993]

$$EE^T E - \frac{1}{2} \text{tr}(EE^T) E = 0, \quad (2.28)$$

and is the key element for all state of the art solvers [Nistér, 2004, Li and Hartley, 2006]. The basic strategy is to compute the null-space of all potential essential matrices from $\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0$ with five correspondences. This null-space is then plugged into the cubic constraints of 2.28, which leads to a polynomial equation system that has to be solved.

Triangulation Given a pair of camera matrices P, P' , computed from the fundamental matrix, the 3D points \mathbf{X}_i can be computed from the 2D point correspondences $\mathbf{x}_i, \mathbf{x}'_i$. Geometrically a 3D point is given by the intersection of the back-projection of the two image points $\mathbf{x}_i, \mathbf{x}'_i$. Because of measurement errors the two back-projected lines will in general not intersect. Simply taking the midpoint of the common perpendicular of the two rays is not the optimal solution in the projective case. The midpoint does not correspond to the midpoint of an Euclidean reconstruction. A good projective triangulation method should be invariant under a projective transformation H . Various triangulation methods are presented and compared in [Hartley and Sturm, 1997].

Linear Triangulation With the correspondences \mathbf{x}, \mathbf{x}' and the camera matrices P, P' a linear equation for the triangulation problem can be formulated. This method is not invariant under a projective transformation but has the advantage of being easily generalizable to more than two views.

The unknown 3D point \mathbf{X} has to satisfy $\mathbf{x} = P\mathbf{X}$ and $\mathbf{x}' = P'\mathbf{X}$. The homogeneous scale factor in $\mathbf{x} = P\mathbf{X}$ can be eliminated with a cross product. Using a cross product \mathbf{X} has to satisfy now $\mathbf{x} \times (P\mathbf{X}) = 0$ and $\mathbf{x}' \times (P'\mathbf{X}) = 0$. When \mathbf{p}^{iT} is the i th row of P and the cross product written out $\mathbf{x} \times (P\mathbf{X}) = 0$ can be written as

$$\begin{aligned} x(\mathbf{p}^{3T} \mathbf{x}) - (\mathbf{p}^{1T} \mathbf{x}) &= 0 \\ y(\mathbf{p}^{3T} \mathbf{x}) - (\mathbf{p}^{2T} \mathbf{x}) &= 0 \\ x(\mathbf{p}^{2T} \mathbf{x}) - y(\mathbf{p}^{1T} \mathbf{x}) &= 0. \end{aligned}$$

Two of these equations are linearly independent. Using two such equations from both cameras the linear system

$$\begin{bmatrix} x(\mathbf{p}^{3T} \mathbf{x}) - (\mathbf{p}^{1T} \mathbf{x}) \\ y(\mathbf{p}^{3T} \mathbf{x}) - (\mathbf{p}^{2T} \mathbf{x}) \\ x'(\mathbf{p}'^{3T} \mathbf{x}) - (\mathbf{p}'^{1T} \mathbf{x}) \\ y'(\mathbf{p}'^{3T} \mathbf{x}) - (\mathbf{p}'^{2T} \mathbf{x}) \end{bmatrix} \mathbf{X} = \mathbf{A}\mathbf{X} = 0 \quad (2.29)$$

can be composed.

The solution from the linear triangulation method may be used as an initialization for a non-linear geometric error minimization. The reprojection error of the 3D point \mathbf{X} gives the geometrically meaningful error. There exists also a non-iterative method to compute the optimal solution. This method is described in [Hartley and Sturm, 1997] and requires the solution of a sixth degree polynomial.

2.4.6 SfM for Multiple Views

A simple SfM algorithm can already be built from the tools described. Initializing two cameras using the essential matrix, triangulating 3D points and then adding new views using the absolute pose localization. A simple SfM algorithm is outlined in Algorithm 2.2. The well known SfM pipeline of [Snavely et al., 2006] is basically an implementation of this algorithm.

Another promising strategy for calibrated SfM is to split the problem into computing rotations and then translations and structure. Globally consistent rotations can be obtained by chaining rotations [Govindu, 2004]. Camera translations (and structure) can then be solved using convex optimization tools, a complete overview of such a SfM approach is for example given in [Zach and Pollefeys, 2010]. For large scale use, computational costs are still prohibitive though and failure modes are not well known yet.

Algorithm 2.2 A Simple SfM Algorithm

- 1: Initialize camera pair with essential matrix
 - 2: Triangulate feature points of camera pair
 - 3: **for** all other images i **do**
 - 4: Register image i using 6DOF localization
 - 5: Triangulate new feature points
 - 6: Bundle adjustment
 - 7: **end for**
-

2.5 Point Correspondences

The geometric relations described above need point correspondences as input. The basic problem statement is, given two images of a rigid scene, compute points that originate from the same world points. This means one has to identify points in both images and then establish how these points correspond. The correspondence problem consists of two parts: Firstly, the identification of points in the images that are suitable for image matching, these points are called interest points. Secondly, correspondences of interest points from different images are identified, this is usually done based on the appearance of image regions around the interest point. The assumption is that points that originate from the same 3D point in the world should have a similar appearance in images.

2.5.1 Interest Points

Points that are suitable for establishing correspondences in images have to be detected reliably from different view points. Structure tensor based corner detectors, such as the Harris detector, evaluate intensity changes in the image. The detector response is only invariant to small changes in scale and is mainly used in short baseline image matching. For large, unstructured SfM data, additional detection of the scale of the interest point is essential.

Corner Detector When $I(x,y)$ is the intensity of a pixel the local structure matrix M is defined as

$$M = \begin{bmatrix} (\frac{\partial I}{\partial x})^2 & \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & (\frac{\partial I}{\partial y})^2 \end{bmatrix}. \quad (2.30)$$

The eigenvectors of M encode edge directions, a corner has two strong edges i.e. a motion in any direction causes an intensity change. For example, the cornerness function C can now be defined as

$$C = \det(M) - \alpha * (\text{trace}(M))^2, \quad (2.31)$$

which is the classical Harris [Harris and Stephens, 1988] corner detector. Other heuristics are possible to define the cornerness of the structure tensor of course. A comparison of different cornerness functions their performance under various image transformations can be found in [Schmid et al., 2000]. Corners are the local maxima of this function, the parameter α controls the sensitivity of the operator. The squared image derivatives may be smoothed to avoid a response to noise. A different approach that does not compute the structure tensor is for example described in [Rosten and Drummond, 2006], where

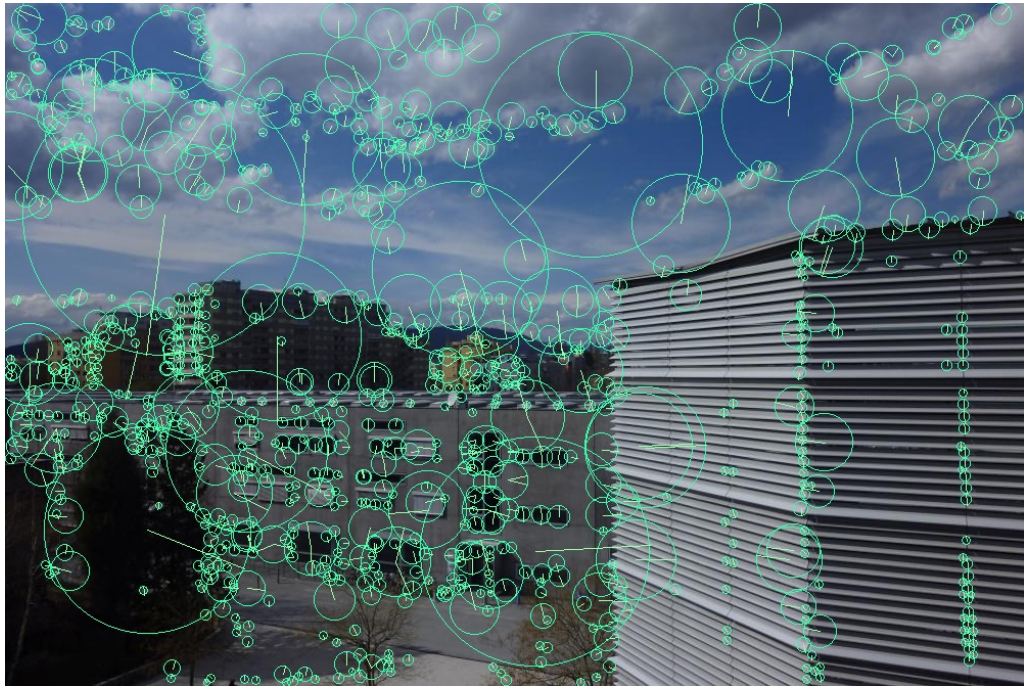


Figure 2.4: SIFT features

a segment test around a potential corner is evaluated with a decision tree to speed up computations.

Scale Space Detector The cornerness function of a potential feature point can be extended from the 2D image plane response to add a scale dimension. This extension is significant creating correspondences in large, unstructured image collections. Scale space theory is developed in [Lindeberg, 1994], the Scale Invariant Feature Transform(SIFT) detector made scale space particularly popular [Lowe, 1999, Lowe, 2004]. Figure 2.4 shows the response of this detector. SIFT uses Differences of Gaussians (DoG) to build a scale-space.

$$DoG(x, y, \sigma) = G(x, y, k\sigma) * I(x, y) - G(x, y, \sigma) * I(x, y),$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The DoG scale-space of an image I is sampled and local extrema are selected. Integral images have been used to accelerate the scale-space computation further, [Grabner et al., 2006, Bay et al., 2006, Agrawal et al., 2008]. The used box filter operations are independent of the kernel size when computed with integral images, SIFT uses image sub-sampling to speed up convolution costs.

2.5.2 Feature Descriptors

After assigning a feature point location and scale, a description has to be found for this feature point, so that it can be found in another image. A description should be as invariant to changes of illumination, scale, rotation, affine and perspective distortion as possible. Image gradient histograms, as used in [Lowe, 2004], are a popular choice, a comparison of different approaches can be found in [Mikolajczyk and Schmid, 2003]. A brief overview of how invariant descriptor are constructed is given here:

- **Illumination:** Image gradients are invariant to a constant additional brightness changes, multiplicative brightness changes are compensated by normalizing the descriptor vector.
- **Scale:** Scale changes are usually handled by the detector.
- **Rotation:** In-plane rotation is handled by assigning a dominant orientation of the patch around a feature point and rotating the patch to a canonical orientation.
- **Distortion:** Imaging distortions are handled in the SIFT case by strong gradient histogram quantization and ambiguous assignment of gradients to histograms at border pixels.

2.5.3 Feature Matching

Point correspondences are found by comparing descriptor vectors. This is basically a high dimensional nearest neighbor problem. Computing the euclidean distance of two sets of descriptors can be represented as a matrix multiplication,

$$C = \begin{bmatrix} v_{21} \\ - \\ v_{21} \\ \vdots \\ v_{2m} \end{bmatrix} [v_{11} | v_{11} \dots | v_{1n}]$$

where the elements C_{ij} represent $\cos\phi$ of the descriptors v_{1i} and v_{2j} when all descriptors are normalized to unit length. Nearest neighbor approximation techniques like [Muja and Lowe, 2009] can also be used to speed up the search.

2.5.4 Large Scale Feature Matching

To reconstruct a scene all image pairs have to be matched. Matching these pairs scales quadratically with the number of images. This is not feasible for larger data sets, so an



Figure 2.5: A larger high resolution image collection. To reconstruct the scene all image pairs have to be matched. Matching these pairs scales quadratically with the number of images, so efficient approximations become necessary.

approximation is necessary. The basic strategy to approximate pair-wise feature matching is to only match each image to n suitable candidates, where n is a small constant. These suitable candidates can be found by object retrieval methods. In [Sivic and Zisserman, 2003] text retrieval methods for large scale text search are used to speed up image-based object retrieval. These ideas are further extended in [Nister and Stewenius, 2006] to improve retrieval performance.

The major building blocks of fast approximated image matching are:

- **Visual Vocabulary:** Feature descriptors are clustered into a fixed visual vocabulary in an off-line training stage. Usually k -means is used. The resulting cluster topology can be flat or hierarchical.
- **Inverted File:** For image matching, feature descriptors are assigned to the most similar cluster centers per image. For each assigned feature descriptor, the originating image ID is stored, the so called inverted file.
- **Scoring:** Potentially similar images are found by searching for documents, i.e. images, that contain similar visual words. Documents that contain a visual word are

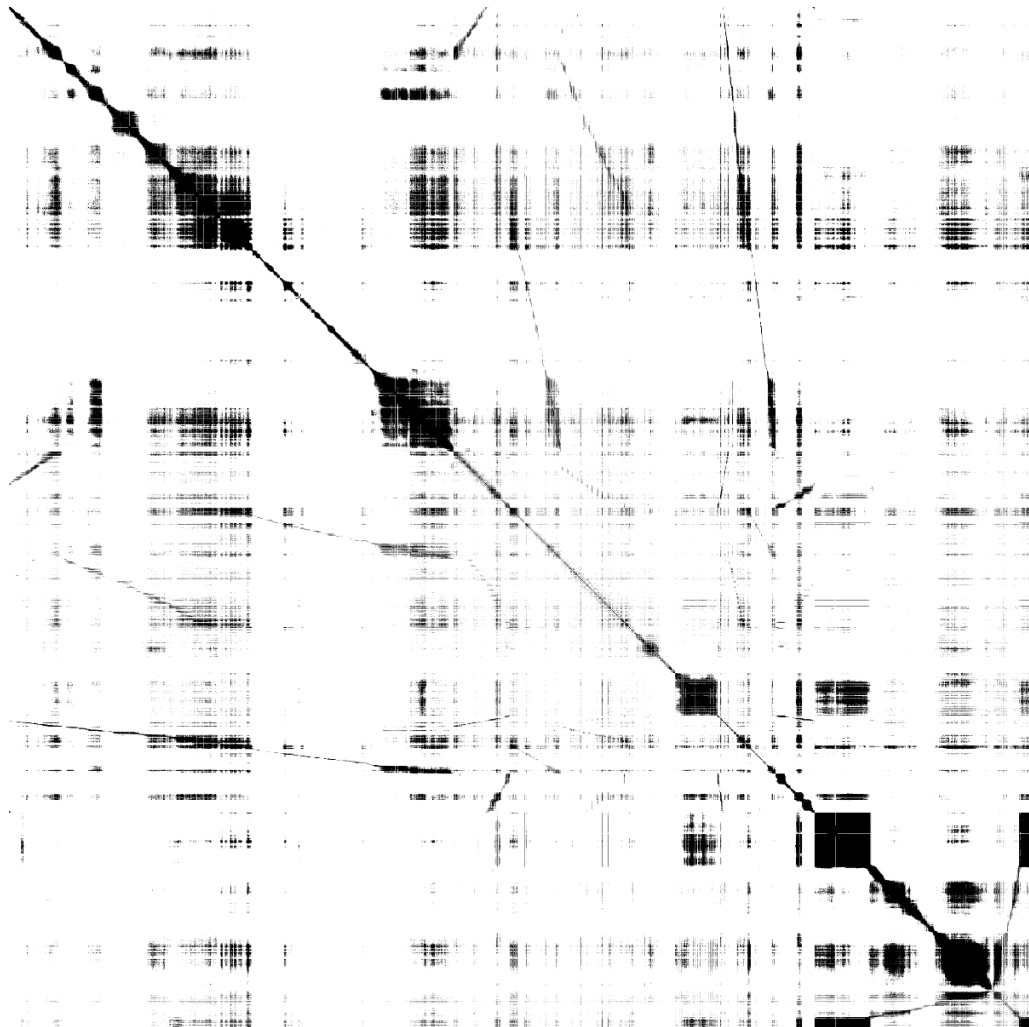


Figure 2.6: Pair-wise image matching matrix: Element (i, j) represents similarity if image i and image j . To speed up pair-wise image matching, the image matching matrix is approximated by a sparse matrix using bag of words methods. Darkness indicates similarity.

already stored in the inverted file, so the search query can be basically computed by accumulating the inverted file entries for each visual word from the query image.

Figure 2.6 illustrates this matching approximation by representing the pair-wise visual vocabulary matching as a matrix. Each element (i, j) corresponds to the scoring result of image i and image j . Detailed feature matching with geometric verification is then carried out on the best n candidates for each image.

This bag of feature matching approximation can of course miss some relevant pair-wise matches. A popular heuristic to reduce missing image pairs is query expansion

[Chum et al., 2007]. The best n query results are simply used to start queries again, the final result is a concatenation of all retrieval result. This idea again originates in text based search.

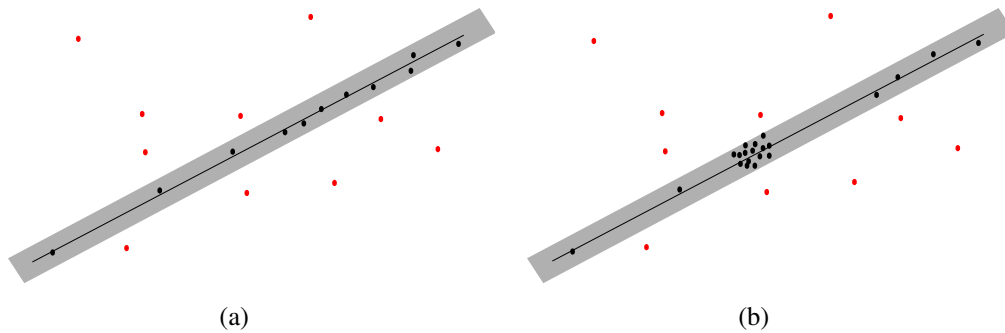


Figure 2.7: Line fitting RANSAC illustration: Black dots are inliers, red dots outliers, the shaded gray area illustrates the distance threshold t . (a) A classical RANSAC line fitting situation. (b) An example where it is not sufficient to draw inlier points to compute a good hypothesis. Most points in the point cluster are inliers for the shown model but point pairs from this cluster can define arbitrary line directions.

2.6 Robust Estimation

Point correspondences obtained by image matching methods are contaminated by noise. The two major reasons of noisy correspondences are feature point location errors and image matching errors. The feature point location error is introduced by the feature detector because the assumptions that are build into these detectors do not hold under general perspective projection. This location error follows roughly a Gaussian distribution and the influence on the estimation error is minimized by least squares techniques. Image matching errors are introduced by wrong descriptor matches and therefore a wrong point correspondence. This assignment failure introduces geometric errors that are outliers.

Currently the most popular methods to handle outliers in robust estimation problems are based on the Random Sample Consensus (RANSAC) [Fischler and Bolles, 1981] framework. RANSAC is based on the idea of computing an estimation repeatedly from a minimal, randomly selected, subset of data points to compute a hypothesis, and then evaluating the quality of the hypothesis using more data. This hypothesis generation and evaluation step is carried out n times and the best result is selected. The metric to compute the quality of an estimation is the number of points that are inliers with respect to a distance parameter t and the model estimation. One reason for the popularity of RANSAC in computer vision is that many estimation problems compute an estimation error in image space. This means the distance parameter t can be expressed in pixel and a threshold that separates detector noise from matching outliers can be easily selected.

In [Hartley and Zisserman, 2004] some basic guidelines for selecting parameters are presented. The number N of samples that have to be drawn to select only inlier points for hypothesis generation depends on the outlier proportion ϵ and the sample size of data

points s that are necessary to compute a hypothesis:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)}, \quad (2.32)$$

where p is the desired probability of drawing an inlier set, this is usually set to $p = 0.99$.

Intuitively it is clear that a higher outlier proportion requires more iterations to find an inlier set and a larger sample size increases this number too. This estimation of N is overly optimistic because drawing an inlier set is not sufficient to obtain a correct hypothesis. Figure 2.7(a) shows a simple 2D line fitting example where Equation 2.32 holds and Figure 2.7(b) shows an example where this assumption is clearly violated.

2.6.1 RANSAC Improvements

Conceptually RANSAC consists of only two stages, a model computation stage and model verification stage. Improvements in terms of computational efficiency compared to the algorithm described in [Fischler and Bolles, 1981] can be achieved by reducing the number of models that are computed and reducing the number of data points that are used to evaluate a model. An overview of many methods to speed up RANSAC is presented in [Raguram et al., 2008].

2.6.2 Model Computation

The original RANSAC algorithm draws samples uniformly from all data points. When a data point can be associated with a reliability measure it can be more efficient to assign a higher probability for drawing points with a higher quality measure. The intuition is that a good hypotheses is computed much earlier and fewer sampling iterations are necessary. For image correspondences the descriptor matching distance can be used as reliability measure for example. This idea is investigated in detail in [Chum and Matas, 2005] and called PROSAC, it is one of the most frequently used RANSAC modifications. Preemptive RANSAC [Nistér, 2003] is another popular modification that computes all hypothesis before evaluating them in parallel.

2.6.3 Model Verification

Another optimization possibility is to minimize the number of data points that are used to evaluate a hypothesis. A sequential probability ratio test (SPRT) is developed in [Chum and Matas, 2008], where Wald's sequential decision test theory is used to motivate an approach for speeding up the hypothesis evaluation stage. This SPRT is provably optimal when the inlier ratio is known.

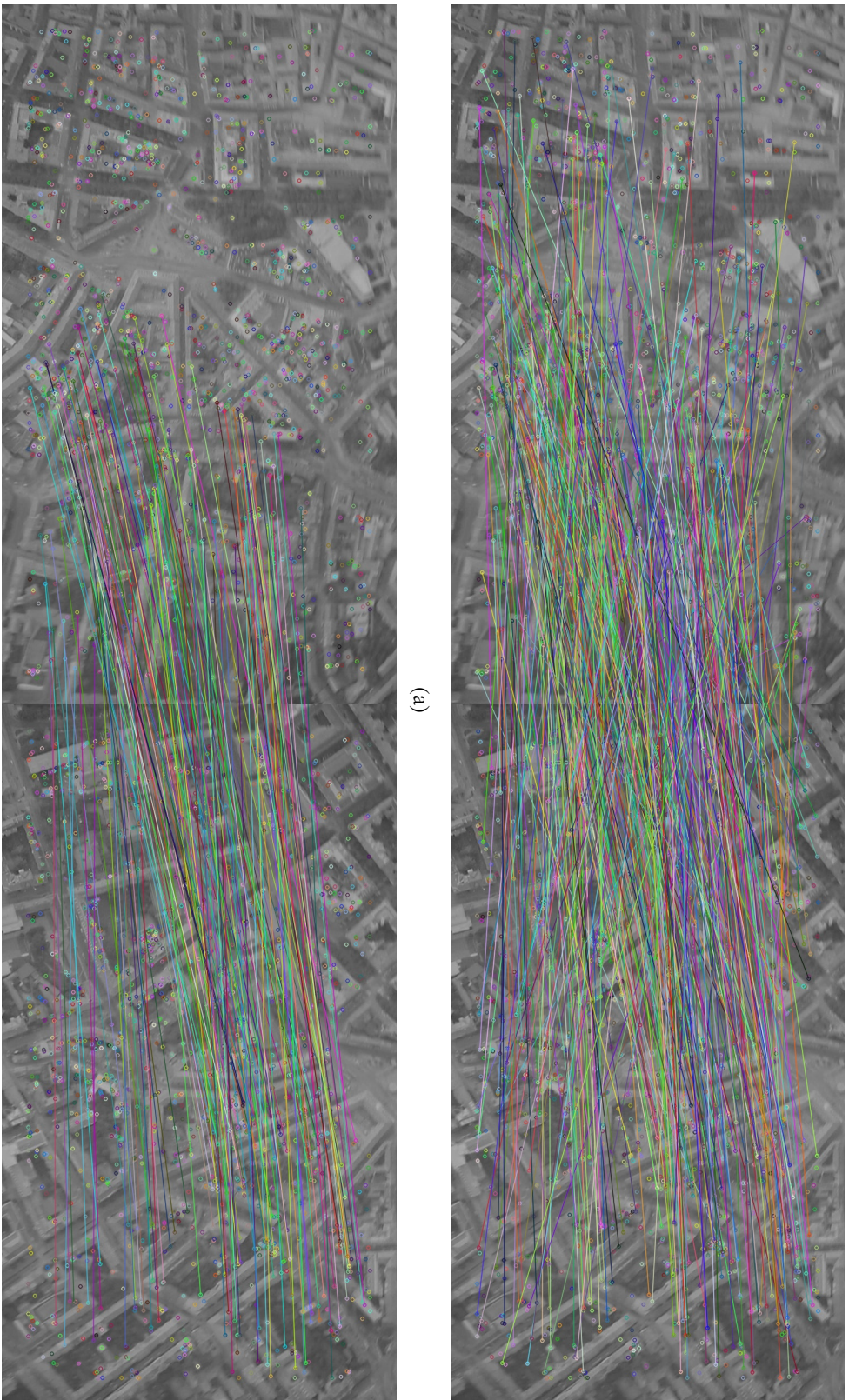


Figure 2.8: Geometric Model Verification: (a) Two images are matched using the best euclidean descriptor distances, point pairs that satisfy a fixed minimum matching quality are shown. (b) The Essential matrix is used in a RANSAC framework to compute a set of inlier correspondences from the same matching input as in Figure (a).

Part I

Robust Scene Reconstruction

Chapter 3

Robust Incremental Structure from Motion

3.1 Introduction

If we want to use SfM to create 3D data for AR tracking one of the first problems that arise is that we usually have a specific place in mind that we want to cover. SfM pipelines like [Snavely et al., 2006] usually use an highly redundant image collection as input and create a reconstruction of this scene. For tracking we want to cover an object or location and acquire the data specifically for the task. The resulting data is usually of sequential nature and makes incremental structure from motion harder. This sequential nature of the data can be modeled by a Markov model,

$$p(x_1, \dots, x_N) = \prod_{i=1}^N p(x_i | x_1, \dots, x_{i-1}), \quad (3.1)$$

where $p(x_1, \dots, x_N)$ represents the probability of a successful reconstruction of N images by adding one image at a time. If we assume for simplicity that we only have a first order Markov process where $p(x_N)$ only depends on $p(x_{N-1})$ and simply assume a constant probability of success c for all images we see that

$$p(x_N) = p(x_i | x_{i-1}) = \prod_{i=1}^N c. \quad (3.2)$$

With $c < 1$ the likelihood of a successful reconstruction decreases as we add images, recovery after failure is not possible.

With this simple model in mind we motivate a method that can build large reconstructions without depending on global structure ($p(x_{N-1})$ in the Markov model) for outlier

rejection and starts from the most reliable parts of the data sets. The two main ideas are a strategy to identify reliable subsets of images that have the highest mutual compatibility and an ordering of the reconstruction buildup that gives higher priority to these subsets and merges new information according to this ordering. Another advantage of our buildup strategy is that loop closing is done unbiased by drift in spite of adding correspondence data incrementally. Correspondence information is only locally verified and merged into reconstructions without immediately removing outliers based on globally reconstructed structure. We demonstrate the robustness and scalability of our approach on several large reconstructions from unordered sets of images and indicate the achieved accuracy by preserving the topology of the 3D structure and cameras.

Current SfM methods may be classified based on the image data they use. Two major types of image sources are still images and video sequences. Recent trends in photo community websites and the ubiquitous availability of consumer grade compact cameras have shifted the research interest towards reconstruction from unordered image data sets [Snavely et al., 2008b]. This choice of input data usually has strong implications on the selection of algorithms used to solve the SfM problem. Unordered sets of still images rely on wide baseline image matching techniques to establish correspondence information and the SfM problem may be solved for all input images simultaneously or in an incremental way. Two examples of this type of systems are [Martinec and Pajdla, 2007] and [Snavely et al., 2006].

Incremental methods can add new images into existing reconstructions when they become available and make the system more flexible. The same is true for using unordered sets of input images instead of video streams. Compared to feature point tracking in video sequences, wide baseline matching provides additional links between images separated by time. Recent image matching methods make the harder wide baseline matching problem more tractable and scalable.

We focus our attention in this work on making the incremental still image based reconstruction process more robust and flexible. We base our reasoning about feature track compatibility and image connectivity on image triplets because they are well known to be more robust against false feature matches and naturally extend to graph based representations. Given a set of pairwise correspondences for input images, we transform these correspondence into corresponding image triplet reconstructions. This reduces the number of outliers compared to epipolar geometry, verifies track compatibility locally using pairs of triplet reconstructions, detect overlap in this triplet representation efficiently with overlapping views and find starting points for the reconstruction that are most reliable. Another important aspect of our work is that we do not use the 3D points in the evolving reconstructions to generate tracks of reprojection inliers. This common strategy makes the implicit assumption that no drift is present and needs an explicit loop closing strat-

egy. We avoid this step by closing loops implicitly, using only the local correspondence information from triplet to triplet registration.

We motivate the concept of *local correspondences* with the example shown in Figure 3.1. In a reconstruction from a small set of unordered images, drift builds up due to slight camera calibration errors. This reconstruction was built by registering individual views to the evolving, global structure. Towards the end, correspondences that would close the loop are immediately discarded as the accumulated drift implies a large reprojection error. As a result the reconstruction is distorted and the loop cannot be closed (see right image in Figure 3.1) because the required correspondences are classified as outliers. In general, incremental SfM pipelines usually discriminate correspondences into in- and outliers by evaluating some robust estimator on the global structure obtained in the build up process. When drift is present this can introduce severe bias and correct classification degrades with the number of added images. We propose to avoid this step and establish global feature correspondences by adding only information from local triplet to triplet correspondences. The in- and outlier classification is done by checking the feature compatibility of triplet pairs. This gives an unbiased classifier for in- and outliers that does not depend on the succession of image insertions.

The main contributions of the proposed method are a strategy to identify the most reliable parts of unordered sets of images, build the reconstructions incrementally, using this information and add only locally verified correspondences. This creates a core structure of the input, i.e. reconstructions using the most reliable information. Because feature point tracks are verified only locally on image triplets, the loop closing problem is not biased by drift.

Recent literature in Structure from Motion comprises a number of approaches addressing the problem of reconstructing a scene from unordered collections of images [Snavely et al., 2006, Li et al., 2008, Martinec and Pajdla, 2007]. Snavely et al. describe in [Snavely et al., 2006] a system that is able to reconstruct a scene from a very diverse set of images, like image collections gathered from the web. A limiting factor regarding the scalability of this approach is mainly pair-wise image matching and large scale bundle adjustment. In [Snavely et al., 2008b] the latter problem is addressed by computing a small but representative *skeletal* set of a scene. Targeting at efficiency, Ni et al. propose in [Kai Ni Steedly, 2007] an out-of-core bundle adjustment capable of tackling larger reconstructions. Closely related is the approach described in [Li et al., 2008] where object recognition techniques are utilized to compute a small subset of *iconic images* that represent the important aspects from a scene. Again the algorithm is designed to work on large-scale image collections gathered from the Internet. Common for all the approaches is to rely on some calibration information, often directly derived from EXIF information. In our approach, we also utilize calibrated cameras with known focal length.

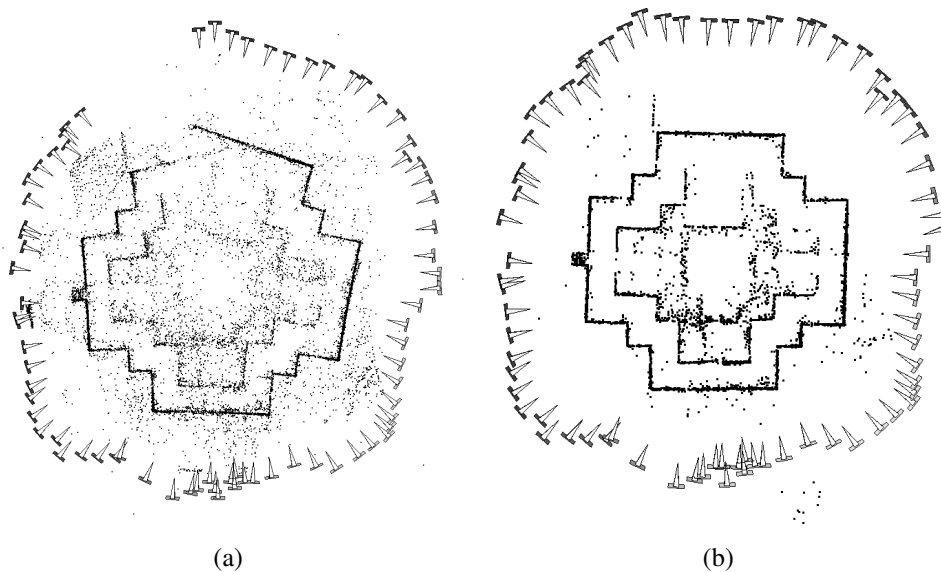


Figure 3.1: *Incremental reconstruction with and without drift.* This example shows the output of an incremental reconstruction pipeline where new views and feature point tracks are registered using the global 3D structure. Due to slight errors in (a) the camera calibration drift builds up in this reconstruction. If additional views are added at the point where the structure breaks up, the new views are registered only at one end of the loop and do not add additional constraints to the true global topology. Figure (b) shows the correct topology.

Considering view triplets as the basic SfM building block was addressed by many authors for image sequences [Fitzgibbon and Zisserman, 1998] and unordered image collections [Zach et al., 2008]. The triple relation based on the trifocal tensor imposes stronger geometric constraints and allows the local detection of mismatches. For instance, Zach et al. apply in [Zach et al., 2008] a non-monotone Bayesian reasoning based on view triplets to detect incorrect two view geometries.

Our approach builds upon recent advances in image retrieval, utilizing vocabulary tree data structures and inverted files [Nister and Stewenius, 2006] to speedup matching. Schindler et al. have shown [Schindler et al., 2007a] that this approach is also suitable for large scale location recognition. In our system a vocabulary tree structure is used for coarse image matching and is therefore related to [Irschara et al., 2007]. Recently, vocabulary based indexing structures were also successfully applied for Simultaneous Localization and Mapping (SLAM) tasks as described in [Eade and Drummond, 2008].

3.2 Structure and Motion Computation

Our algorithm consists of three major steps. (i) An epipolar graph $G_{\mathcal{E}}$ is created with images as nodes and correspondences verified by epipolar geometry as edges. The feature matching process is accelerated with a bag of words approach. (ii) This graph is then transformed into a graph $G_{\mathcal{T}}$ of triplet reconstructions. The nodes in this graph are all trifocal reconstructions created from $G_{\mathcal{E}}$ and are connected by overlapping views. These connections, i.e. edges, of $G_{\mathcal{T}}$ are created when triplets share at least one view and pass a test for 3D point compatibility. The feature correspondences of triplets are established by using tracks from the overlapping views. (iii) These edges of $G_{\mathcal{T}}$ are then merged incrementally into reconstructions, while loop closing is handled implicitly. Because the process is incremental, additional sets of unordered images can be easily added to extend created reconstructions after a set of input images has been processed.

3.2.1 Epipolar Graph $G_{\mathcal{E}}$

Given a set of unordered input images, SIFT features [Lowe, 2004] are extracted from every image. We use a vocabulary tree [Nister and Stewenius, 2006] based approach for coarse matching of similar images. Hence we can greatly reduce the computational effort of pair-wise image matching, by only matching the most relevant images as reported by the vocabulary scoring.

In our system the vocabulary tree is trained in an unsupervised manner with a subset of 2.000.000 SIFT feature vectors randomly taken from 2500 images. Thus our vocabulary is generic and allows the generalization to different data sets. The descriptor vectors are hierarchically quantized into clusters using a k-means algorithm. As proposed in [Nister and Stewenius, 2006], we set the branch factor to 10 and allow up to 7 tree levels. The image retrieval performance can be increased by using a higher branching factor [Schindler et al., 2007a]. Once the vocabulary tree is trained, searching the visual vocabulary is very efficient and new images can be inserted on-the-fly.

In our current setting we rely on an entropy weighted scoring similar to the *tf-idf* “term frequency inverse document frequency” as described in [Sivic and Zisserman, 2003]. Let \mathcal{D} be an image in our database and t be the term in the vocabulary associated to feature f of the current query image \mathcal{Q} , then our scoring function $sim(\mathcal{Q}, \mathcal{D})$ is,

$$sim(\mathcal{Q}, \mathcal{D}) = \frac{1}{|\mathcal{Q}| + |\mathcal{D}|} \sum_{t \in \mathcal{Q} \cap \mathcal{D}} \log \left(\frac{N}{n(t)} \right) \quad (3.3)$$

where N is the total number of images in the collection, $n(t)$ is the number of images that contain term t and $|\mathcal{Q}|$, $|\mathcal{D}|$ are the number of features from the query and database im-

age, respectively. This weighting allows fairness between database images with different number of features.

The tentative sparse image correspondences retrieved from the vocabulary tree are then matched using an approximated nearest neighbor technique. The epipolar geometry is computed using a five-point [Nistér, 2004] minimal solution inside a RANSAC loop. The correspondence inlier set is used to build the epipolar graph $G_{\mathcal{E}}$ of image connections. The nodes are the images and the edges the inlier set of the pairwise epipolar geometry.

3.2.2 Trifocal Graph $G_{\mathcal{T}}$

By using three images as the basic geometric entity, the number of false correspondences can be reduced for points that are visible in all three views and a more reliable basic graph representation of the image connections can be established. Image triplets are the nodes of this graph and are created from the epipolar geometries. These nodes are basically connected by overlapping views. Degenerate configurations can also be present in the trifocal case, but these configurations are usually geometrically incompatible with other triplets and are at most present at the fringe of the graph.

3.2.2.1 Trifocal Reconstructions

In the next step we create all potential trifocal reconstructions from the edge information in the epipolar graph $G_{\mathcal{E}}$. A Minimum Spanning Tree (MST) of $G_{\mathcal{E}}$ is created in a similar way as in [Steele and Egbert, 2006] but is only used to enumerate potential image triplets efficiently. The MST is traversed and all possible triplet candidates are generated from this list. Then the edges of the MST are removed from $G_{\mathcal{E}}$ and the next MST is generated. Figure 3.2 shows how triplets are enumerated from one MST. This process is iterated until all edges of $G_{\mathcal{E}}$ are processed. The advantage of the MST creation over a brute force triplet enumeration is that it can be stopped after a few iterations and uses the best globally connected matching epipolar geometries first.

Triplet Reconstruction and Reconstruction Quality: We reconstruct the trifocal structure if the three images are fully connected by three epipolar edges. Two connections suffice if all three views share at least one point but we require that all three connections are present. For all three pairwise relative orientations obtained with a minimal solver [Nistér, 2004] the third view is inserted with a three point calibrated absolute pose solver [Horn et al., 1988] inside a RANSAC loop. The configuration with the highest inlier count is selected and optimized with bundle adjustment [Triggs et al., 2000].

A third view of 3D structure only increases the discriminability of false three view feature point matches if it adds additional information, i.e. reduces the covariance of the triangulated scene point. We use [Beder and Steffen, 2006] to compute the median

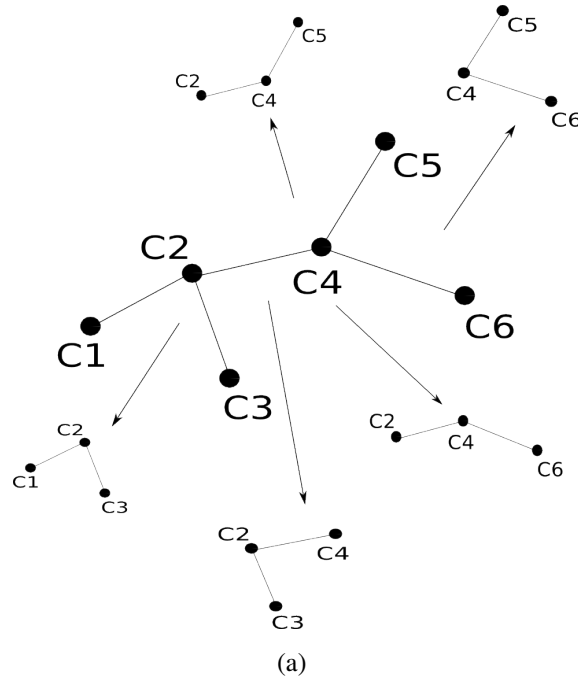


Figure 3.2: *Triplet enumeration.* This example shows how a MST is used to enumerate image triplets. The MST consists of 6 cameras $C1\dots C6$ and consists of 5 triplet candidates.

roundness of the 3D structure uncertainty that is visible in all three views for all three image pairs and reject a triplet if at least one epipolar combination does not provide a certain baseline. Therefore, we enforce a minimal triangulation angle between all pairs of images, and avoid degenerated triple configurations (e.g. epipolar relations from pure rotational motions).

Each accepted trifocal reconstruction is inserted as node into $G_{\mathcal{T}}$. In the next step, connections between these nodes will be created.

3.2.2.2 Trifocal Graph Edges

Overlap between the set of trifocal reconstructions has to be established. We distinguish between the detection of potential overlap of two trifocal reconstructions (connectivity) and the geometric consistency of two trifocal reconstructions (compatibility).

[Fitzgibbon and Zisserman, 1998], two image triplets can share zero, one or two images. To simplify matching we only consider triplets that have at least one view in common for a potential edge in $G_{\mathcal{T}}$. Correspondences between the two 3D point sets are established with the common images. The correspondence information from common views does not take all possible structure matches into account, but different combinations of triplets will usually contain this information.

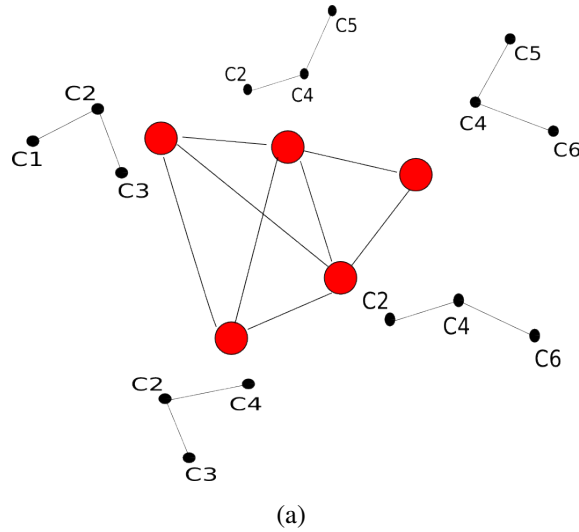


Figure 3.3: *Triplet Graph.* Each node in this graph represents a reconstructed image triplet. Common views of the reconstructed image triplets are used to determine the connectivity in this trifocal graph $G_{\mathcal{T}}$.

Compatibility: Two trifocal reconstructions that are potentially connected are registered into one coordinate system by computing the similarity transform of the two point sets in a RANSAC loop and evaluating the reprojection errors of the transformed points. The two aligned triplets are then used to create all possible local tracks and contradictory measurements are removed. For two triplets with two overlapping views this means local tracks of length two, three and four are created.

This local correspondence information is inserted into $G_{\mathcal{T}}$ as edge. Figure 3.3 shows how overlapping views induce the graph structure. The edge information will be used in the next step to create a global representation of the scene.

3.2.3 Reconstruction

The main idea of our merging process of image triplets into global reconstructions is that we start from the most reliable parts simultaneously and merge only the “local” edge information of $G_{\mathcal{T}}$ into reconstructions in an incremental way and handle loop closing implicitly. In contrast to methods that compute all camera positions and 3D points in one step, new images can be easily added to an existing reconstruction and the scalability is only bound by bundle adjustment and therefore numerically stable. Various methods have been proposed to speed up the bundle adjustment optimization, for example, either by reducing the number of iterations and varying views [Mouragnon et al., 2006a, C. Engels

and Nister, 2006] or by repartitioning the problem [Shum et al., 1999, Steedly et al., 2003, Kai Ni Steedly, 2007].

3.2.3.1 Identifying Most Reliable Image Triplets

We begin with a full graph $G_{\mathcal{I}}$ of triplet reconstructions and corresponding geometrically verified edges. With an unordered set of images no meta information about topology like in [Fitzgibbon and Zisserman, 1998] is given. One obvious reconstruction strategy would be to use a MST or skeletal graph (the skeletal graph also reduces the data set, a topic we are not dealing with in this work) of triplets or views connected by triplets, similarly to the epipolar equivalents of [Steele and Egbert, 2006] or [Snavely et al., 2008b].

We propose to start with the most reliable edges between nodes and start the reconstruction from these points. We select the node N_C that has the highest cardinality of connected triplets, i.e. the vertex with the highest degree and search for a second node N_{CC} that is connected to N_C . This edge will be inserted into the reconstructions. The vertex N_{CC} is the one with the highest degree of the set of adjacent nodes of N_C . This gives us an edge that represents the vertex N_C with the highest cardinality of connected triplets in the graph and the best (highest cardinality) adjacent node N_{CC} . An edge is selected with this strategy, merged into a reconstruction and removed from $G_{\mathcal{I}}$. This is repeated until $G_{\mathcal{I}}$ is empty.

The advantage of this strategy is twofold. Firstly, all edges in $G_{\mathcal{I}}$ are to some degree compatible and the node with the highest cardinality is therefore a highly reliable starting point and is located in a neighborhood with high information redundancy. The same observation is valid for N_{CC} . This is a simple measure of reliability that incorporates more information/views than searching simply for the best edge weight based on feature correspondences between two geometric primitives. We focus on the “easy” parts of the problem first. Errors introduced due to mismatches are usually introduced at a late stage in reconstruction and do not affect the robust cost functions during bundle adjustment. Secondly, this strategy reduces the number of reconstruction merging operations because the probability of selecting an edge with a node already present in a reconstruction is maximised. This is of course only valid for edge selection strategies that do not create an insertion ordering by spanning the graph.

3.2.3.2 Integrating Local Correspondence Information

The edges selected by the cardinality strategy have to be merged into potentially existing reconstructions. Four different update strategies exist: (i) If both nodes are not present in a reconstruction, a new one is created. (ii) If one triplet of the edge e_{ij} , connecting triplet i and j , is present in a reconstruction new correspondences and potentially new cameras

have to be added. Adding new correspondences of existing cameras is straightforward. The correspondences from e_{ij} are already local inliers and are simply added to existing tracks or triangulated using the cameras from the reconstruction if the track is new. If new cameras have to be added, the reconstructions can be transformed into the global system by using the similarity transform T_r from the local edge to edge registration and a similarity transform T_g transforming the triplet that is already in the reconstruction into the global one. No outlier rejection is done using the structure or cameras of the global reconstructions. (iii) If both triplets of the edge e_{ij} are already present in the same reconstruction, only new correspondences have to be added. (iv) If both triplets of the edge e_{ij} are already present in different reconstructions, R_k and R_l , these reconstructions should be merged.

Merging Reconstructions: The reconstructions R_k and R_l should be merged if an edge e_{ij} connecting them is inserted. A similarity transform is computed using RANSAC and the reprojection errors of overlapping cameras and the involved 3D points. The reconstruction with fewer cameras is transformed into the coordinate system of the larger reconstruction. New cameras and tracks of the smaller reconstruction are added in a similar way as in the triplet merging process.

Adding New Images: Because the reconstruction is built incrementally, it is possible to add new sets of images. The new images can be matched with the old and new ones and the epipolar graph is extended. The new triplet edges are added to $G_{\mathcal{I}}$ and merged into the existing reconstructions.

Implementation Aspects: Bundle adjustment is used to integrate the correspondences and cameras. Given the reconstruction problem $\mathbf{x}_j^i = P^i \mathbf{X}_j$ where the 2D point measurements \mathbf{x}_j^i are the observations of unknown 3D points \mathbf{X}_j observed in the unknown cameras P^i , bundle adjustment is defined as the (in practice local) minimum of the cost function $\mathcal{C}(P^i, \mathbf{X}_j) = \sum_i \sum_j v_{ij} d(P^i \mathbf{X}_j, \mathbf{x}_j^i)^2$, where v_{ij} is a binary variable that is 1 if the point \mathbf{X}_j is visible in image P^i and 0 otherwise. We replace $d(P^i \mathbf{X}_j, \mathbf{x}_j^i)^2$ with two robust cost functions. These two types of bundle adjustment iterations are: Huber cost function $\gamma_h(e)$ iterations and saturated error $\gamma_s(e)$ (also called Blake-Zisserman cost function) iterations. The two cost functions with inlier threshold b are:

$$\begin{aligned} \gamma_h(e) &= \begin{cases} e^2 & \text{if } e < b \text{ inlier} \\ 2b|e| - b^2 & \text{if } e \geq b \text{ outlier} \end{cases}, \\ \gamma_s(e) &= \begin{cases} e^2 & \text{if } e < b \text{ inlier} \\ b & \text{if } e \geq b \text{ outlier} \end{cases} \end{aligned} \quad (3.4)$$

$\gamma_h(e)$ bundle adjustment iterations are used after a new camera is added. This cost function is robust to outliers and still establishes the global topology when loops are

closed. $\gamma_s(e)$ iterations are only used after at least 10 cameras have been added and strong outliers (we set $b = 5$ for $\gamma_h(e)$ and $b = 25$ for $\gamma_s(e)$) are then removed. This removes mainly contradictory feature tracks (usually two concatenated tracks that have length two in each triplet) and prevents error build up that can negatively affect the Huber cost function.

3.3 Results

Data Sets: We have tested our method with several large image collections. Table 3.1 summarizes basic properties of three of those collections. The data sets Opera and SoL contain mostly images from the one scene, the data set Cathedral contains a large number of images from different in- and outdoor locations. Table 3.2 summarizes timing results and the number of triangulated points for some data sets.

Reconstructions: Table 3.3 presents some results obtained with our method and shows a comparison with the publicly available bundler software ¹. The topological properties of the data sets are reflected in the visualizations of the trifocal and epipolar graphs for the data sets. Nearly all images of the Opera data set are connected and the topological structure is present in the graph representations. The diverse nature of the images of the Cathedral data set are also visible in the corresponding trifocal graph $G_{\mathcal{T}}$. The last column, City Block, shows an experiment around a closed loop in a city. The image data set is weakly linked at some points. Our method obtains two reconstructed blocks for the data set. The first graph of the City Block column shows the epipolar graph $G_{\mathcal{E}}$ of the complete data set. The topological structure of the loop is correct but not all epipolar geometries are suitable for reconstruction. The second graph of the City Block column shows the trifocal graph $G_{\mathcal{T}}$. The data is split into two parts that can be reconstructed reliably.

The comparison with the bundler software shows that [Snaveley et al., 2006] has difficulties with data sets that have low redundancy. These kind of data sets are prone to build up of drift. In the Opera data set drift builds up and bundler is not able to close the loop. The Cathedral data set works also well with the bundler approach because the images of the cathedral are very well textured. In the City Block experiment, bundler follows the epipolar links around the graph and reconstruction fails at the geometrically weak points at both ends of the reconstruction.

Unordered community photo collections of famous locations are usually densely sampled and consist of many redundant views. The image data sets we use here were obtained at least to some degree with reconstruction in mind and are more of a sequential nature

¹ bundler 0.3 <http://phototour.cs.washington.edu/bundler/>

| Data Set | Images | Triples | LCCC |
|-----------|--------|---------|------|
| SoL | 248 | 1724 | 139 |
| Opera | 347 | 1951 | 304 |
| Cathedral | 1920 | 7395 | 564 |

Table 3.1: *Overview of data sets.* This Table shows the number of input images, reconstructed triples and the number of views in the largest connected component (LCCC).

| Data Set | $G_{\mathcal{E}}$ | $G_{\mathcal{T}}$ | LCCP2 | Merge |
|-----------|-------------------|-------------------|--------|--------|
| SoL | 187 min | 13 min | 23210 | 5 hrs |
| Opera | 8 hrs | 30 min | 116961 | 7 hrs |
| Cathedral | 50 hrs | 2 hrs | 321000 | 40 hrs |

Table 3.2: *Performance overview.* This Table shows timing results for the epipolar and trifocal graph creation, the number of triangulated points visible in at least two views of the largest connected component (LCCP2) and the timing results of the complete edge merging/reconstruction steps (Merge). The timing results were obtained on an Intel Pentium D CPU with 3.00 GHz.

because we want to sample larger parts of a city for image-based localization and therefore want to take as few images as possible. This makes the reconstruction process more challenging and prone to drift.

Surviving Repetitive Structure We test the proposed method on a small data set containing about 70 images. This data set contains images where large parts are covered by repetitive content. This is an often neglected but omnipresent problem in real image data sets. The cardinality based search for the most reliable parts of the data set seems to identify these parts well and the ambiguous edge information is added at a later step. Because the edge selection strategy distributes new information over the entire reconstruction using the most reliable parts first, a correct core structure is established and the high amount of outliers, mainly introduced at later steps, does not influence the topology of the result. Figure 3.5 presents sample images and qualitative results.

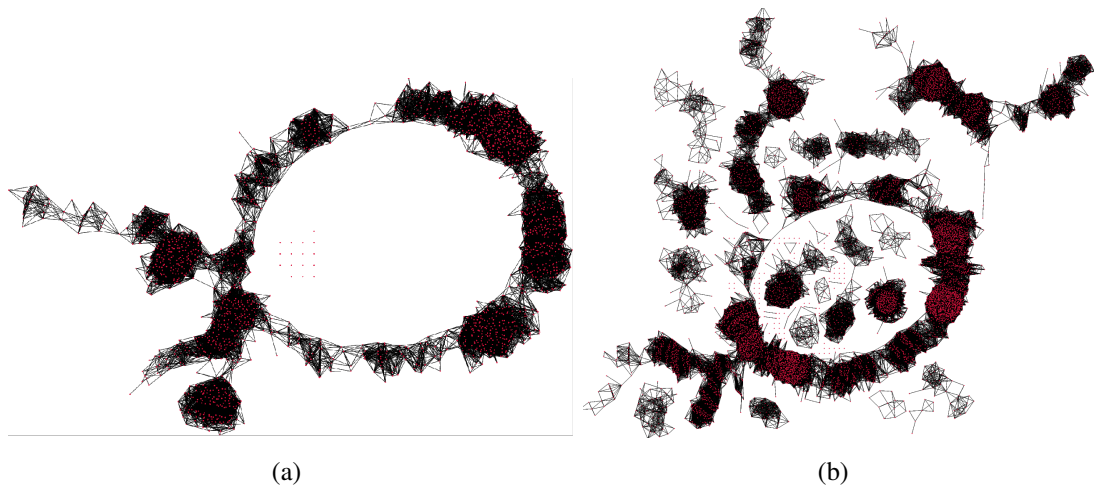


Figure 3.4: Visualization of the trifocal graphs $G_{\mathcal{T}}$. (a) Opera data set. The red points represent 1951 triplet reconstructions that are connected by overlapping views (33000 edges) and have compatible 3D point structure. (b) Cathedral data set. The red points represent 7400 triplet reconstructions that are connected by overlapping views (180000 edges) and have compatible 3D point structure. This data set contains images from different parts of a city and include indoor and outdoor images.

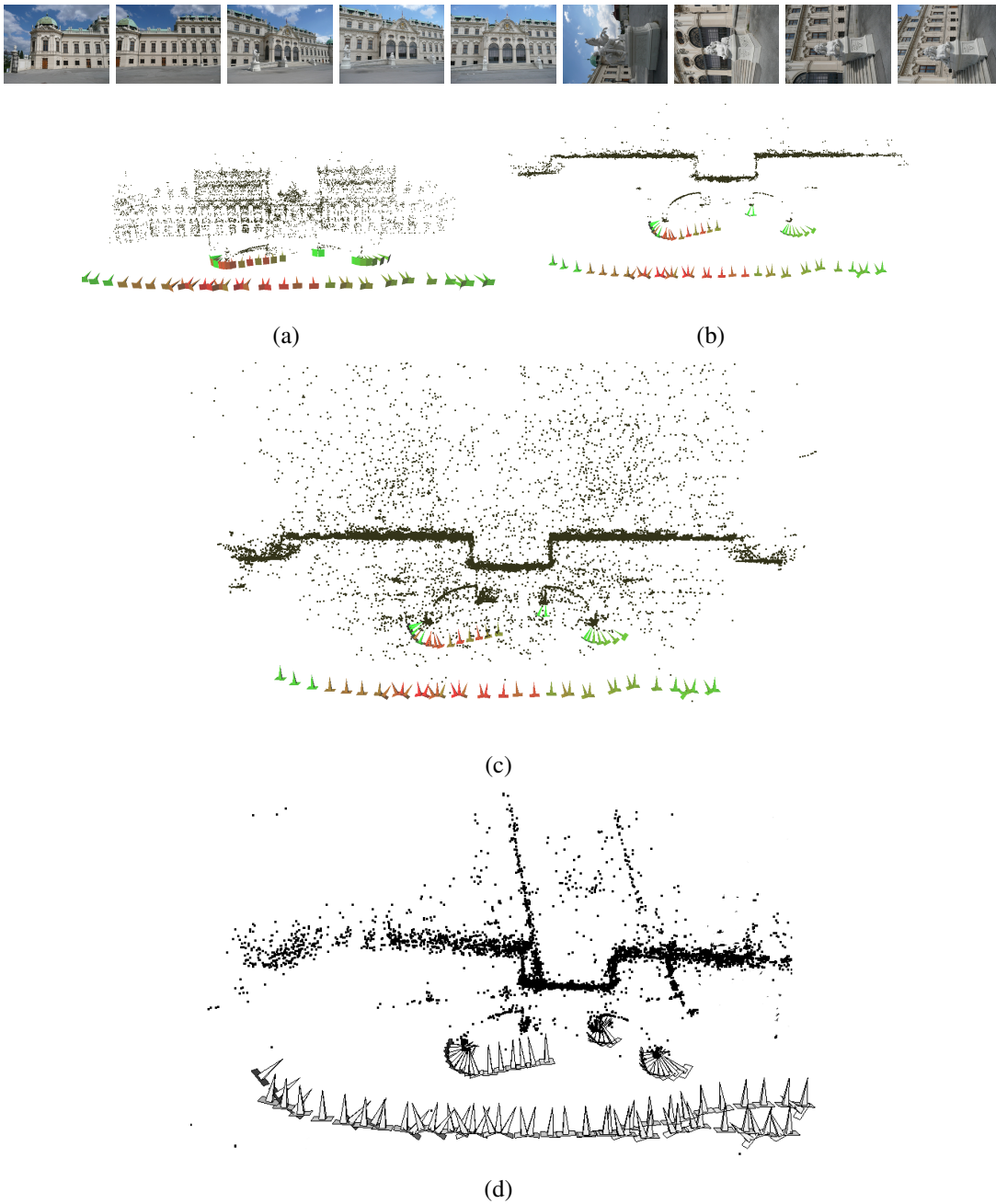
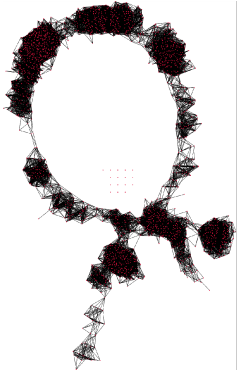
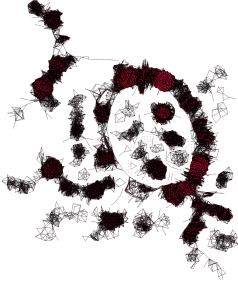

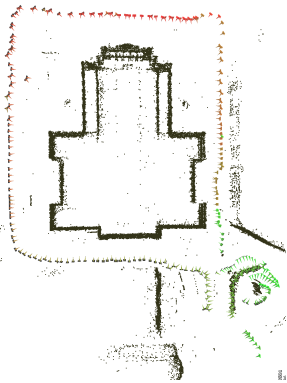
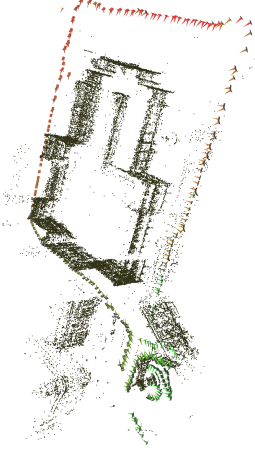
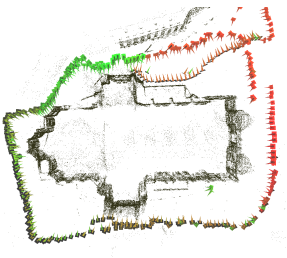
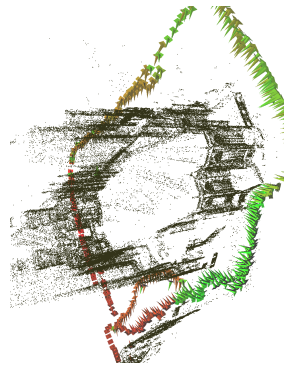

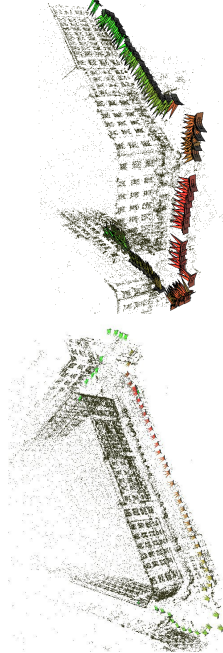


Figure 3.5: *Repetitive structure.* The top row shows some of 70 images from a scene where strong repetitive structures are present. Figure (a) and (b) shows the visualization of the resulting reconstruction and displays 7400 3D points that are visible in at least four views. Figure (c) shows the same reconstruction with all 24000 tracks, revealing a high amount of outliers. Figure (d) shows the same scene reconstructed by simply adding views to global structure, note that there is no clean distinction between in- and outliers (similar robust cost functions have been used during bundle adjustment).

| | | | |
|-------------------------------|--|--|--|
| <p>Graphs</p> | <p>Opera</p>  | <p>Cathedral</p>  | <p>City Block</p>  |
| <p>Triplet Reconstruction</p> |   |   |   |

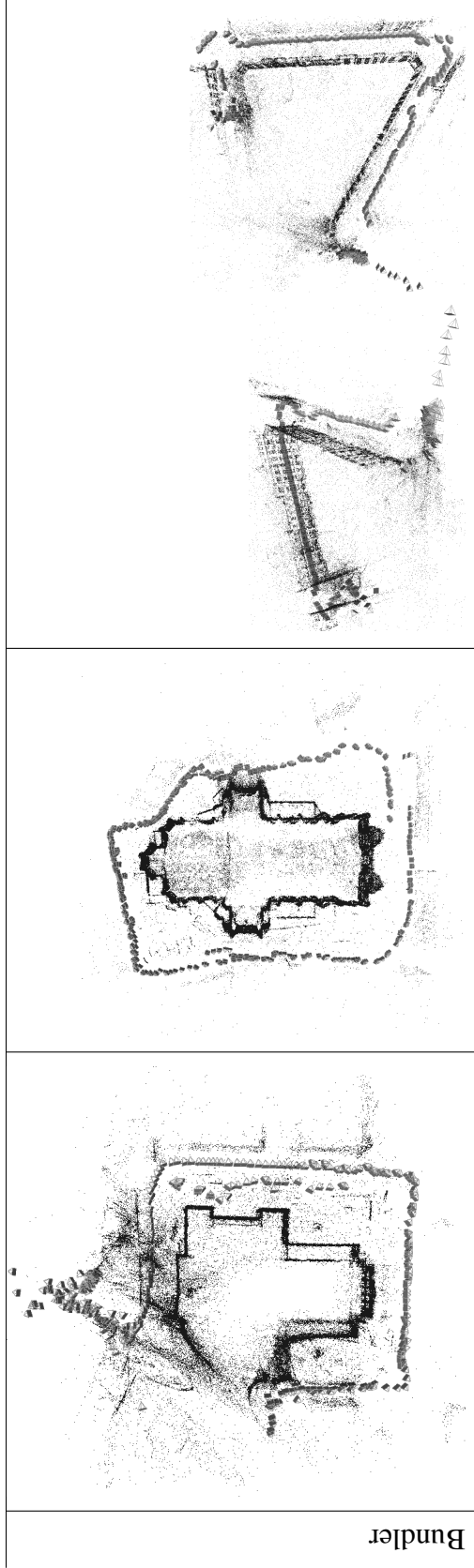


Table 3.3: Results and comparison with *bundler* software. The first and second column present results for the Opera and Cathedral data sets. The third column City Block shows two results of the same data set and a comparison of the epipolar and trifocal graph topology. *Opera*: The trifocal graph shows that nearly all images of the Opera data set are connected and the topological structure is present in the graph representation. Bundler fails to close the loop on this data set. *Cathedral*: The diverse nature of the images of the Cathedral data set are also visible in the corresponding trifocal graph. Bundler succeeds on this data set. *City Block*: The first graph of City Block shows the epipolar graph $G_{\mathcal{E}}$ and the second graph the corresponding trifocal graph $G_{\mathcal{F}}$. Note that the epipolar graphs shows the correct topological structure of the data set, but some images are weakly linked for reconstruction. This can be seen in the topology of the trifocal graph. The loop is split into two parts and our reconstruction method obtains two major blocks of the data set. We show two views of the result obtained with bundler, the reconstruction diverged at both ends of the structure.

3.4 Conclusion and Discussion

We have introduced a novel method for building the core structure of image based reconstructions from unordered sets of input images incrementally and robustly. The main contributions of the proposed method are (i) the implicit loop closing while building the reconstructions incrementally by using only correspondence information from locally matched image triplets and (ii) a strategy to identify and start with reliable subsets of images and their corresponding geometric primitives that have the highest mutual compatibility consensus. Experimental results show that the algorithm can build large reconstructions incrementally without depending on global structure to separate in- and outliers. Furthermore, experimental results indicate that this strategy provides superior results when repetitive structure and a high amount of feature matching outliers (even in image triplets) are present.

Chapter 4

Feature Matching for Sequential Data

Image data with sequential parts does not only have drawbacks for SfM of course. When the time-line of the captured images is known, additional information is available. This can be used for example to add constraints over time for feature matching. We propose use this information to detect overlaps in image sequences, and use this information to integrate overlapping sparse 3D structure from video sequences. The additional temporal information of these images is used to increase robustness over single image pair matching. A scanline optimization problem formulation is used to compute the best sequence alignment using wide-baseline image matching techniques. Compared to a direct dynamic programming approach, the scanline optimization formulation increases the robustness of sequence alignment for general relative motions. The proposed alignment method is employed to integrate sparse 3D models reconstructed from separate sequences. This is especially applicable when the image sources are video streams. In addition loop closures can be detected more robustly than compared to two-view image matching based techniques.

4.1 Introduction

We examine how to extend the concept of image matching to the matching of consecutive video sequences and apply this matching technique in a 3D vision system. We want to combine the advantages of sequential data with the advantages of global reconstruction optimization. The goal is to integrate partial reconstructions into a larger database so that the user can be informed about the quality of the reconstruction and missing areas and add additional data selectively.

Many reconstruction systems in computer vision are based on images from a moving video camera. These video based systems can use uncalibrated [Pollefeys et al., 2004] or calibrated cameras [Mouragnon et al., 2006a] [Mouragnon et al., 2006b] [Nistér et al.,

2004] and are applied for example to cultural heritage modeling, odometry, robot navigation and city modeling. Multi-camera heads can be used to extend the field of view [Akbarzadeh et. al, 2006]. These methods are particularly appropriate to create large sparse reconstructions of continuous movements in real time.

Another source of images with additional sequential information are vision based Simultaneous Localization and Mapping (SLAM) methods. Incremental map building and continuous localization increase the robustness of SfM as demonstrated in [Davison et al., 2007] [Davison, 2003] [Eade and Drummond, 2006]. The area that can be covered is limited by the number of landmarks that can be recognized and optimized efficiently. Furthermore, classical SLAM algorithms use every single image from the video stream for the tracking and mapping operations. To handle these amounts of data, bundle adjustment is replaced by simpler methods to integrate the mapped landmarks. An exception is [Klein and Murray, 2007], where tracking and mapping is split into separate tasks, but this SLAM approach is explicitly designed for small workspaces.

A logical extension of visual odometry based reconstructions is the integration of multiple sequences. We propose a method to integrate multiple sparse reconstructions from a visual odometry front end into a global coordinate system. Loops are a special case of overlaps of sequences and structure that provide a way to reduce drift from an odometry trajectory. An example of a system where loops are detected in a sparse reconstruction to reduce this drift can be found in [Verbiest and Gool, 2004]. In [Ho and Newman, 2007] the additional sequential information of image sequences is used for loop closing and therefore drift reduction in robot navigation.

Sequential information is either available by known time-stamps and a sequential capturing strategy or when the data source is a video stream. In both cases have these images in common that they have been obtained consecutively and contain additional sequential information.

4.2 Image-based Sequence Similarity

This section describes how a first matching of individual images of two sequences is done efficiently. The matching score of image sequences uses a similarity score of individual image pairs. The image pair similarity score is feature based. A vocabulary tree can be used to avoid the matching of all image pairs. The extracted features are further used to fuse the sparse reconstructions of the sequences.



Figure 4.1: Each row shows an image sequence. Only a subset of key-frames obtained from the SfM input video is shown.

4.2.1 Image Matching

The cost of matching all image pairs of two sequences would severely limit the possible sequence lengths. Retrieving similar images for a given one is currently a very active research topic e.g. [Schaffalitzky and Zisserman, 2002, Nister and Stewenius, 2006, Jegou et al., 2007]. To speed up the pairwise matching we employ a visual vocabulary tree approach similar to [Nister and Stewenius, 2006]. The vocabulary tree enables us to efficiently match a single image against all images in the sequence.

4.2.2 Visual Similarity Matrix

Given two image sequences s_1 and s_2 , the image features of s_1 are inserted into an empty vocabulary tree to create the inverted file structure. For each image in s_2 a query with the the vocabulary tree and the scores of the k best matching images are returned. The obtained matching scores are used to construct a Visual Similarity Matrix (VSM). Figure 4.2 shows the VSM obtained from the two sequences of Figure 4.1. Each element $e_{i,j}$ of the VSM corresponds to an image similarity between image i of the first sequence s_1 and image j of the second image sequence s_2 . Each row of the VSM has at most k non-zero entries.

Our experiments have shown that it is sufficient to use the image similarity scores from the vocabulary tree directly to construct a VSM. No further image to image feature matching or geometric verification is done to enhance the scoring accuracy at this stage.

4.3 Sequence Alignment

After computing the VSM, a contiguous path of corresponding images in this matrix can be extracted to represent the video sequence overlap. This section describes our approach to solve this problem. An optimal local sequence alignment can be computed in principle using the well known Smith-Waterman [Smith and Waterman, 1981] algorithm. This dynamic programming algorithm is used for example in bioinformatics to align protein

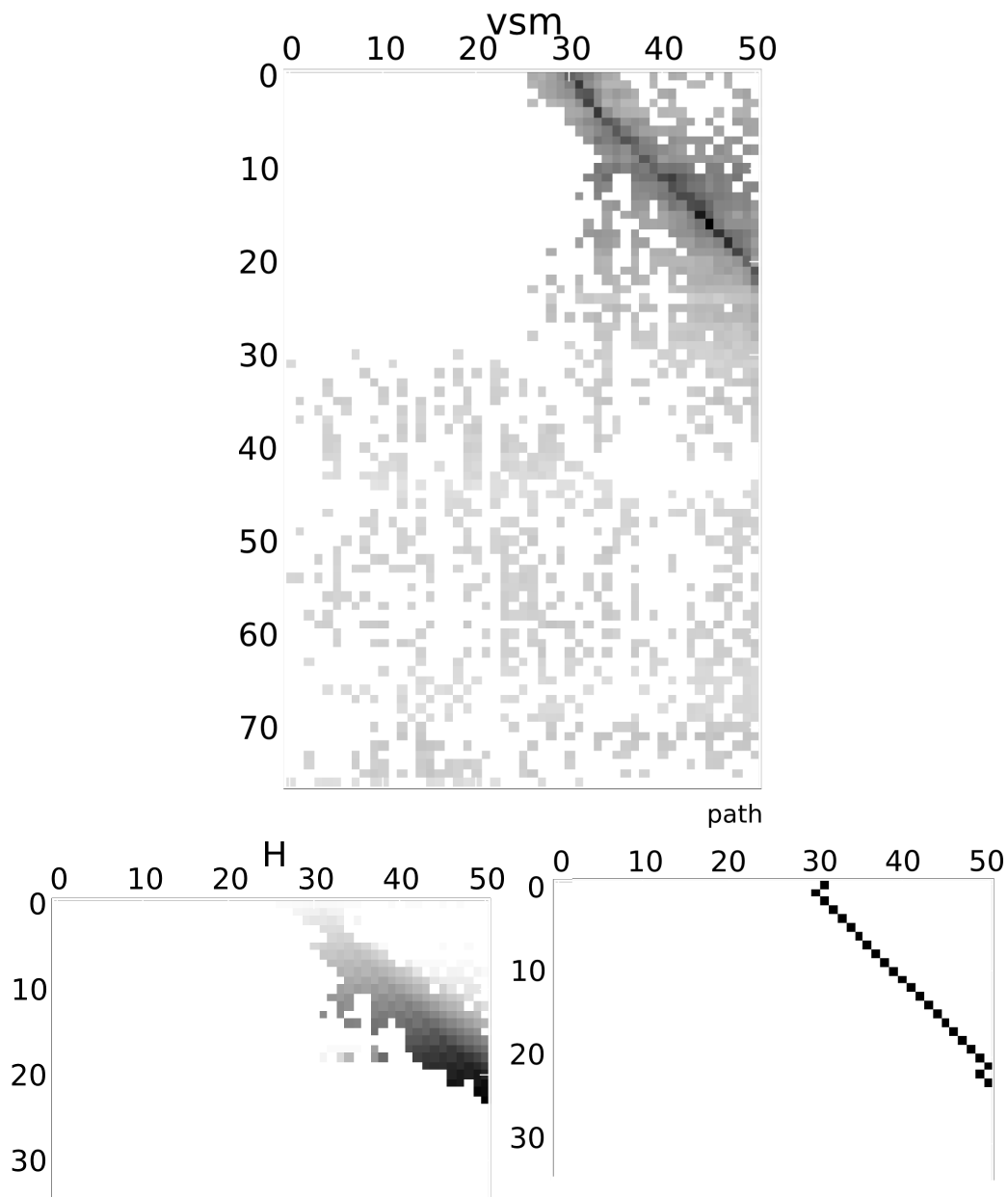


Figure 4.2: VSM matrix of the two sequences from Figure 4.1 and the resulting dynamic programming matrix H and the extracted correspondence path for the two sequences.

or nucleotide sequences. *Local* sequence alignment includes the ability to detect and match only subsequences of the input and to ignore non-corresponding sections. *Global* sequence alignment can be achieved by a slightly simpler variant of the Smith-Waterman algorithm, the Needleman-Wunsch [Needleman and Wunsch, 1970] method.

A limitation of this approach for image sequence matching is that only sequence over-

laps in the forward direction of relative movement can be obtained. This is due to the ordering constraint inherent in all classical dynamic programming approaches. This means that image sequence matching has to be done two times for a sequence pair if the relative sequence movement is not known a priori. In more complex cases, where the relative movement direction of a sequence pair changes multiple times, the Smith-Waterman algorithm can only find sequence parts with consistent relative movement. Hence, the full overlapping sequence is unnecessarily split into several subsequences, which need to be merged in a post-processing step. We propose scanline optimization for local sequence alignment to find longer matching sequences, and therefore to avoid any later post-processing step.

We propose a variation of scanline optimization [Scharstein and Szeliski, 2002] to compute the best sequence alignment. In general, scanline optimization is a dynamic programming approach to determine the maximum a posteriori solution of 1-D Markov random fields. Most prominently, it is used in several methods for dense depth estimation from stereo images [Scharstein and Szeliski, 2002, Hirschmüller, 2005]. In contrast to earlier Dynamic Programming (DP) approaches for stereo, scanline optimization does not enforce the ordering constraint. In our application, this feature enables sequence alignment for more general motions, which we consider the main advantage of our method compared with DP-based ones. A slight drawback of scanline optimization is the non-commutativity, i.e. the returned path for swapped inputs is not just the transpose of the original path.

4.3.1 Scanline Optimization Problem Formulation

Scanline optimization computes the optimal assignment of a sequence of images x_i to corresponding images of another sequence d_x . It finds the value of

$$\text{score}(x, d) = \arg \min_{d_x} \sum_{i=1}^N \left(D(x_i, d_x) + \lambda V(d_x, d_{x-1}) \right), \quad (4.1)$$

where $D(x, d) = -S(x, d)$ is the dissimilarity score of two images at positions x and $x + d$, and $V(d, d')$ is the regularisation cost and λ weights the relative influence of these two factors.

In order to obtain similar results in cases of pure forward/backward motion, we model the regularisation to approximate the moves favored by the Smith-Waterman algorithm. The smallest regularisation cost, zero, is assigned to diagonal moves, i.e. if $|d_{x-1} - d_x| = 1$. The cost of any occlusions in the image sequence (i.e. skipping images) is equal to the number of skipped frames. E.g., moving in one image, but not in the other ($d_{x-1} = d_x$) has cost one. More formally, the regularization cost for two successive image assignments

d_{x-1} and d_x is given by:

$$V(d_x, d_{x-1}) = \begin{cases} i-1 & \text{if } d_x = d_{x-1} - i, i = 2, 3, \dots \\ 0 & \text{if } d_x = d_{x-1} \pm 1, \\ 1 & \text{if } d_x = d_{x-1}, \\ i-1 & \text{if } d_x = d_{x-1} + i, i = 2, 3, \dots \end{cases}$$

4.3.2 Efficient Minimization

Minimizing Eq. 4.1 and determining the corresponding optimal assignment d_x can be efficiently performed using a dynamic programming approach by maintaining the minimal accumulated costs $H(x, d)$ up to the current position in the first image sequence x :

$$H(x, d) = D(x, d) + \min_{d'} (H(x-1, d') + \lambda V(d, d')).$$

We have the initial values $H(1, d) = D(1, d)$. Note, that our specific choice of $V(\cdot, \cdot)$ can be written as

$$V(d, d') = \min(|d+1-d'|, |d-1-d'|), \quad (4.2)$$

i.e. it is the minimum of two linear discontinuity cost functions. Consequently,

$$\begin{aligned} \min_{d'} (H(x-1, d') + \lambda V(d, d')) = \\ \min \left\{ \min_{d'} (H(x-1, d') + \lambda |d+1-d'|), \right. \\ \left. \min_{d'} (H(x-1, d') + \lambda |d-1-d'|) \right\}. \end{aligned}$$

Following [Felzenszwalb and Huttenlocher, 2004], the simultaneous calculation of the sub expressions

$$\min_{d'} (H(x, d') + \lambda |d+1-d'|) \quad (4.3)$$

and

$$\min_{d'} (H(x, d') + \lambda |d-1-d'|) \quad (4.4)$$

for every d can be performed in linear time using a forward and a backward pass to compute the lower envelope. Hence, the proposed energy can be minimized in $O(nm)$ time, where n and m are the lengths of the two sequences, respectively. A direct approach would have $O(nm^2)$ time complexity.

The procedure to fill the entries of H is summarized in Algorithm 4.1. The necessary instructions to maintain the backtracking table for fast subsequent alignment extraction

are omitted. This procedure is very similar to the scanline optimization method proposed for stereo, with two main distinctions: first, the discontinuity cost V has a different shape; second, clamping the accumulated cost to 0 indicates the potential termination of a locally aligned sequence. Note that in this application the accumulated costs are less or equal zero.

Figure 4.2 shows an example of the matrix H and the extracted sequence correspondence path.

Algorithm 4.1 Dynamic programming scanline optimization

Input: Dissimilarity scores $D_{n \times m} = -S_{n \times m}$
 $H \leftarrow 0_{n \times m}$
 $H[d, :] \leftarrow D[1, :]$
for $x = 2 : n$ **do**
 \triangleright h can be computed in $O(m)$ time using [Felzenszwalb and Huttenlocher, 2004].
 $\forall d : h[x, d] \leftarrow \min_{d'} (H[x-1, d'] + \lambda V(d, d'))$
 \triangleright Note: $H[x, d] = 0$ terminates the local alignment sequence.
 $\forall d : H[x, d] \leftarrow \min(0, D[x, d] + h[x, d])$
end for

4.3.3 Matching Multiple Sequences

We use an incremental approach to find the overlap of multiple image sequences. To compare multiple sequence matches, the optimal scanline assignment score, $\text{score}(x, d)$, is computed for all pairs.

A slow relative movement that covers only a small amount of structure overlap but contains many images produces a similar score as a larger movement with the same amount of wider placed images. We normalize the sequence matches to favour sequence matches that cover a wide range of structure over slow relative movements. This is done by scaling the matching score with the deviation from an ideal diagonal movement. This leads to the normalized score

$$\text{score}_n(x, d) = \text{score}(x, d) \frac{\text{width}(\text{path}, x) \text{width}(\text{path}, d)}{\|\text{path}\|^2},$$

where path is the sequence of image matches and $\text{width}(\text{path}, a)$ is the number of different images from the sequence a that is contained in the image correspondence path.

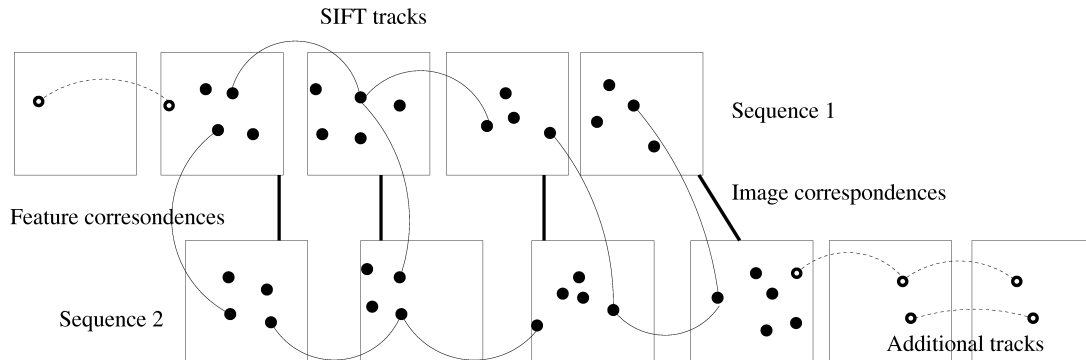


Figure 4.3: The different correspondences used for sequence merging. Image correspondences: These are the image matches from the scanline sequence matching. Feature correspondences: The matched inter-sequence correspondences that remain after the geometric verification. SIFT tracks: The extracted SIFT points are combined to tracks for each sequence. Additional tracks: The tracked corners from the original sequences.

4.4 Results

We demonstrate our algorithms on two data sets. The first experiment demonstrates the added image matching robustness under weakly textured scenes and the concept of merging of different sparse odometry reconstructions. Three image sequences were captured by hand with a digital compact camera. The Motion JPEG videos with 640×480 pixels resolution were used to compute three separate sparse reconstructions. Figure 4.4 shows the result of the sequence overlap extraction and geometry merging process. The second data set demonstrates the extraction of image correspondence paths that change their relative movement direction and the loop closing capabilities of the merging process. One Motion JPEG video with 840×480 pixels resolution was used in this experiment as odometry input. A reversed copy of the video was added to the stream before the initial reconstruction was obtained so that the begin and end of the camera trajectory are the same. Figure 4.5 shows the results.

4.5 Summary and Conclusions

To obtain the initial sparse reconstructions fast and robust feature point tracking and visual odometry can be used for video input or a still image-based SfM pipeline for time-stamp data. The initial sequence overlaps are computed with the key frames from the sparse reconstruction. Matching the whole image sequences using a scanline optimization problem formulation increases the robustness compared to single image pair matching. No 3D

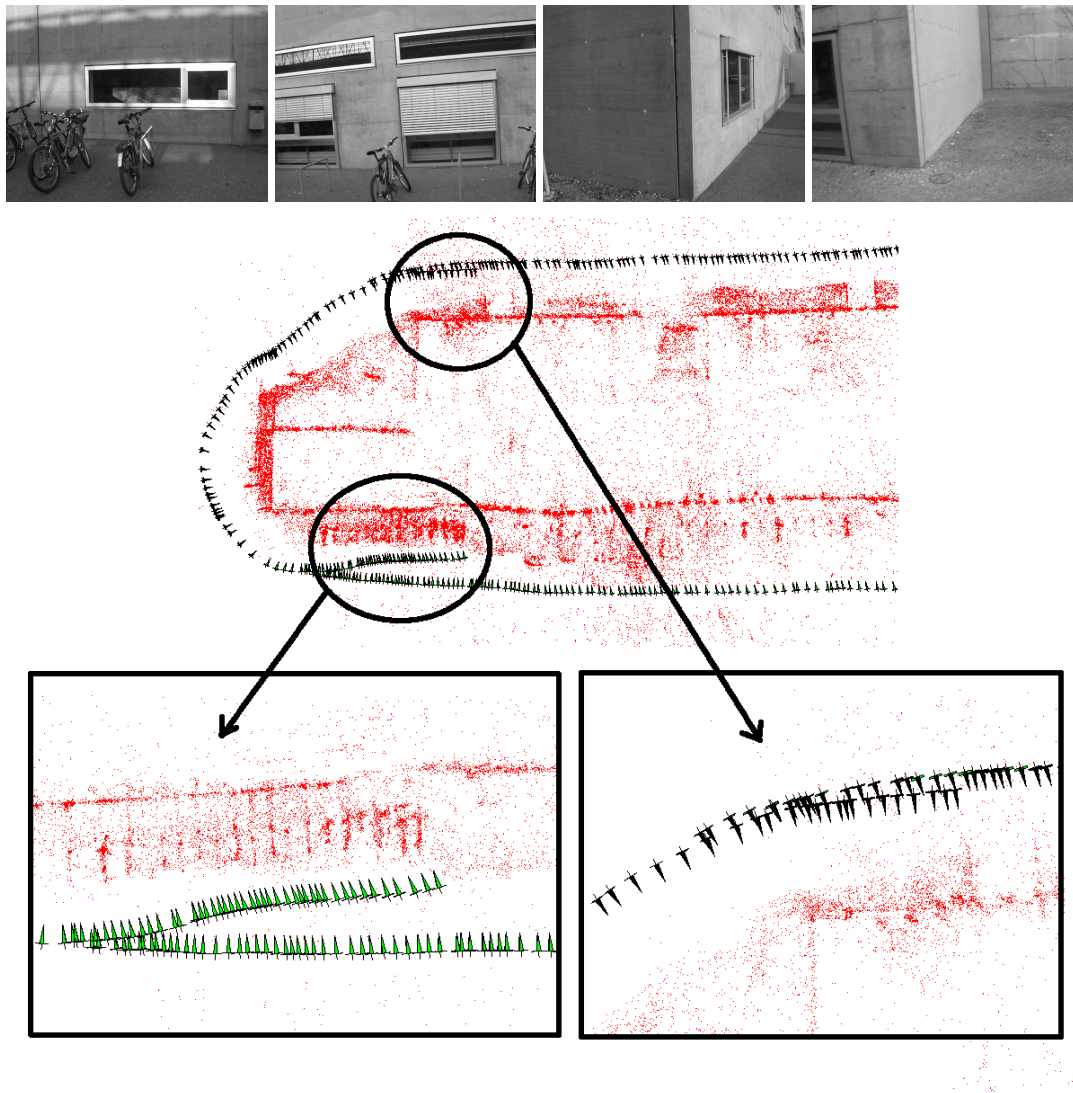


Figure 4.4: Merging of three odometry sequences in a difficult environment around a building. The pyramids represent camera positions, structure points are shown red. The three sequences overlap in two regions. These regions are highlighted in the zoomed in views. The merged reconstruction contains 449 cameras, the side length of the shown part of the building is about 20 meters.

structure is used at this point because structure can be severely distorted by drift and is difficult to match between different sequences. The image-based sequence overlaps are then used to connect common 3D structure.

Results show that sequence relations can be obtained for data sets even where single images cannot be matched unambiguously. The presented image-based techniques are

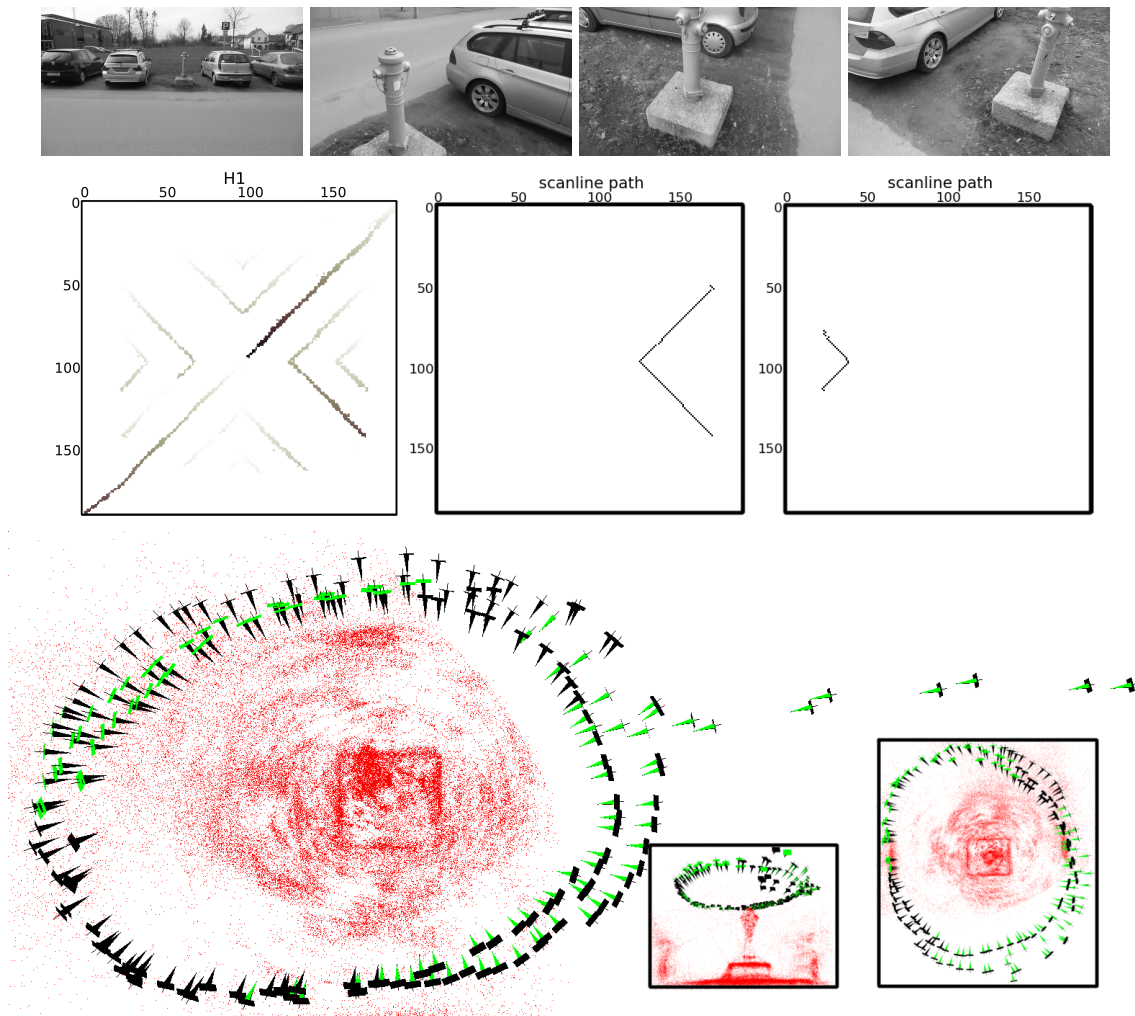


Figure 4.5: Sequence around a fire hydrant. The camera was moved towards the hydrant and then three loops were obtained. Before starting the initial visual odometry processing, a reversed copy of the video was added to the video stream, so that drift could be noticed easily. The second row shows the dynamic programming matrix H that was obtained by scoring the sequence with itself (the VSM diagonal scores have to be removed when a sequence is matched with itself) and two example correspondence paths. The last row shows the integrated reconstruction with successfully removed drift. Note that the reversed images do not have exactly the same position in space as the original because our odometry system selects images from the video stream dynamically.

particularly well suited for large scale reconstructions. Loop closing is just a special case of more general sequence overlaps. A limitation in our implementation at the moment is that the actual structure integration work is done using global bundle adjustment.

Chapter 5

Disambiguating Visual Relations Using Loop Constraints

The SfM method discussed so far uses an epipolar graph $G_{\mathcal{E}}$ as input. We have seen that we can relax the sequential nature of the reconstruction success by using correspondence information from locally matched image triplets that are not influenced by drift. Further robustness was gained by starting SfM on a hierarchical tree to delay the negative effects of the more difficult parts of the data.

In repetitive and ambiguous environments $G_{\mathcal{E}}$ itself is heavily corrupted by outlier matches. Identification of incorrect geometric relations between images solely based on low level features is not always possible, and a more global reasoning approach about the consistency of the estimated relations is required. We propose to utilize the typically observed redundancy in the hypothesized relations for such reasoning, and focus on the graph structure induced by those relations. Chaining the (reversible) transformations over cycles in this graph allows to build suitable statistics for identifying inconsistent loops in the graph. This data provides indirect evidence for conflicting visual relations. Inferring the set of likely false positive geometric relations from these non-local observations is formulated in a Bayesian framework. We demonstrate the utility of the proposed method in several applications, most prominently the computation of structure and motion from images.

5.1 Introduction

Computing the geometric relations from unorganized image sets purely from visual features is a difficult task. In order to obtain a tractable method, usually a pairwise matching procedure is applied first, which is followed by a fusion step to merge the initially obtained pairwise relations into some global reference frame. The approaches proposed in the lit-

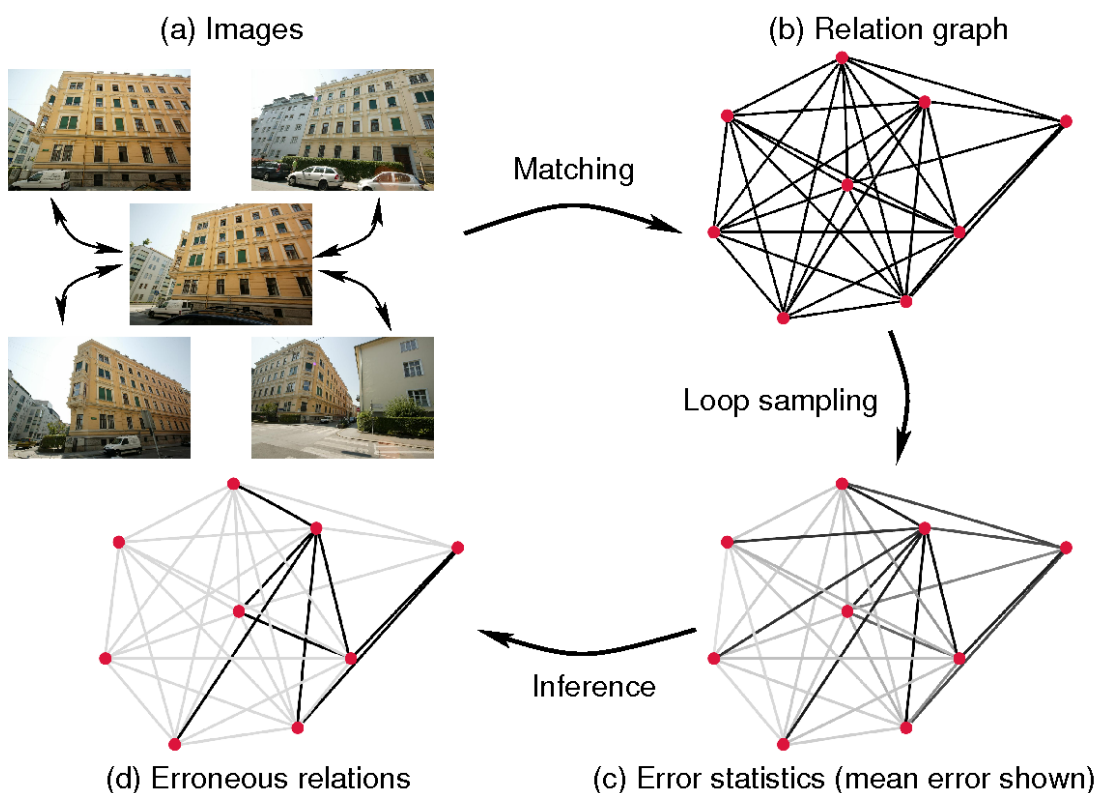
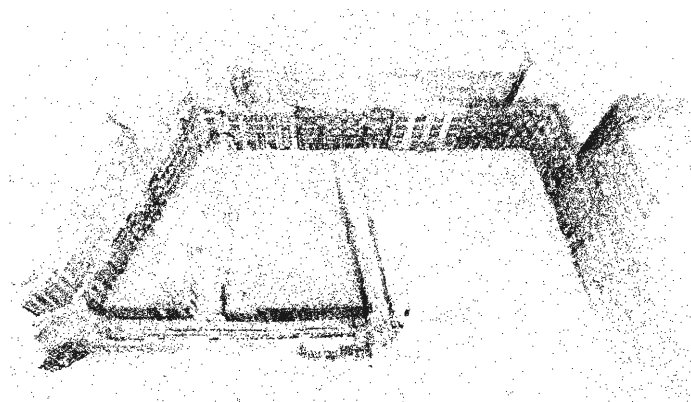


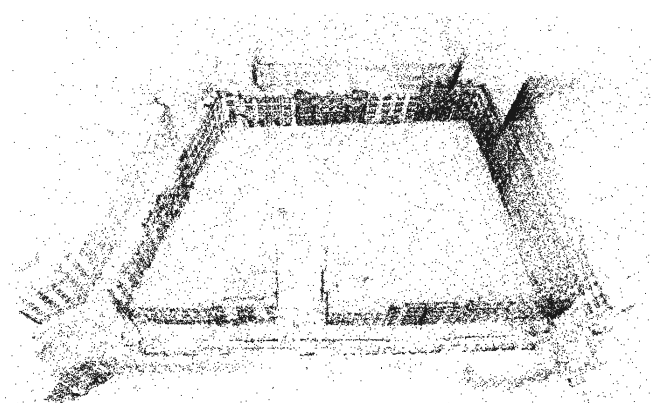
Figure 5.1: The original set of images (a) is robustly matched yielding a graph containing all potential pairwise relations (b). Acquisition of deviation statistics over loops results in non-local error observations (c), from which the incorrect relations are inferred (d). Large error and erroneous relations are indicated by dark edges. Observe that the central and the top right image look similar, but actually show different sides of the building.

erature vary widely in the details of this latter upgrade procedure. Since the first pairwise matching step uses only very limited information, the reported pairwise relations are susceptible to inconsistencies due to visual ambiguities, and the subsequent fusion method must be able to cope with such erroneous input. We do not restrict the notion of pairwise matching solely to images, but also consider e.g. mutual alignment of 3D point sets.

We propose to detect and remove conflicting pairwise relations, and thereby cleaning the input for the subsequent upgrade step from incorrect data. The principal components are illustrated in Figure 5.1. The pairwise relations generated by the preceding matching stage are typically highly redundant, which enables checking the intrinsic geometric consistency of these relations. The set of reported pairwise relations corresponds directly to a graph structure associating its edges with the relations (Fig.5.1(b)). Most classes of pairwise relations relevant in computer vision applications—e.g. homographies, relative pose, Euclidean and similarity transformations—allow the concatenation of geometric re-



(a) 3D model using all visual relations



(b) 3D model using only consistent relations

Figure 5.2: Distorted and correct output of a robust structure and motion pipeline using all geometrically verified epipolar relations (a) and using only those satisfying loop consistency (b).

lations to hypothesize new, potentially not directly observed relations. Large deviations between predicted (chained) and actually observed transformations indicate at least one conflicting edge among the involved relations. Under the weak assumption of invertible transformations we can restrict the focus on cycles in the graph structure. Concatenating the transformations along a loop in the graph should return the identity function in an ideal, noise-free setting. Again, the likelihood of having at least one incorrect edge in the loop is strongly related to the deviation of the chained transformation from the identity map. Collecting these statistics over loops indirectly points to potentially incorrect edges. The statistics for correct edges is generally contaminated by false positives also participating in the cycles, hence the conflicting edges cannot be read directly e.g. from the mean deviations (see Fig. 5.1(c)). Our proposed method uses a Bayesian network to infer the most likely set of incorrect transformations in the graph (Fig. 5.1(d)).

We propose a solution towards resolving the two conflicting goals encountered in 3D computer vision: creating as connected results as possible (i.e. maximizing the recall), while simultaneously avoiding incorrectly merged components (maximizing the precision). We augment robust vision methods addressing these issues with explicit reasoning steps on the geometric consistency in order to increase the recall while maintaining the precision. The objective of the proposed method is avoiding distorted results as seen in Figure 5.2(a) by removing conflicting visual relations as preprocessing step (Figure 5.2(b)).

5.2 Related Work

Extracting information from erroneous data is generally the goal of robust estimation; in computer vision random sampling [Fischler and Bolles, 1981] and its subsequent extension are practical approaches to robustly estimate a small set of parameters from data contaminated with outliers. In certain problem setting the L^∞ cost function can be used to identify outliers in the given data [Sim and Hartley, 2006], but usually robust cost functions like the Huber or Cauchy cost function are used in larger scale parameter estimation tasks (like bundle adjustment [Triggs et al., 2000]). Inconsistencies in the visual relations are only implicitly addressed and may result in arbitrarily distorted outputs.

Correctly separating unrelated structure-from-motion models, which are otherwise merged into an incorrect single representation due to visual similarities, is addressed in [Zach et al., 2008, Li et al., 2008]. An explicit Bayesian framework to detect false positive epipolar relations from undetected features is proposed in [Zach et al., 2008], which uses belief networks for view triplets to assess the correctness of epipolar relations. The procedure to generate a 3D model is very conservative and potentially leads to unnecessarily many separate models by assuming that detected false positive edges always link completely unrelated models. False positives found by means of [Zach et al., 2008] may also refer to incorrect relations within the same model. A method to disambiguate visually similar copies of well-known landmarks reconstructed from community photo collections is presented in [Li et al., 2008]. A combination of appearance-based clustering and geometric verification techniques is utilized to filter relevant images from unrelated ones, resulting in multiple unrelated instances or copies of widely known landmarks correctly being reconstructed.

In order to have an efficient method, we employ Bayesian inference on the abstract level of transformations between nodes (images, locally reconstructed models) and do not reconsider the association between e.g. image features and corresponding 3D points. In [Bibby and Reid, 2007] also the correspondence between image observations and latent variables is re-evaluated and possibly reverted, but this is applied only on smaller sub-

problems incorporating the recently observed data. The states of the latent variables and the associations are optimized by alternating minimization, therefore resembling the ICP method.

The method presented in [Govindu, 2006] tries to identify consistent relative rotations before determining global camera orientations using a RANSAC scheme by sampling spanning trees from the epipolar graph. The estimated hypothesis parameters are the global orientations of all involved cameras. Evidently, the size of the epipolar graph that can be handled in such an approach is rather limited, and the author uses a sliding-window procedure to reduce the problem size. We demonstrate that accumulating suitable statistics over cycles in the respective graph directly points to problematic edges, e.g. relative orientations. Further, Bayesian inference is much more tractable than random sampling for such a large hypothesis space.

Loops generated by linking smaller sub-maps are an important cue in robotics, in particular in simultaneous localization and mapping approaches. Upgrading the relative orientations between sub-maps to absolute orientations in a common coordinate frame using explicit loop constraints is proposed in [C. Estrada and Tardós, 2005]. We utilize a different approach following [Govindu, 2004, Martinec and Pajdla, 2007] to obtain globally consistent transformations from relative ones (see Section 5.5). Recently, the same authors suggest to consider only “compact” loop constraints derived from minimum cycle bases [C. Estrada and Tardós, 2009].

5.3 Inference of False Visual Relations

This section describes the inference of false positive relationships between images from observation gathered by chaining local transformations. First, we describe the underlying generative model based on loop inconsistencies, followed by a depiction of how these loops are sampled.

Inference from Loop Inconsistencies Let i and j be indices of images (or some entity derived from images), and T_{ij} is a hypothesized geometric relation between i and j e.g. obtained by robust estimation from feature correspondences. T_{ij} might be the relative pose, a homography, or a similarity transformation between locally reconstructed models. We require that T_{ij} is invertible, i.e. for a given T_{ij} the reverse transformation T_{ji} can be determined. In principle, it is not necessary that $T_{ji} = T_{ij}^{-1}$ holds exactly (e.g. both directions can be estimated separately), but for the sake of simplicity we assume that T_{ji} is the exact inverse of T_{ij} in the following.

If a set of transformations $\{T_{ij}\} = \{T_e\}$ is given, such that the underlying undirected graph $G = (V, E)$ with $E = \{e = (i, j)\}$ has cycles, then chaining all transformations along

a loop should result into the identity transformation (if one ignores noisy measurements for now). Let $L = (e_1, e_2, \dots, e_{|L|})$ denote an arbitrary loop in G with length $|L|$ and starting with edge e_1 , and T_L the accumulated transformation, $T_L = T_{e_{|L|}} \circ \dots \circ T_{e_1}$. If the transformations T_e are subject to measurement noise, then the deviation between T_L and the identity I follows some noise characteristic, which can be modeled for particular problem instances. We will measure the discrepancy between T_L and I using a non-negative function $d(T_L)$.

Observe that with G being a loopy graph in most applications, there is some redundancy in the set of hypothesized transformations $\{T_e\}$. If T_L deviates substantially from the identity map for a loop L , this strongly suggest that at least one of the individual transformations T_e in the loop is incorrect and should be discarded. By accumulating these deviations over a large set of loops one can obtain the statistics needed to infer the the set of false positives. If we visualize the mean deviations for a small example (recall Figure 5.1(c)), then one observes that one incorrect edge influences all loops containing this particular edge, and the mean error attributed to all edges in the graph is “blurred.” The main question is now how to infer the false positives from observation over cycles?

Obviously this problem can be casted as a Bayesian inference task. We introduce latent binary variables x_e for every edge, such that $x_e = 1$ indicates a false positive edge. The event that at least one of the loop edges is a false positive is abbreviated by $x_L = 1$, i.e. $x_L = \max_{e \in L} x_e$. We have to model two prior probabilities:

- The likelihood observing the deviation $d(T_L)$ for a loop under the assumption that none of the edges in the loop is incorrect, $P(d(T_L)|x_L = 0)$. This distribution is induced by the assumed noise model.
- The probability measuring $d(T_L)$ if at least one of the edges is a false positive, $P(d(T_L)|x_L = 1)$. As commonly employed in the literature, we generally model this likelihood by a uniform, least informative distribution. In our applications the range of $d(\cdot)$ can be easily bounded, and therefore $P(d(T_L)|x_L = 1)$ has finite support.

Optionally, a prior likelihood $P(x_e)$ can be provided for every edge, which can be determined e.g. from the confidence in the estimated transformation T_e . In our experiments we did not use the prior likelihoods (corresponding to a uniform prior on the unknowns). The structure of this belief network is illustrated in Figure 5.3. We are interested in an assignment for all the edge variables $x_e \in \{0, 1\}$ maximizing the joint probability

$$\prod_{L \in \mathcal{L}} P(\{x_e\}_{e \in L} | d(T_L)) \propto \prod_e P(x_e) \prod_{L \in \mathcal{L}} P(d(T_L) | x_L). \quad (5.1)$$

We have several options to perform (approximate) inference in this network. First, the Bayesian network can be directly converted to a factor graph representation by introducing factor nodes corresponding to the loops (and optionally factors for the unary priors).

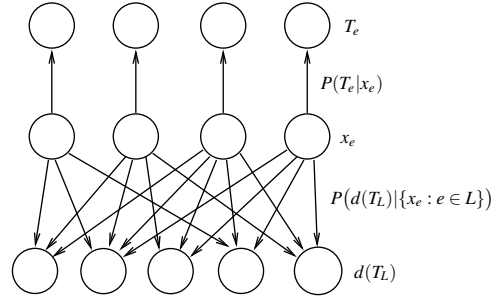


Figure 5.3: The Bayesian network for cycle inference.

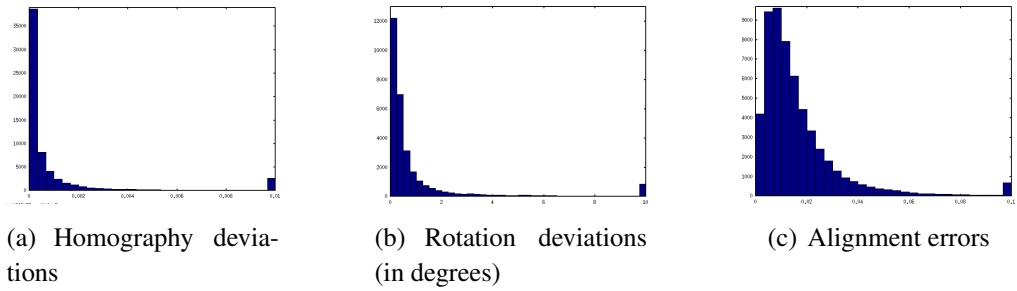


Figure 5.4: Histograms of empirical deviations from the respective identity map. The inlier portion of the histogram roughly follows an exponential distribution.

Loopy belief propagation (LBP) [Kschischang et al., 2001] is an efficient method for approximate inference in such graphs. We utilize LBP implementation provided by the libDAI library [Mooij, 2010].

Since the node variables and factors are tightly connected, and LBP does not provide quality guarantees, we also explored inference directly based on optimizing the energy functional corresponding to the joint probability Eq. 5.1. The log-likelihood of Eq. 5.1 reads as

$$\begin{aligned}
 E(\{x_e\}) &:= \sum_e l(x_e) + \sum_L l(d(T_L)|x_L) \\
 &= \sum_e \left(x_e l(x_e = 1) + (1 - x_e) l(x_e = 0) \right) \\
 &\quad + \sum_L x_L l(d(T_L)|x_L = 1) \\
 &\quad + \sum_L (1 - x_L) l(d(T_L)|x_L = 0),
 \end{aligned}$$

with $x_L := \max_{e \in L} \{x_e\}$. We abbreviate the cost coefficients of x_e and x_L by

$$\begin{aligned}
 \rho_e &:= l(x_e = 1) - l(x_e = 0) \quad \text{and} \\
 \rho_L &:= l(d(T_L)|x_L = 1) - l(d(T_L)|x_L = 0),
 \end{aligned}$$

which leads to

$$E(\{x_e\}, \{x_L\}) = \sum_e \rho_e x_e + \sum_L \rho_L x_L + \text{const} \quad (5.2)$$

subject to $x_e \in \{0, 1\}$ and $x_L = \max_{e \in L} \{x_e\}$. In order to obtain a convex problem, we replace the non-convex constraints $x_e \in \{0, 1\}$ by $x_e \in [0, 1]$. Next, the (non-convex) constraint set

$$C := \{(x_L, x_{e_1}, \dots, x_{e_{|L|}}) \in [0, 1]^{|L|+1} : x_L = \max_e \{x_e\}\},$$

linking x_L and the edge variables x_e , is replaced by the convex constraints

$$x_L \geq x_e \quad \forall e \in L, \quad x_L \leq \sum_{e \in L} x_e, \quad x_L \in [0, 1], x_e \in [0, 1].$$

Overall, determining the optimal x_e in the convex relaxation setting is now a linear program, for which efficient solvers are available. In our experiments we observed that most or even all variables x_e are either 0 or 1, and only a few variables attain fractional values. Hence, a branch and bound method is also a viable (and exact) inference procedure for this set of problems.

One interesting aspect of Eq. 5.2 is that the global solution (e.g. found by branch and bound) explains all inconsistent loops and there is no need to iterate the inference to detect additional conflicting edges. This can be seen as follows: Let $x^* = (\{x_e^*\}, \{x_L^*\})$ be the optimal solution of Eq. 5.2, and \mathcal{I} be the indices of inconsistent edge and loop variables, i.e. $x_k = 1$ iff $k \in \mathcal{I}$. The energy Eq. 5.2 can be split into two parts, $E(\{x_k\}) = \sum_{k \in \mathcal{I}} \rho_k x_k + \sum_{k \notin \mathcal{I}} \rho_k x_k$. Iterating the inference procedure corresponds to fixing x_k to one for all $k \in \mathcal{I}$ and only optimizing over the unknowns $\{x_k\}_{k \notin \mathcal{I}}$. Clearly,

$$E(\{x_k^*\}) \leq \min_{\{x_k\}_{k \notin \mathcal{I}}} \left(\sum_{k \notin \mathcal{I}} \rho_k x_k \right) + \sum_{k \in \mathcal{I}} \rho_k,$$

since x_k^* is the global minimizer of $E(\cdot)$ and $x_k^* = 1$ for $k \in \mathcal{I}$. Equality is attained by setting $x_k = 0$ for $k \notin \mathcal{I}$, hence no additional conflicting edges are reported by repeating the inference.¹ Since loopy belief propagation does generally not report global solutions, repeating the inference procedure may label additional edges as conflicting. We observed only minimal changes after the first inference pass.

Cycle Generation Generating all cycle in a loopy graph is obviously intractable, hence we need to restrict the number of inspected loops to a more manageable amount. In several graph-related applications the notion of cycle bases (optionally also augmented

¹ Here we ignore the possibility of different, equally global solutions.

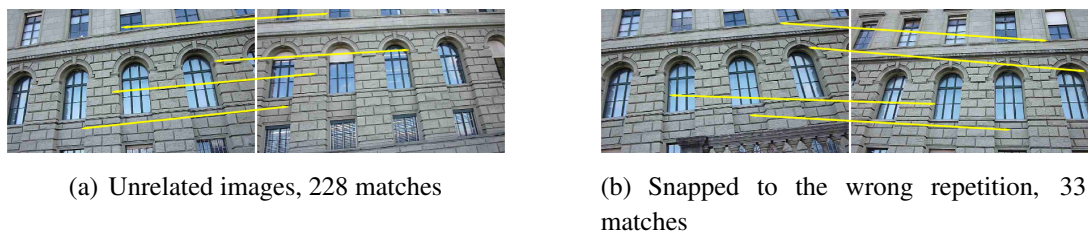


Figure 5.5: Rejected image pairs passing geometric verification using homographies. The yellow arrows indicate a few matching positions to assist the interpretation.

with minimality in some sense, see e.g. [Kavitha et al., 2009]) plays an important role. Cycle bases allow the generation of all loops in a graph by simple vector arithmetic in \mathbb{Z}_2 . We use the exhaustive set of cycles with length three together with loops induced by a so-called spanning tree bases. These cycle bases are derived from spanning trees of a (connected) graph by forming loops using non-tree edges. Thus, every edge in a graph not appearing in the spanning tree creates a loop together with the unique path on the tree between the respective nodes. In order to avoid explicit modeling of the transformation uncertainties with respect to the cycle length, we limit the maximal loop length to six.

Since in our application we strive for redundancy in the loop statistics, we use a sequence of spanning trees to gather more cycles in the graph. The first spanning tree is a minimum spanning tree induced by estimated edge uncertainties (i.e. derived from the number of inlier correspondences). Drawing loops using a spanning tree cycle basis leads to very uneven sampling of edges in the graph, since tree edges are part of cycles much more frequently than non-tree edges. Hence, we assign the weights used to determine the subsequent spanning trees inversely proportional to the number of sampled loops containing the respective edge. This approach ensures, that loop statistics over edges are acquired roughly uniformly. If several components of a graph are connected by only a few edges, data for these edges is still sampled very frequently. But these edges are usually very important e.g. to connect weakly linked parts of a 3D reconstruction, and acquiring well supported statistics for those links is a welcome feature.

5.4 Application: Homography Matching

This section discusses the specific details of our approach, when the geometric transformation between nodes (i.e. images) is described by homographies. In contrast to the fundamental or essential matrix used as primary transformation associated with edges described in the next section, homographies provide a very strong cue to verify their mutual consistency via chaining transformations along loops. If a set of hypothesized homographies H_e between two images i and j forming the edge $e = (i, j)$ in the graph network

is given, then we have $H_L = H_{e_{|L|}} \circ \dots \circ H_{e_1} \propto I$, $e_i \in L$, in the ideal, noise-free case. A simple, but effective way to measure the deviation of H_L from the identity matrix I is

$$d(H_L) := \min_{\alpha} \|\alpha H_L - I\|_F = \|\tilde{H}_L - I\|_F$$

with α determined as $\alpha = \text{tr}(H_L) / \|H_L\|_F^2$, and $\tilde{H}_L := \alpha H_L$. Observe that $d(H_L)$ is bounded by $\sqrt{3}$, since the elements of \tilde{H} have magnitude less or equal one. In order to obtain numerically stable results, all H_e 's are computed from normalized feature positions in $[-1, 1]^2$ (i.e. translated with respect to the image center and scaled by the reciprocal image width). This ensures roughly equal magnitudes for all the elements of H_e .

For real data, the deviations $d(H_L)$ between the concatenated homographies and the identity map is a sharply decreasing function for correct transformations (see Fig. 5.4(a)). Hence, we observe that $d(H_L)$ is much smaller than $\sqrt{3}$ for inliers, and the prior likelihood $P(d(H_L)|x_L = 0)$ can be modeled by an exponential distribution (we set its mean to 0.01). The observations $d(H_L)$ under presence of erroneous edges in the loop is generated by the least informative, uniform distribution, i.e. $P(d(H_L)|x_L = 1) \sim U[0, \sqrt{3}]$.

Figure 5.5 displays a few image pairs passing the geometric verification, but failing the stronger consistency check proposed in this section. This image sequence shows a highly repetitive, roughly planar facade (see Figure 5.6(b) for the 3D structure and camera path).

The currently dominant application for homography-based image alignment is the generation of panoramic images. The requirement of zero baseline between the image leads to a restricted class of homographies, for which specific minimal solvers and refinement procedures exist. It turns out, that the zero-baseline constraint is already quite strong, since it essentially rules out matching e.g. repetitive visual structures residing on the same facade. Future applications of homography verification are the extension of [Agarwala et al., 2006] from captured videos to unorganized image collections, and the enhancement of relative pose verification discussed in the following section.

5.5 Application: Screening the Epipolar Graph

The prototypical example for removing incorrect pairwise geometric relations between images is computing a 3D model from visual input. In [Snavely, 2008] several failure cases of the widely known Photo Tourism software [Snavely et al., 2006] are presented and discussed. In particular, the confusion of the structure and motion pipeline due to similar visual structures and scene repetitions is addressed. Due to the incremental structure of the Photo Tourism approach, failure cases are not solely induced by erroneous relations between images, but can also be the result of drift in the camera poses, or due to numerous outliers at the feature correspondence level.

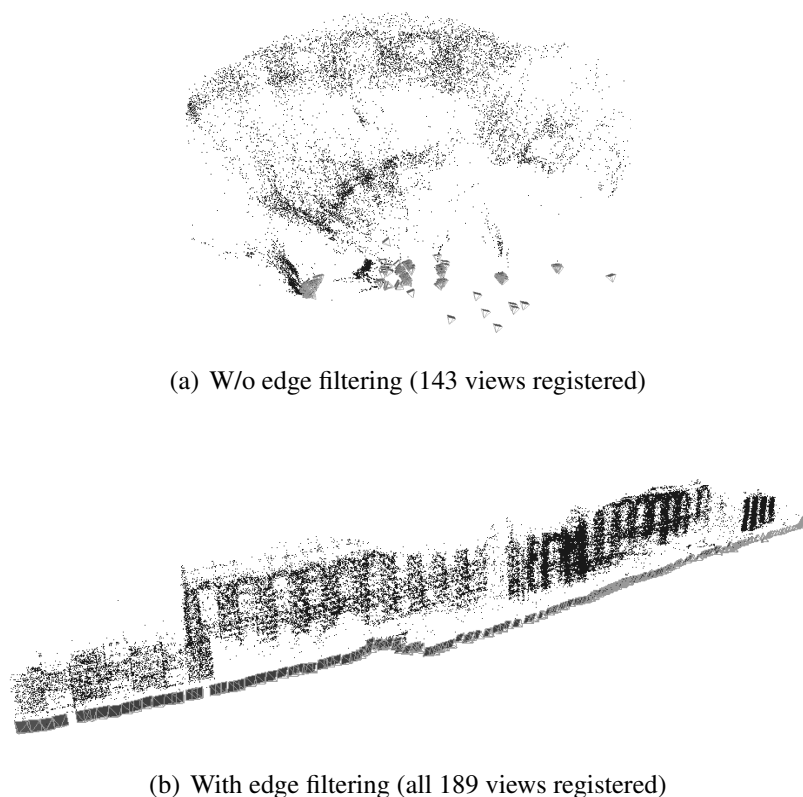


Figure 5.6: Model generated by Bundler for a facade with highly repetitive elements (a) without using epipolar graph filtering, and (b) with epipolar filtering solely using relative rotations.

This section discusses the detection of conflicting edges in the epipolar graph obtained by pairwise image matching and subsequent geometric verification. By assuming (potentially only roughly) calibrated cameras, each edge $e = (i, j)$ in the epipolar graph is associated with a relative transformation $(R_{ij}, t_{ij}) = (R_e, t_e)$ relating the coordinate frames of views i and j . Since merely the direction, but not the length of the baseline t_e is known, only the relative rotations R_e can be directly chained along a path. Similar to image-to-image homographies we have a consistency criterion over loops L ,

$$R_L = R_{e_{|L|}} \times \cdots \times R_{e_1} = I \quad e_i \in L \quad (5.3)$$

in a noise-free setting. In principle, there are (relatively weak) consistency conditions for the translation vectors t_e (e.g. [Govindu, 2001, Brand et al., 2004b]), but we restrict the discussion in this section to the rotation component. The next section describes stronger verification criteria, if the (relative) lengths of the baselines are known.

In a noisy setting, Eq. 5.3 holds only approximately, and the deviation of R_L from I is related to the likelihood for the correctness of the loop L . We use the rotation angle α_L

of R_L , i.e. $\cos(\alpha_L) = (\text{tr}(R_L) - 1)/2$, as the observed quantity for the sampled loops in the epipolar graph. For the inference procedure we need to model $P(\alpha_L|x_L)$. If the loop L is contaminated by an incorrect epipolar edge ($x_L = 1$), then we adopt $\alpha_L \sim U(0, \pi)$, since arbitrary accumulated rotations R_L can be generated in this case. If the loop is presumed to contain no error ($x_L = 0$), then the empirically observed angular errors can be approximately modeled by an exponential distribution (see Fig. 5.4(b)). We discovered in our experiments, that the exact choice of parameters for the fitted distribution (from a reasonable range) has a minor effect on the result of the inference. In our experiments we choose the mean to be 2 degrees.

In addition to extending our own structure and motion pipeline with loop consistency checks (see Section 5.6), we incorporated a “black-list” feature to the freely available Bundler software¹, which discards epipolar matches found to be incorrect by the proposed epipolar graph verification. Due to the specific incremental structure and motion approach employed in the Bundler software, the provided epipolar black-list is not fully utilized avoiding erroneous results. Nevertheless, solely verifying the epipolar graph can improve the resulting 3D model drastically as shown in Fig. 5.6. The reconstruction of a highly ambiguous facade is severely distorted without filtering the pairwise image matches (Fig. 5.6(a)) and correctly modeled otherwise (Fig. 5.6(b)). Figure 5.7 and 5.8 illustrate the identified “short-cuts” visible in the epipolar graph due to incorrect matching of repeating visual structures.

5.6 Application: Structure and Motion

While filtering the epipolar graph solely based on the consistency of relative rotation is already a powerful tool, additional inference steps can be applied subsequently. In particular, merging partial reconstructions obtained in initial steps of a structure and motion pipeline (e.g. as proposed in [Irschara et al., 2007, Havlena et al., 2009]) can benefit from verification of loop consistencies. In the following we briefly summarize our framework for structure and motion (SaM) computation².

SaM Computation Overview SIFT features extracted from the images are fed into a generic vocabulary tree in order to obtain a set of potentially matching images. Geometric verification based on essential matrix computation [Nistér, 2004] is applied on these image pairs using either the known intrinsics or approximate values from the EXIF tags. These steps required to generate the epipolar graph consume about 70% of the processing time, and the subsequent stages are computationally cheaper. The epipolar graph is

¹ <http://phototour.cs.washington.edu/bundler/>

² Corresponding software will be made publicly available at <http://www.inf.ethz.ch/personal/chzach>.

filtered using the method discussed in Section 5.5, and the remaining epipolar edges are used to generate image triplets. These triplets are geometrically verified. In order to be able to robustly handle undetected incorrect triplets, our approach is based on generating a set of small submodels (at most 15 views) first. These submodels are generated by random growth from a starting view and are highly redundant, such that every image participates in 10 submodels. Similarity transformations between triplets belonging to the same submodel are robustly determined, and the consistency of these transformations is verified as follows.

Triplet Verification Screening the homographies (Section 5.4) and epipolar relations (Section 5.5) identifies erroneous transformations, i.e. relations between visual entities. Inconsistent loops in the triplet graph indicate incorrectly established image triplets rather than erroneous transformations between triplets. Hence, we modify the interpretation of the latent variables (now x_k , where k is a triplet) to represent the validity of image triplets in contrast to edges/transformations between triplets. The generative model Eq. 5.1 remains the same otherwise. This conversion also lowers the number of latent nodes by orders of magnitude, since the number of edges in a triplet graph grows combinatorially with the connectivity in the epipolar graph.

It remains to discuss the utilized deviation $d(T_L)$ for chained similarity transformations $T_L = T_{e_{|L|}} \circ \dots \circ T_{e_1}$. T_L reads as 4-by-4 matrix, $T_L = \begin{pmatrix} s_L R_L & t_L \\ \mathbf{0} & 1 \end{pmatrix}$, with $s_L = \prod_k s_k$, $R_L = \prod_{|L|}^1 R_k$, and

$$t_L = \sum_k t_k \left(\prod_{j=k}^{|L|} s_j \right) \left(\prod_{j=|L|}^{k+1} R_j \right), \quad (5.4)$$

where s_k , R_k , and t_k are the scale, rotation and translation components of T_{e_k} . As in Section 5.4 we use essentially $d(T_L) = \|T_L - I\|_F$ to quantify the geometric inconsistency. Since the uncertainty (variance) in the relative translations t_k is multiplied by the respective factor in Eq. 5.4, we scale t_L by $\left(\sum_k \prod_{j=k}^{|L|} s_j \right)^{-1/2}$ to bring the translation component to a normalized range. The empirical distribution of $d(T_L)$ is illustrated in Fig. 5.4(c).

After verification of triplet correctness based on their relative transformations in the same submodel, the image triplets are upgraded into a common coordinate frame and a few (at most 10) iterations of a local bundle adjustment are performed. Subsequently, similarity transformations between the submodels can be hypothesized using 3D point correspondences, which can be filtered by repeating the loop inference.

Results Upgrading the triplets in a submodel into a common coordinate frame can still fail due to undetected erroneous visual relations. Such cases are discovered if a substantial fraction (25%) of the triangulated points are outliers (with respect to the reprojection error) or very few 3D points are visible in one of the cameras. Such submodels are

| Dataset | #views | #submodels | #components | largest component | inference time |
|---------|--------|------------|-------------|-----------------------|----------------|
| Abbey | 126 | 81/84 | 1 | 126 views / 29229 pts | 0.6s |
| | | 84/84 | 1 | 126 views / 29250 pts | 0.6s + 10s |
| Block | 479 | 176/229 | 3 | 238 views / 23126 pts | 17s |
| | | 215/229 | 1 | 476 views / 56230 pts | 17s + 27s |
| Block2 | 3482 | 786/1152 | 26 | 395 views / 42686 pts | 256s |
| | | 982/1152 | 29 | 550 views / 56233 pts | 256s + 80s |

Table 5.1: The effect of additional screening of similarity transformations between submodels. The first row for each dataset displays the characteristics only with epipolar graph filtering, and the second one shows the figures with all verification steps enabled. The inference times for the mixed integer solver (last column) are provided separately for epipolar screening (first number) and triplet verification (second value).

discarded and not considered in the final model generation. Table 5.1 summarizes the performance figures for several datasets with varying complexity. Adding this triplet verification step raises the number of submodels passing this criterion, and thus increases the size of the largest connected component in the final result.

5.7 Comparison Between BP and BnB Inference

The inference problem was solved with loopy belief propagation (BP) and with an LP solver using a branch and bound algorithm (BnB). BnB computes the exact solution if it converges (and the solution is binary), but BP has possible scalability advantages. We show a qualitative comparison of both methods:

5.8 Discussion and Future Work

We demonstrate that enforcing the consistency of geometric relations estimated from visual input identifies conflicting relations and assists in generating improved final results in several applications. An interesting extension of this screening of invertible transformations is the propagation of faulty relations detected by some other method (e.g. using the one proposed in [Zach et al., 2008] or even by user interaction) through consistent loops. This allows to infer additional erroneous visual relations by identifying only a very small set of incorrect transformations. The method proposed in this work specifically exploits the redundancy in the matching graph, but future research will address other sources of redundant information, e.g. visibility constraints that need to be satisfied when merging

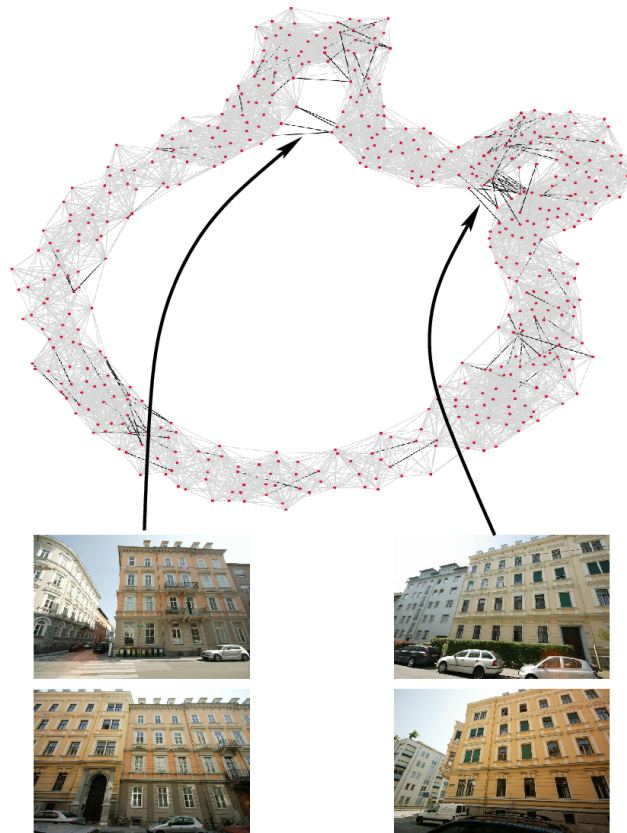


Figure 5.7: The epipolar graph with verified (light gray) and discarded (black) edges for the “Block” dataset (see also Figure 5.2), and selected image pairs corresponding to discarded edges.

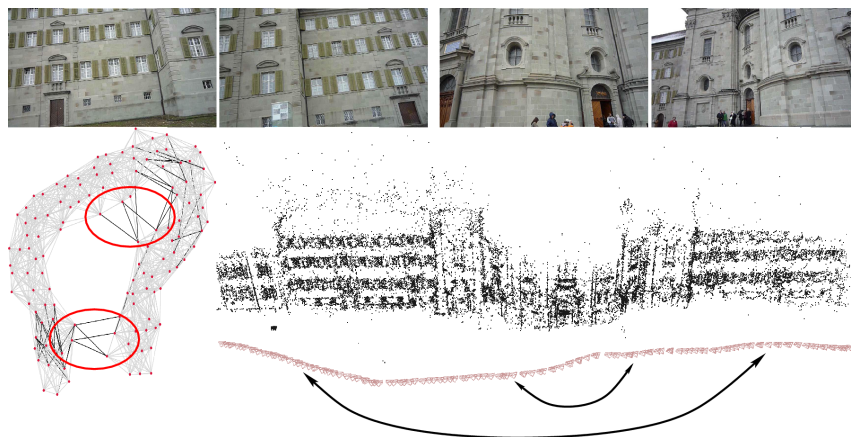


Figure 5.8: Epipolar graph with filtered edges (bottom left) and the reconstructed model (bottom right). The arrows indicate the approximate position of the erroneously matched images (top row, at the wings and at the central structure, respectively).

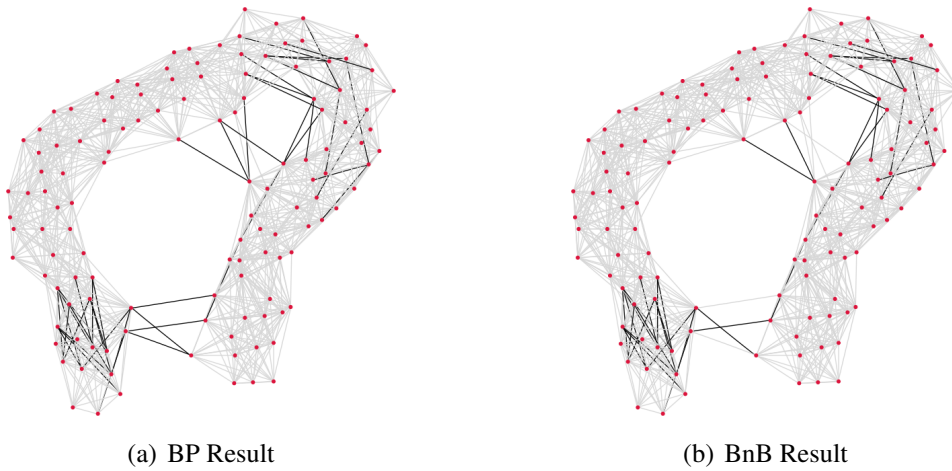
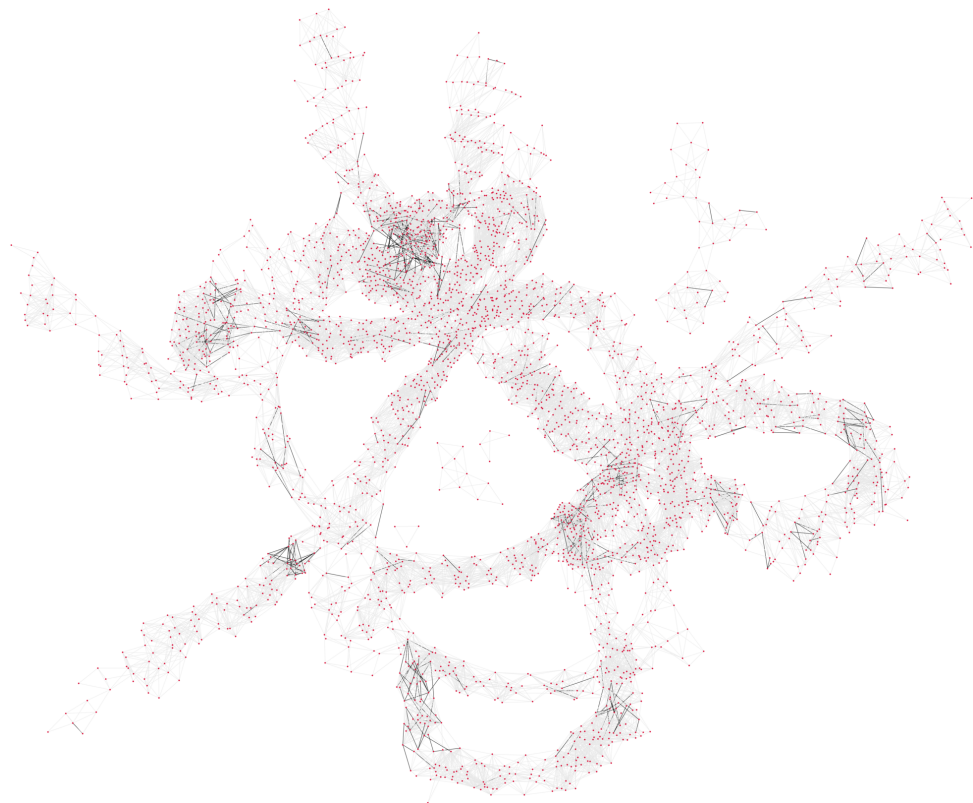
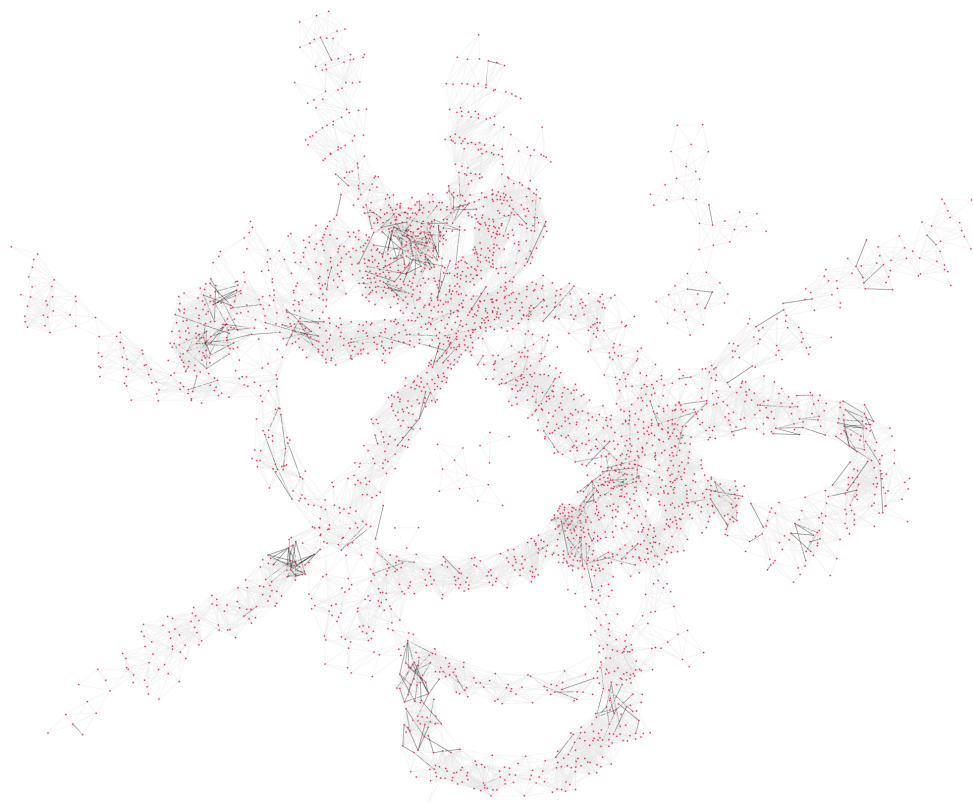


Figure 5.9: Dark edges represent inferred erroneous edges in the epipolar graph for the “Abbey” dataset using belief propagation (a) and branch and bound (b).

separately reconstructed 3D models.



(a) BP Result



(b) BnB Result

Figure 5.10: Dark edges represent inferred erroneous edges in the epipolar graph for the “Block2” dataset using belief propagation (a) and branch and bound (b).

Part II

Efficient Localization

Introduction

Image-based localization allows for high precision 6DOF pose estimation and self-contained operation without complex and expensive infrastructure. In this thesis the application of image based localization is focused on AR. Further applications include for example browsing photo collections within a spatial user interface and speeding up structure from motion computations.

Image based Localization in AR For a high-quality AR the current camera pose with respect to the environment must be estimated with full 6DOF at real-time update rates. Such 6DOF localization works for small workspaces [Wagner et al., 2008, Klein and Murray, 2009], but scaling such techniques up to larger environments is challenging, even more so with the additional constraint of limited computational power of mobile devices. We present two AR localization application scenarios in detail. The first use case is an indoor setting where no GPS data is available but strong assumptions about visibility of features can be made to reduce the search space and memory foot print. The second application is a large scale outdoor environment. Outdoor environments make feature matching even more challenging. The environment changes and is modified continuously, lightning conditions, weather and a moving crowd are difficulties that have to be handled. To enhance robustness we propose to increase the field of view of the localization image data. This can be done by letting the user capture panoramic images. This is a general, high level approach that can be used with different image features and matching techniques.

Further Localization Applications Photo Tourism [Snavely et al., 2006] uses 6DOF registered images to provide a 3D user interface for navigating image collections. This enables the user to experience the spatial relations of the relative image positions and navigate to desired view-points. In Photo Tourism images are registered in 3D using an incremental SfM implementation that makes heavy use of the absolute pose algorithm. This navigation idea has been integrated for example into Google maps, where image collections from Panoramio users are linked with the professionally captured street view image data. The position and orientation of the street view images are known. By matching user images (using a GPS prior and some planar facade assumptions) to registered street view images, navigation methods similar to Photo Tourism can be implemented on a much larger scale.

Another potential application of image based localization is speeding up SfM. Internet photo collections of famous places are highly redundant. Reducing the set of images to create in initial reconstruction can speed up SfM, as was demonstrated in [Snavely et al., 2008a, Irschara et al., 2009]. The main idea here is to reduce the set of images to an



(a)



(b)

Large Scale Offline Localization: Figure (a): A Google maps street view screen shot. This 360-degree street-level imagery is captured by professionals. Special purpose hardware mounted on a vehicle is used for localizing these images. Figure (b): Another Google maps screen shot. Panoramio image data that is captured by users using consumer digital cameras is superimposed onto the street view data.

essential set to span the reconstruction and then adding the redundant or additional images by computing the 6DOF position without modifying the initial reconstruction.

Chapter 6

Visibility Constrained Localization

Current off-the-shelf mobile devices offer integrated video cameras as a potential tracking sensor. These devices can be used as a complete user AR device, equipped with a display, computational power and at least one video camera. Mobile phones are a particular successful example of this class of devices. These devices have limited memory and computational power compared to desktop computers.

All units in a mobile phone are primarily optimized for low power consumption rather than raw processing speed. Additionally, memory is slow and caches are small, such that cache misses create serious performance hits. Code that is well written for a modern mobile phone will still run about 20-40 times slower than on a modern PC. Another important factor is memory size itself. Limitations in the mobile phone operating systems do usually not allow more than about 5-10 MB per application. However, modern smart phones possess storage capabilities of several gigabytes, making hierarchical data handling attractive. Although the file storage is too slow for live data processing, it can be used for out of core processing or to cache large datasets to prevent continuous downloading. Finally, the mobile phone's camera contributes to the level of detection quality that can be expected. We have seen enormous improvements in the still image taking quality of mobile phone cameras. Multi-megapixel cameras are common today, and given enough light, the quality is now comparable to compact cameras. Video capabilities, however, have only minimally improved, making it hard to select a good device for AR usage. A limitation comes from the small field of view of mobile phone cameras, which typically lies in the range of $50^\circ - 60^\circ$.

With these additional hardware limitations, a state of the art vision based re-localization scheme like [Irschara et al., 2009] can not be used directly on current mobile devices. The main reason is that [Irschara et al., 2009] needs access to the entire search data structures that represent the 3D scene in main memory. We propose to partition the 3D scene and therefore search data structures into potentially visible sets (PVS). The visibility informa-

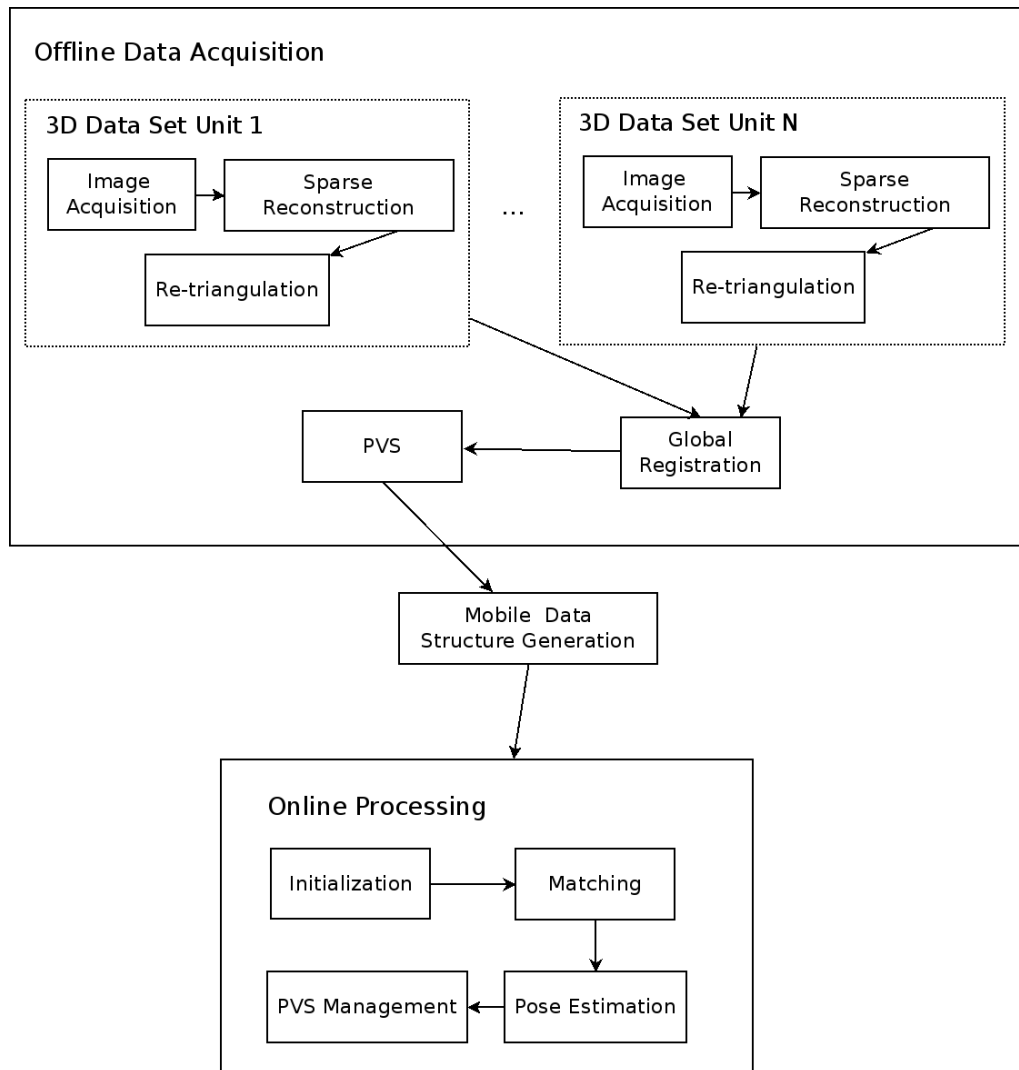


Figure 6.1: *Workflow overview.* The system can be divided into an offline data acquisition and an online localization part. The offline data acquisition step creates globally registered 3D points with associated feature descriptors that are grouped into potentially visible sets. Optimized representations of this data are used on the mobile phone to localize the user efficiently.

tion can be obtained for example from floor planes for indoor navigation, footprints of buildings or estimated automatically.

Starting with a coarse guess, our system only considers features that can be seen from the user's position. Our method is resource efficient, usually requiring only a few megabytes of memory, thereby making it feasible to run on low-end devices such as mobile phones. At the same time it is fast enough to give instant results on this device class.

6.1 Overview

Our approach builds upon the idea of using image-based reconstruction methods from computer vision to solve the data acquisition step. Because the image-based reconstruction methods operate on the same type of data as the image based localization and pose estimation, it is straightforward to feed data from a 3D reconstruction into a localization system. Furthermore, reconstruction and localization both depend on texture information present in an environment. This means that if an environment is suitable for image-based localization, it is usually also suitable for image-based reconstruction and vice versa. A major advantage of this approach is that we do not have to segment our environment into explicit tracking targets and we are not limited to planar or other special geometrical structure. On this account we propose to use a globally registered image-based reconstruction of an environment as a large 3D tracking target.

Because real world buildings usually contain untextured areas that are not suitable for image-based reconstruction or localization we use a pragmatic approach to create large-scale reconstructions. We divide the problem of reconstruction into suitable blocks and do not try to reconstruct complete buildings in one step. In this work, we manually register individual reconstruction into a single global coordinate system to create a single large globally registered 3D reconstruction which serves as a base for self-localization. Automatic global registration of reconstructed parts is left for future work.

Our goal is to build a system which is highly efficient and can run on mobile phones. In order to accomplish this, the reconstruction has to be represented efficiently in terms of memory consumption, scalability and computational handling. Apart from that, a pose estimation algorithm must be fast, yet robust to deliver stable results even under the presence of a significant amount of noise and outliers. We propose an approach for generating a compact representation of 3D reconstructions in an indoor environment which is highly efficient in terms of memory consumption and handling on mobile phones.

Taking advantage of well-known approaches from computer graphics for visibility estimation, we propose a method for dividing large 3D reconstructions into several smaller parts that can be compactly represented and allow for building a highly scalable system. We build upon the idea of potentially visible sets originating from work in the computer graphics community.

6.2 Related Work

In the following we review related work in the area of localization for AR, as well as previous work on localization purely based on computer vision techniques. Furthermore we introduce some work previously done for location recognition on mobile phones and

the occlusion handling principle used in our approach that originates from well established concepts in computer graphics.

6.2.1 Localization for Mobile Phones

Because phones did not have sufficient computational power, the first approaches for mobile phone localization were based on a client-server architecture [Fritz et al., 2006, Hile and Borriello, 2007]. To overcome resource constraints of mobile phones, tracking is outsourced to a PC connected via a wireless connection. All of these approaches suffer from low performance due to restricted bandwidth as well as the imposed infrastructure dependency, which limits scalability in the number of client devices. The AR-PDA project [Gausemeier et al., 2003] used digital image streaming from and to an application server, outsourcing all processing tasks of the AR application reducing the client device to a pure display plus camera. Hile reports a SIFT based indoor navigation system [Hile and Borriello, 2007], which relies on a server to do all computer vision work. The server-based approaches are not suitable for AR; typical response times are reported to be about 10 seconds for processing a single frame.

However, recent approaches have shown that natural feature tracking with 6DOF can be done in real-time on mobile phones [Wagner et al., 2008]. Takacs *et al.* [Takacs et al., 2008] present an outdoor localization system working on mobile devices. Keypoint detection and matching is performed directly on the mobile phone. Features are clustered in a 2D grid and pre-fetched by proximity. Each grid element contains a set of clustered meta-features that represent the most repeatable and stable features of the scene. Geometric consistent meta-features are created. However, in contrast to our approach no 3D model is reconstructed and thus true geometric consistency is not enforced and no full 6DOF pose is computed.

6.2.2 Potentially Visible Sets

In computer graphics and virtual reality applications, visibility is a fundamental problem, since it is necessary for occlusion culling, shadow generation and image-based rendering. One of the earliest methods addressing the problem of visibility culling is the potentially visible set (PVS) approach [Airey et al., 1990]. The basic idea is to discretize 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 reduces the amount of data that has to be processed for rendering. Indoors, the natural structure of cells (rooms) and portals (doorways) can be easily exploited [Teller and Séquin, 1991]. In our localization system, we take advantage of the

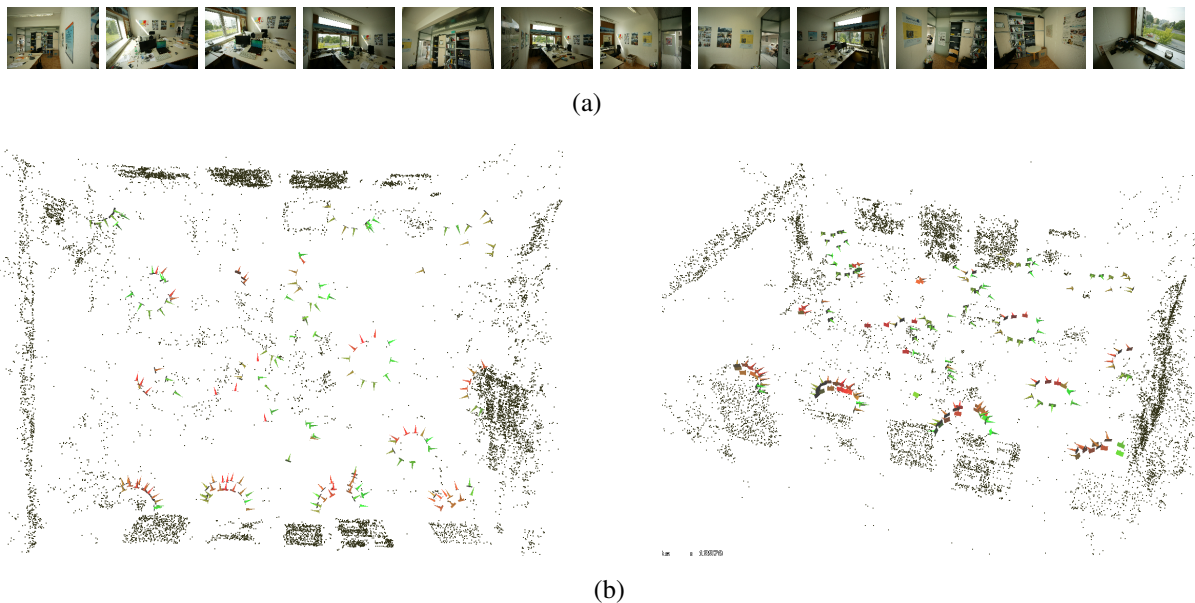


Figure 6.2: *3D segment*. Figure (a) shows some of the input images used to create an initial sparse 3D reconstruction. Figure (b) shows two views of this reconstruction, triangulated natural feature points are shown as dark points, the camera positions and orientations of the input images are visualized by the coloured cones.

PVS data management by splitting the 3D model into chunks of 3D points that are organized by visibility constraints. Thus, the potentially visible set determines the number of 3D points and descriptors that have to be considered according to the current view cell. This is in contrast to the 2D grid method of Takacs *et al.* [Takacs et al., 2008], which uses a regular subdivision on a map to partition the data and does not exploit visibility.

6.3 Offline Data Acquisition

This section explains the steps involved in creating the 3D tracking data in more detail and motivates some design decisions. Triangulated natural image features, obtained with a structure from motion (SfM) system, are used as a basis. These 3D points are registered into a global coordinate system and partitioned into a representation suitable for efficient localization on mobile phones.

Feature Extraction and Triangulation

While the reconstruction pipeline uses a standard SIFT key point detector and descriptor, the detection on the mobile phone must be optimized for speed and therefore uses a pro-

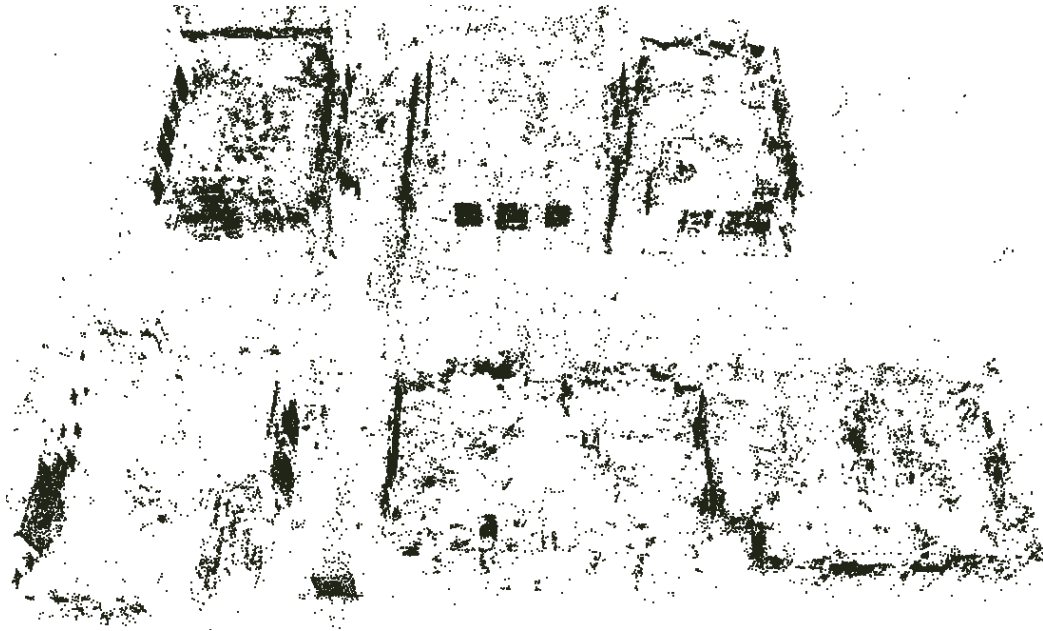


Figure 6.3: Global registration of individual reconstruction into a common coordinate system.

prietary method. Consequently, we must extract new features from the original images and triangulate them using the camera poses obtained from the SfM step. This adds no significant overhead to the reconstruction process, because the computational effort of pair-wise image matching has already been solved during reconstruction and well matching image pairs are already known.

6.3.1 Global Registration

Our goal is to compute the full 6DOF pose, therefore each 3D reconstruction has to be registered relative to a global coordinate system. We find this similarity transform by aligning each of our newly created reconstructions manually to a 2D floor plan of the building. All 3D points used for localization are transformed into a single global coordinate system. Another option would be to compute the 6DOF pose relative to a reconstruction segment and transform the result into the global coordinate system. The advantage of transforming all 3D points into one coordinate system is that 3D points from multiple reconstruction segments can be used simultaneously during the localization pose computation. This may happen if feature points from multiple reconstruction segments are visible in the same image. Figure 6.3 shows an example of eight globally registered reconstruction segments.

6.3.2 Potentially Visible Sets

Since the overall feature database does not fit into memory, it is necessary to split the dataset into chunks that can be loaded independently. Takacs *et al.* [Takacs et al., 2008] suggested to partition the database into cells using a 2D regular grid. In their approach, the computer always holds the feature sets of the closest 3x3 cells in memory. In contrast, we split the dataset by visibility, using a PVS structure.

We follow a common practice that PVS are often built at the same granularity as the cells that they index: For every cell there exists one PVS that refers to all other cells that are visible from inside this cell (see Figure 6.4). Each cell stores a list of all features that are located inside its area. The associated data is stored in a separate file and loaded on demand. Figure 6.5 shows such a partitioning of a larger 3D reconstruction into individual cells. While automated techniques to compute the PVS exist [Teller and Séquin, 1991], for simplicity the PVS structure used in our evaluations was created by hand.

For each feature, we also store the indices of all images containing the feature. This additional information can be used to vote for areas that are likely to be seen in a camera image. For each PVS, we build a k-means tree for fast searching and voting, using Lloyd's algorithm [Lloyd, 1982]. Additionally, we built an inverse file structure in the form of additional k-means trees for all original camera images that were used to reconstruct the PVS. The inverse file structure allows matching against only those features that come from a specific image. This approach greatly improves the matching rates, but comes at the price of increased memory usage for the search structures and probably a reduced matching accuracy.

We experimented with various branching factors to build the k-means trees and found a branching factor of 10 to deliver good results at reasonable speed. In our tests, a larger branching factor did not deliver significantly better results. We grow the trees until a given maximum number of descriptors fits into the leaf nodes. In our current implementation we target an average of 45 descriptors per leaf node. When searching for a match, we traverse the tree until encountering a leaf node and then compare against all descriptors in that node. Hence, we do not require back tracking.

6.4 Localization

In previous work [Wagner et al., 2008], an extremely fast descriptor called PhonySIFT was developed. It is loosely based on SIFT, but designed to operate in real time on a mobile phone. One restriction is that tracked objects must be planar, which allows highly effective outlier removal, yielding enough robustness for typical AR applications. However, for the use case of indoor localization, the PhonySIFT descriptor is not sufficient.

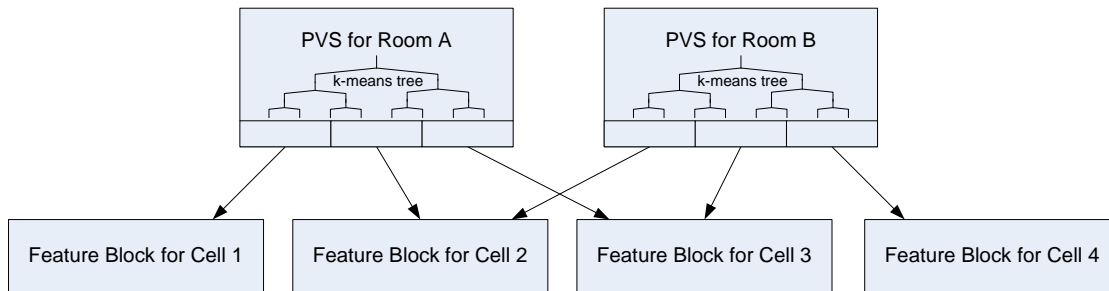


Figure 6.4: Two PVS with three cells each, sharing two of the cells.

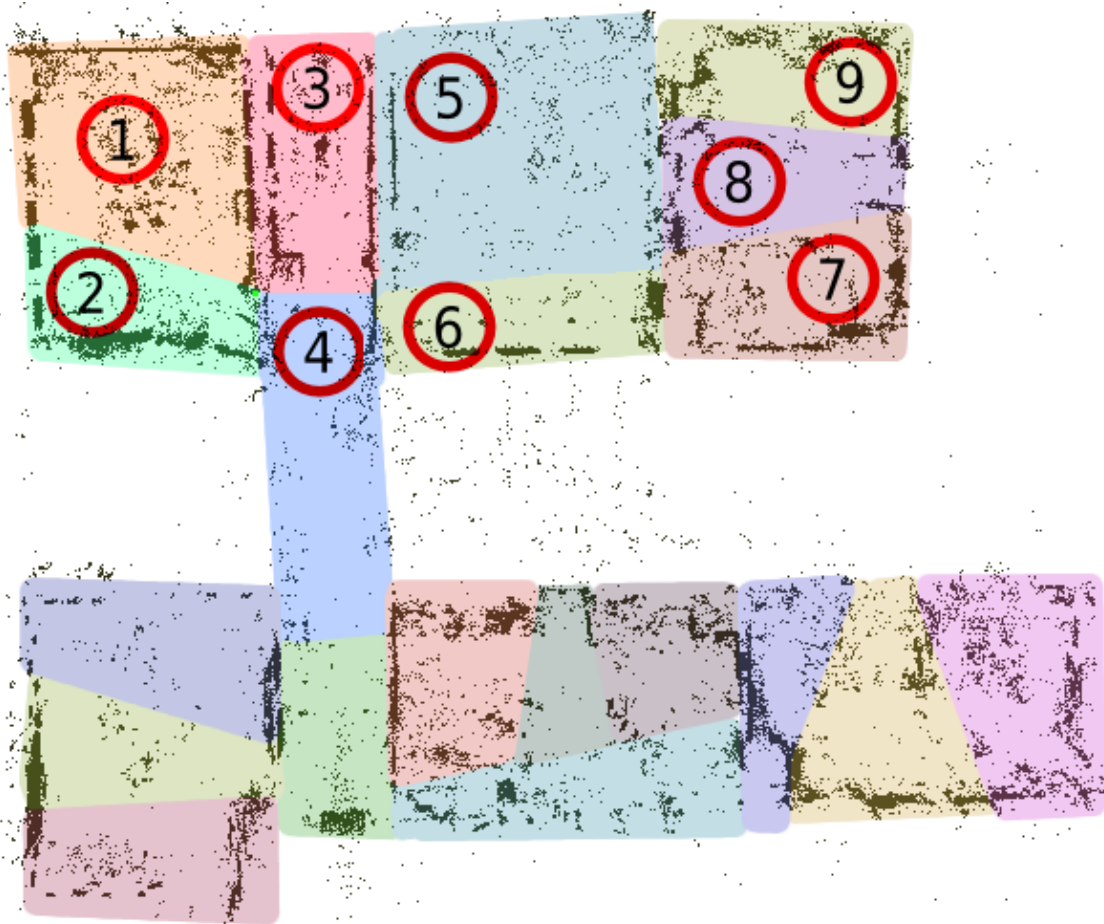


Figure 6.5: Representation of the reconstruction as separate PVS cells. The assignment of features to cells is color coded. The first 9 cells are indicated by the numbers inside red circles. Because we use cell based visibility, a PVS is created for each of these color coded cells and consists of an enumeration of all cells that are visible from one cell. For example, the PVS of cell 2 consists of cells (2, 1, 4).

The outlier removal techniques used in [Wagner et al., 2008] do not work for 3D objects, and more importantly its robustness does not scale well enough to match against tens of thousands of key points.

On this account, a new proprietary detector/descriptor combination which was inspired by SURF [Bay et al., 2006] is used. Key points are identified in scale space and a histogram of gradients is built. Tests done with the framework of Mikolajczyk *et al.* [Mikolajczyk and Schmid, 2003, Mikolajczyk et al., 2005] show that it performs as well as SIFT and SURF and sometimes outperforms both. However, we noticed that SIFT performs better in some real-world scenarios, which is why we continue to use it for the 3D reconstruction. Our detector/descriptor is several times faster than the publicly available implementation of SURF¹ and faster than a GPU-SIFT implementation (comparing our method on a single-core of a 2.5GHz Intel Core 2 Quad against GPU-SIFT on an NVIDIA GeForce GTX 280). For localizing and describing keypoints in a 640x480 image, our descriptor requires about 20ms on that 2.5GHz CPU and about 120ms on a mobile phone².

6.4.1 Run-time Memory Management

Feature and PVS data is stored in a format that is optimized for fast loading, which includes storing items in large chunks and minimizing the number of memory allocations required. The feature datasets of the cells are stored in feature blocks. A custom memory manager loads and discards feature blocks by counting references from the PVS structures to the cells. When a cell is no longer required by any PVS, the memory manager can discard it to free memory for other cells. Feature blocks are loaded on demand when a PVS requests them. In our tests, a feature block typically has a memory footprint of ~ 1 -2MB and a PVS is built from 2-4 cells. The memory footprint of the PVS is dominated by the search structures and range between ~ 1 -2MB. The overall memory footprint is ~ 5 MB, which is small enough to fit into a mobile phone's application memory.

6.4.2 PVS Selection

Since only a part of the whole dataset is in memory at a given time, no global localization can be performed. Consequently, it is necessary to initialize the localization process by giving the system a hint about where to search. In the following, we outline several variants for providing an initial location for the PVS.

¹ <http://code.google.com/p/opensurf1>

² calculated on an ASUS P565 mobile phone with an Intel XScale processor running at 800MHz for about 240 features.

The most user friendly approach relies on additional sensors and therefore does not require any user interaction. Outdoors, GPS can give a sufficiently accurate position for searching a single PVS only. Indoors, WiFi triangulation, Bluetooth or infrared beacons are able to deliver a coarse location, which requires searching the closest few cells. If no sensor data for automatic coarse localization is available, the user can select the current position on a map or from a list (rooms, streets, etc). If the tracked object or environment does not have a unique location (for example, a consumer product for advertising), the dataset can be manually selected or determined using a barcode on the object. Once the mobile device is localized, the system can switch into an incremental tracking mode.

Compared to first initialization, a re-initialization step can then benefit from the fact that a previous position is known. If the time between losing tracking and re-initialization is short, it makes sense to restrict the search area to the PVS of the last known position. However, in many practical scenarios tracking interruptions will last longer (*e.g.*, many users may put a handheld device down while walking). Depending on the assumed speed of a user, the search area must therefore be widened over time until it reaches a level that can not be searched in a meaningful time anymore.

6.4.3 Localization

Localizing a mobile users position involves the following steps: feature extraction and description, feature matching, optional voting, 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 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 3-point

pose [Haralick et al., 1991] as hypothesis and use a subset of up to 50 correspondences for validation.

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.

6.4.4 Detection vs. Tracking

Tracking by detection at every frame is common, because detection is always required and tracking is then solved at the same time. Our wide-area detection system runs in about 1/3 of a second on a fast mobile phone and about 10 times faster on an average PC. It would therefore, at least on the PC, qualify for tracking by detection.

However, tracking by detection has several disadvantages compared to an approach with separate detection and tracking stages as in [Wagner et al., 2009]. A pure detection system does not take advantage of frame-to-frame coherence and therefore always searches without a strong prior. This requires more computational effort and reduces the chances of finding a correct solution. A tracker pre-selects the features to look for using a motion model and therefore reduces the search space. It is therefore typically more robust and efficient than a detector. However, tracking usually requires a different kind of dataset compared to the detector, which means that two separate datasets have to be created, maintained, stored and held in memory.

6.5 Experiments

This section presents evaluation results for the wide-area localization. For simplicity, the test cases are based on indoor reconstructions taken at the university campus. Note that although we do not explicitly provide results on outdoor scenarios, our methods are also fully applicable outdoors.

6.5.1 Test Data Acquisition

For testing our approach we generated a fully registered 3D reconstruction of several adjacent rooms in an office building on the university campus, using a digital SLR camera and

| Database | PVS 1 (cells 1,2,4) | PVS 2 (cells 2-4,6) | PVS 3 (cells 4-6) | Grid (whole area in Fig. 6.3) |
|----------|------------------------|------------------------|----------------------|----------------------------------|
| Size | 5,378 kB | 5,159 kB | 2,348 kB | 16,789 kB |

Table 6.1: Amount of memory needed for storing the individual working sets, the PVS as marked in Figure 6.5 and the entire reconstruction as shown in Figure 6.3.

the methods described in Section 6.3. For the localization experiments, two smartphones were used, a *Meizu M8*¹ and an *Asus P565*. One of them, the *M8*, was used to take a set of 94 high resolution pictures while walking along a specified path through three rooms of this reconstruction. For every position, two images were taken, one with the built-in lens of the *M8* and one with a special adhesive wide-angle lens from *AmacroX*², which can be mounted on a phone. A collection of images for both lens types is shown in Figure 6.6. The phone was mounted on a tripod to ensure that the image pairs with and without the *AmacroX* lens were taken from the exact same position. The mobile phone and the setup are depicted in Figure 6.7. For simplicity, we calibrated the camera once with and without the *AmacroX* lens, but did not recalibrate the camera each time after reattaching the wide angle lens.

Using two different lenses makes the optimal choice of parameter settings for the various algorithms more difficult. Moreover, effects resulting from the radial distortion of the *AmacroX* are difficult to analyze. Nevertheless, we included the image sets obtained with both lenses to better understand the effect of field of view in complex localization problems.

6.5.2 Memory Consumption

We investigated the effect of using a PVS structure on the size of working set selected from the overall database. The memory consumption of the PVS structure is compared to a regular subdivision as used by by Takacs *et al.* [Takacs et al., 2008]. Results for the working set consisting of descriptors k-means tree, using 8-bit quantized values for every histogram entry, are listed in Table 6.1.

Note that the amount of memory for the databases in the individual PVS is considerably smaller than the amount needed for an entire area. Also note that due to our management of sharing cells among PVS structures (see Section 6.4.1), the amount of memory is even smaller. This means that the values in Table 6.1 refer to the maximum amount of memory needed and include shared cells.

¹ <http://www.meizu.com>

² <http://www.amacrox.com>

| Strategy | Exhaustive Matching (1) | Tree-based Matching (best 3 candidates exhaustively) (2) | Tree-based Matching (best 3 candidates tree-based) (3) |
|----------|-------------------------|--|--|
| Time | 17.95 s | 1.660 s | 150.78 ms |

Table 6.2: Runtime performance for different matching strategies on the *ASUS P565*.

6.5.3 Matching Strategies

In the preparation of our performance tests, we evaluated different strategies for generating correspondences between features detected in a given 768x576 pixel image and the features contained in a database (refer to Table 6.2 for averaged results). We tested exhaustive matching (1) and k-means tree based matching with subsequent histogram filtering, where the best 3 candidates were evaluated. For matching with the features of the best candidates we tested two configurations, one using exhaustive matching (2) and one using a k-means tree (3). For all three strategies we matched five test images against a database containing 40,670 features.

In the query images, on average 784 features were detected. The runtimes for the second and the third approach also include the time needed for histogram voting. Note that the second and the third method only approximate the result, while only the first method delivers the exact matching. Nevertheless it is easy to see that in terms of the computation time, the third strategy outperforms both other methods by far. Consequently it was adopted as our preferred method for our online localization. For a comparison of the accuracy, the reader is referred to the next experiment.

6.5.4 Image Resolution

The aim of this experiment is to evaluate the influence of image resolution on robustness. Starting with an initial resolution of 1920x1440 pixels (the maximum still image resolution of the *Meizu M8* camera), we iteratively reduced the image resolution down to 320x240 pixels. For all image resolutions, we allowed a reprojection error of 1.5 % of the corresponding focal length, which is equal to about 20 pixels in the largest resolution. In the lowest image resolution, this is equivalent to about 3.3 pixels. To keep the rate of features detected in the images at a reasonable level, we lowered the threshold for detecting features with decreasing image resolution, starting with a relatively high threshold. This was mainly done to remove the large amount of noise due to very small features detected in larger image resolutions.

To prove the performance of our matching approach based on individual k-means trees and histogram-based filtering of candidates (with consecutive k-means tree based

| | Image Resolution | 512x384 | 640x480 | 768x576 | 1024x768 |
|-----------|-----------------------------------|------------------|------------------|------------------|--------------------|
| | Avg. Num. of Features | 372 | 557 | 784 | 1253 |
| Meizu M8 | Feature Extraction | 278.30 ms | 387.58 ms | 593.11 ms | 2,155.1 ms |
| | Histogram Voting | 32.48 ms | 48.18 ms | 67.25 ms | 109.96 ms |
| | Matching best (max.) 3 cand. + RP | 129.15 ms | 178.95 ms | 214.24 ms | 302.77 ms |
| | Total Time | 439.94 ms | 614.71 ms | 874.60 ms | 2,567.83 ms |
| ASUS P565 | Feature Extraction | 125.72 ms | 193.19 ms | 306.11 ms | 481.85 ms |
| | Histogram Voting | 26.69 ms | 39.5 ms | 54.59 ms | 90.57 ms |
| | Matching best (max.) 3 cand. + RP | 96.18 ms | 120.20 ms | 141.77 ms | 237.79 ms |
| | Total Time | 248.59 ms | 352.89 ms | 502.43 ms | 810.21 ms |

Table 6.3: Runtime performance for different image resolutions and the individual algorithms contained in our approach.

| Individual Algorithm | % [512x384] | % [640x480] | % [768x576] | % [1024x768] |
|-----------------------|-------------|-------------|-------------|--------------|
| Feature Extraction | 50.57 % | 54.75 % | 60.93 % | 59.47 % |
| Histogram Voting | 10.74 % | 11.19 % | 10.86 % | 11.18 % |
| Matching best 3 cand. | 21.61 % | 20.18 % | 19.15 % | 20.62 % |
| Robust Pose Estim. | 17.08 % | 13.88 % | 9.06 % | 8.73 % |

Table 6.4: Average percentage of computational time spent in the individual parts of our algorithm on the *P565*.

matching), we compared it with an exhaustive search over the whole database, *i.e.*, all feature points contained in the reconstruction. The results of this evaluation are shown in Figure 6.8. Note that for convenience, this test was performed on a standard desktop computer.

It is easy to see that the matching performance for the image sequence acquired with the wide-angle lens is generally better than for the sequence acquired with the normal camera lens, except for very low resolution. The advantage of using the wide-angle lens clearly stands out and highlights the importance of a wide field of view for localization.

Note that the matching performance does not increase for resolutions above 640x480

pixels, and may even slightly drop for higher resolutions. Using a 3D reconstruction procedure, such as the one described previously in Section 6.3, the resulting database contains image features which constitute a coarse to moderately detailed representation of the environment. Image features which represent fine details of objects (*e.g.*, text on a wall poster) are usually not represented in the reconstruction. If high image resolutions are used for comparison, fine details may be extracted and can lead to increased mismatches. We partially overcome this by choosing a corresponding threshold for higher resolutions, but the effect is still noticeable. Another observable fact is that our preferred method of tree-based matching has only a ~ 5 -10 percent lower performance rate due to the errors in the matching stage.

Although we chose our limit for feature correspondences initially accepted by the robust pose estimator quite high, Figure 6.9 shows that almost 80 % of all inliers used for calculating the final pose for a 768x576 pixel image have a reprojection error smaller than 4 pixels. Thus we assume the resulting pose to be quite accurate, despite alignment errors introduced during manual global registration of individual reconstructions, and despite errors resulting from the triangulation step in a typical small-baseline indoor environment.

6.5.5 Full System Mobile Phone Evaluation

Finally we performed an evaluation of our localization method with the two smartphones, the *Meizu M8* (800MHz ARM11 CPU with FPU) and the *ASUS P565* (800MHz Intel XScale without FPU). On the *M8* we ran a version of our software using hardware floating point, whereas on the *P565* we ran a version using fixed-point math. We tested several different image resolutions for five test images, averaging the results. The database used is PVS 1 from our reconstruction, containing 40,670 features (equaling the same number of reconstructed 3D points). The results of our evaluation are listed in Table 6.3. A breakdown of the amount of computation time spent in the individual algorithms contained in our approach is given in Table 6.4.

As can be seen from row 2 of Table 6.3, the number of features increases almost linearly with the number of pixels of the query images and corresponds to the time spent in feature extraction. Interestingly, the CPU of the *M8* runs at the same rate as the CPU of the *P565*, but the performance of the *P565* is far better, which may be caused by caching or memory bandwidth limitations of the *M8*. Thus, for the rest of the discussion we focus on the results from the *P565*.

The amount of time spent in the histogram voting and the robust pose estimation stage only slightly increases, compared to the time spent in the feature extraction stage. The total amount of time for the whole process from feature extraction to robust pose estimation is about 248.59ms for a 512x384 pixel image, up to about 810.21ms for a

1024x768 pixel image. Comparing this result with the results from Figure 6.8, one can see that for a 768x576 pixel image a correct 6DOF pose (which is correct in about 82% of all cases) was computed at approximately $2fps$. Note that in this case the use of the wide-angle lens is assumed.

As can be seen in Table 6.4, the major amount of time is spent in the feature extraction stage. This is due to the computationally intensive task of scale-space search for extremal points and the subsequent generation of descriptors. The percentage of time spent on histogram voting for finding the best candidates and matching the descriptors stays almost constant across different image resolution. The time spent in the robust pose estimation stage drops due to the increased percentage of time spent in the feature detection stage. Moreover, our robust pose estimation method is able to discard a large number of outliers in a very early stage of the algorithm, thus keeping the overall amount of time needed relatively low (compare rows 5 and 9 in Table 6.3).

6.6 Conclusion and Outlook

In this paper we presented an approach for wide-area 6DOF pose estimation. 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. Our evaluations show that robust recovery of full 6DOF pose can be obtained on a current smartphone at about 2-3Hz, which is sufficient to initialize incremental tracking methods.

Due to the complexity of the localization task and the large number of different algorithms involved, we cannot discuss all aspects of our system in detail. In this work, we focus on the general feasibility of computing a localization efficiently with the proposed methods. An evaluation of the behavior of localization performance over large time periods would be an interesting research topic and will be addressed in future work. From our experience, large parts of the individual reconstructions do not change much over time, and localization worked well when reconstruction data acquisition was carried out several weeks earlier. Nevertheless there are also parts of the reconstructions which are likely to change, and thus more advanced methods are needed to overcome the resulting problems. For example, a bookshelf seems to be a good tracking target, but the ordering of books on the shelf changes frequently, as users take books and put them back at different positions. Note that this also happened during our evaluations and caused some localizations to fail. This will be subject of further investigations concerning the frequent updating of existing reconstructions or invalidating outdated parts of reconstructions.

In the future, we plan to combine the localization with an incremental tracker, working from the pose calculated with the approach proposed here. However, probably different types of datasets are necessary to make this feasible [Wagner et al., 2009]. In our current work, we did not include any investigations about special properties of cellular networks or wireless communication protocols necessary for online-sharing of 3D reconstructions from a centralized server. Nevertheless, this will be – together with a more in-depth investigation of PVS for efficient database management – subject to more closer research in the future as we plan to enlarge the reconstructed area at our department and think about deploying a test system for demonstration purposes.



Figure 6.6: Sample images from the test set collected along a path through three rooms of our department. On the left side, images taken with the normal lens are shown, on the right side pictures taken from the same position with the wide-angle lens are depicted.



Figure 6.7: Meizu M8 mobile phone front view, back view with magnetic mounting ring, wide angle camera lens from AmacroX, mounted camera lens (from upper left to lower right).

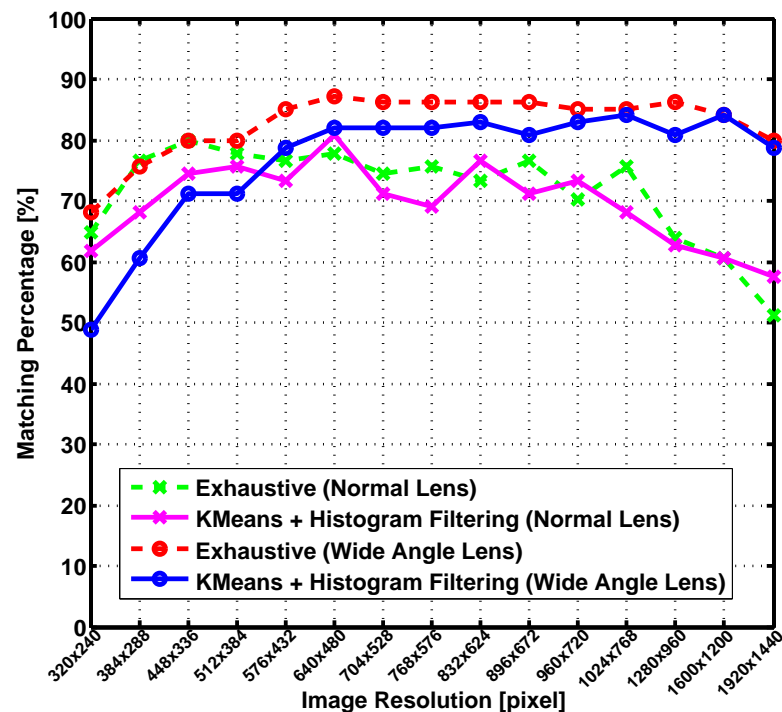


Figure 6.8: Image resolution vs. matching percentage for both lens types and both matching methods. The results for the wide-angle lens are drawn with dashed lines.

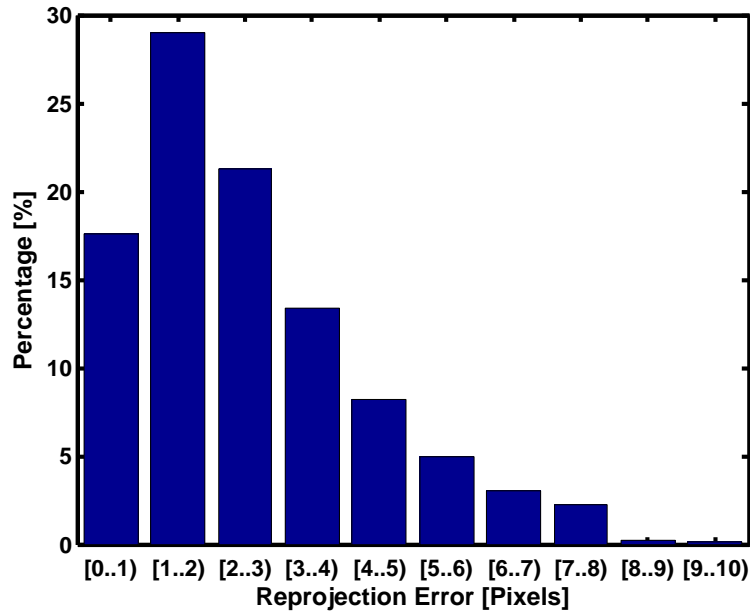


Figure 6.9: Reprojection error for inliers found during robust pose estimation. A reprojection error of ~ 10 pixels was allowed for an image size of 768×576 pixels.

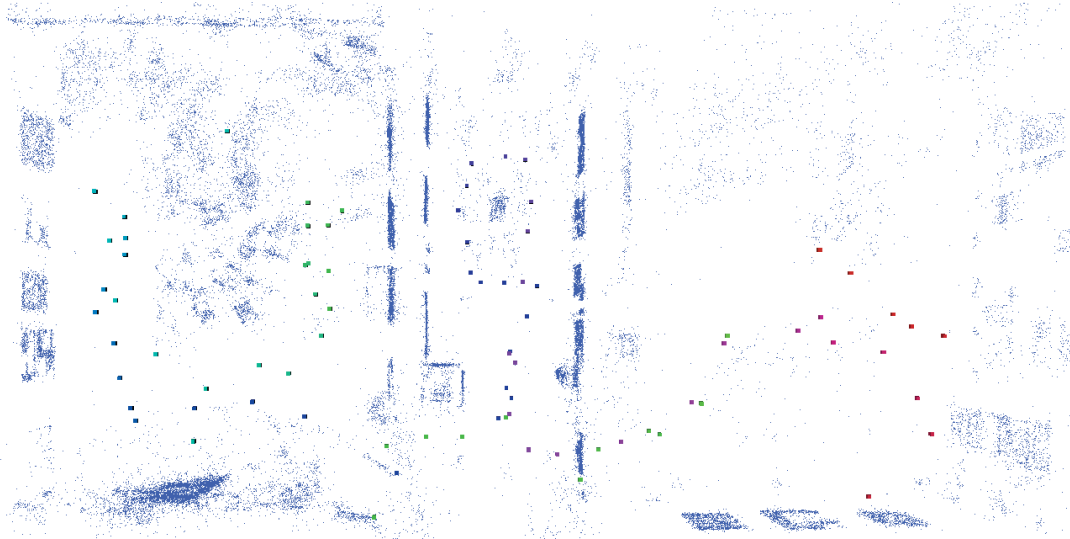


Figure 6.10: Three reconstructed rooms in which the set of test pictures was acquired. The camera poses estimated by our algorithm are represented as small, color-coded cubes from red via magenta and cyan to green. The path started in the right room (red), walking around in the middle room (magenta), going into the left room (cyan) and back into the right room (green).

Chapter 7

Robust Localization from Panoramic Images

The previous chapter demonstrated with an indoor localization scenario that by constraining feature sets with visibility memory requirements can be reduced and matching ambiguities can be reduced too. Outdoor localization is even more challenging than the indoor scenario for vision based 6DOF localization due to an ever-changing environment, difficult lightning conditions, crowd and vehicle movement, repetitive facades and so on. As also indicated in the last chapter, absolute pose based localization robustness and accuracy is strongly linked with the field of view of the observing camera. Cameras built into today's mobile devices are typically based around an aperture angle of 50° , which offers only a narrow view of the world. On the other hand real-time generation of panoramic images with these devices is well established. This gives us the opportunity to let the user contribute to the localization performance by increasing the field of view. Extending the field of view is again a general concept that does not depend directly on a particular feature detector, descriptor or matching technique.

7.1 Panorama Generation

High quality panorama generation is a well-known image stitching task. For a detailed overview of image stitching, the interested reader is referred to a comprehensive tutorial on this topic by Szeliski [Szeliski, 2005]. In most cases, the task of finding the geometrical relationship between individual images can be solved sufficiently well by determining image point correspondences, *e.g.*, by using the well-known SIFT algorithm, as also used in the popular Autostitch software [Brown and Lowe, 2003, Brown and Lowe, 2007, Lowe, 2004]. The majority of panorama creation methods are working on high-resolution still images and rely on significant amounts of computational and memory resources. Since



Figure 7.1: Localization result given the panoramic image shown at the bottom.

the actual process of framing individual images is still prone to camera artifact errors, these methods incorporate complex algorithms to remove seams and other visual artifacts.

Our panorama creation method works on the continuous image feed from a mobile phone camera and has to cope with the resources available on the device. Since our approach relies on an online mapping approach, only incremental techniques can be used. Many existing panorama stitching techniques, which rely on a complete set of source images for the creation of the panorama, are not applicable.

The panorama generator tracks relative orientation with 3DOF and simultaneously builds a cylindrical environment map. We are assuming that the user does not change position during panorama creation, *i.e.*, only a rotational movement is considered, while the camera stays in the center of the cylinder during the entire process of panoramic mapping (*c.f.* Figure 7.2). The first frame is mapped onto the cylinder surface to build the initial portion of the emerging panorama.

While the user is rotating the device, consecutive frames from the camera are processed. The algorithm computes the rotation between the current camera image and the already mapped panorama by extracting FAST corners in the live image. These features are matched with normalized cross-correlation against the tracking dataset taken from the partial panorama.

For each new frame the area in the panoramic view, which has not yet been mapped, is determined automatically. If this area is not empty, the panorama is extended with the new pixels. The map is subdivided into tiles of 64×64 pixels. Whenever a tile is entirely filled, the contained features are added to the tracking dataset. A closer description of the approach can be found in the work of Wagner *et al.* [Wagner et al., 2010].

7.2 Reconstruction and Global Registration

Rather than assuming that all reconstruction data is fully available when the reconstruction process starts, our technique supports the global registration of multiple partial reconstructions that were obtained separately. This enables a more realistic workflow of acquiring a large reconstruction model and maintaining it over time.

When building a global reconstruction from several individual reconstructions, they must all be aligned in a common global coordinate system. This could be done by using a fully automatic method as presented by Kaminsky *et al.* [Kaminsky et al., 2009], for instance. However, we chose to provide an initial rough alignment manually, and then let an algorithm refine it. Providing initial alignment can be done quickly with a suitable map tool, and prevents pathological errors resulting from too sparse image coverage and repetitive structures.

In order to refine the alignment of two reconstructions, we calculated matches for each



Figure 7.2: Panoramic Mapping of the environment onto a cylindrical surface. We use the term *angular aperture* to describe the FOV of a camera.

image in the first reconstruction to 3D features in the second reconstruction. From these matches, an initial pose estimate for the image in the first reconstruction with respect to the second reconstruction is obtained. The manual alignment is used to verify if the estimated pose is correct.

Using this approach, we can generate verified links between individual, initially not related reconstructions. We were able to improve the result of the manual alignment by using bundle-adjustment to reduce the reprojection error. A result of this approach is depicted in Figure 7.3.

7.2.1 Visibility Partitioning

Since feature database sizes grow with the covered area, it is necessary to partition the data to accommodate the storage limitations of mobile devices. We created blocks on a heuristically generated irregular grid to partition the reconstruction into smaller parts.

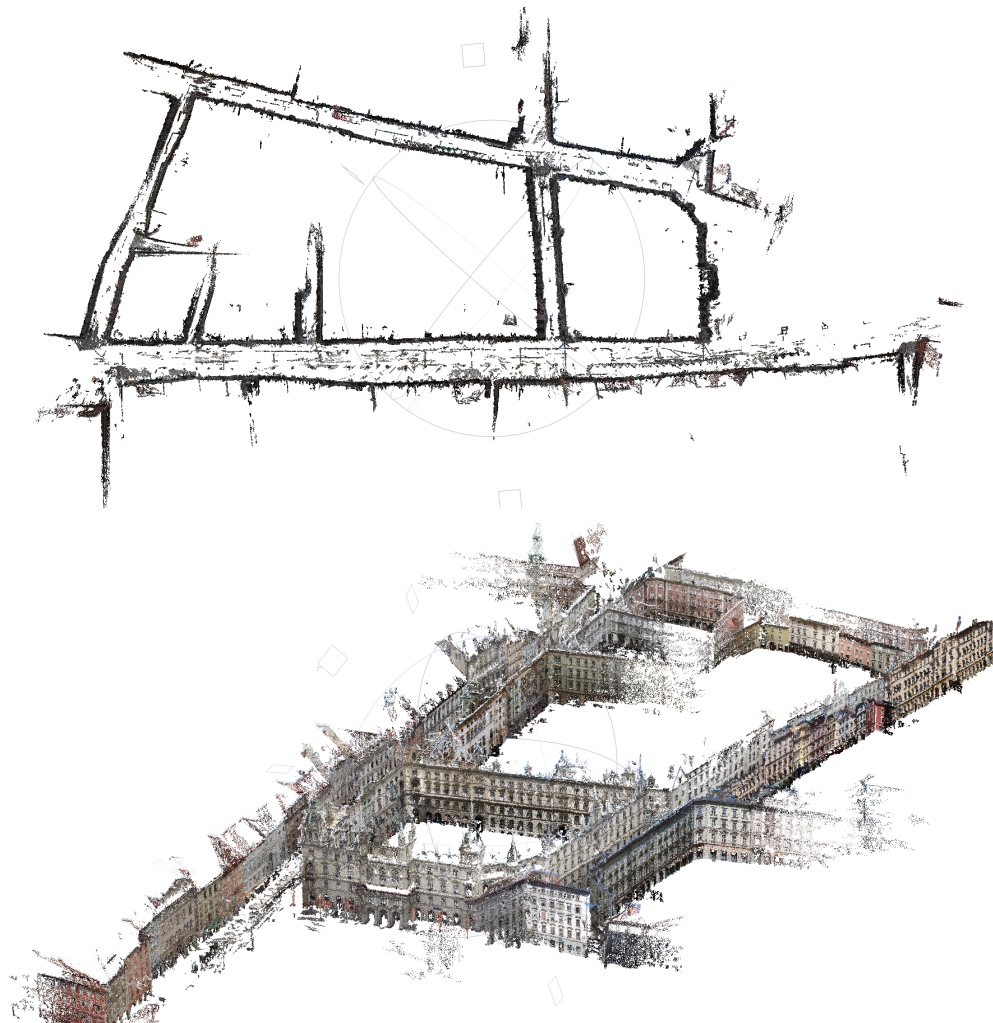


Figure 7.3: Sparse reconstruction data set. For better visualization a quasi-dense point cloud was created using [Furukawa et al.,].

Feature scale and estimated surface normal vectors could be added easily as additional visibility cues.

The partitioning of data blocks is on the one hand driven by visibility considerations and on the other hand by the accuracy of GPS receivers.

Most of the features in the database are generated from patches extracted from and therefore coplanar with building façades. These features can only be matched within a certain angular range¹. Its constraint is often violated when looking down a street and viewing façades at a steep angle. In this case, which is frequent in practice, only a small

¹ In general, this range depends on the capabilities of the feature detector. An angle smaller than $\pm 40^\circ$ seems reasonable in practice.

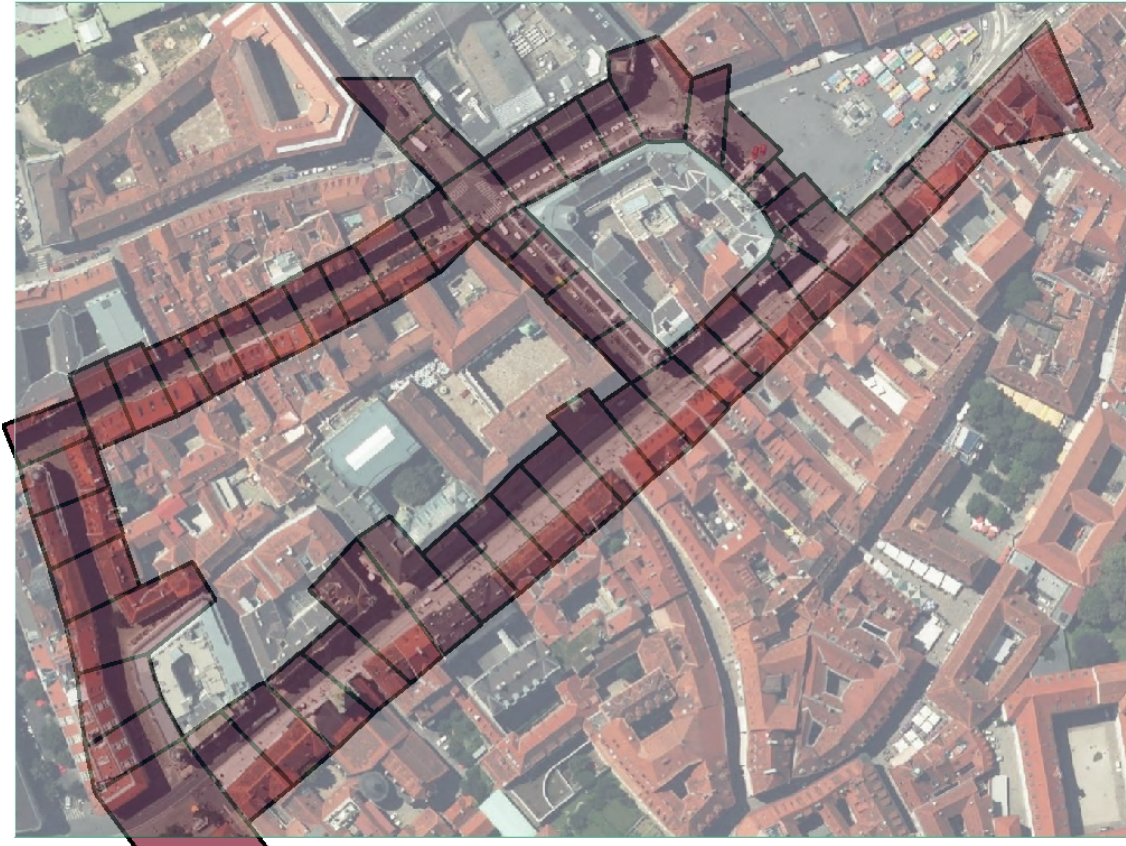


Figure 7.4: Partitioning of our feature database into individual feature blocks. Each block contains around 15000 features on average, which is around 2.2MB of memory.

area of the panorama that depicts the near streetside contains useful features, while further away features are not “visible” to the algorithm (*i.e.*, they cannot be reliably detected). We empirically determined a feature block size covering 20 meters of road direction and both sides of the road in order to yield best results.

An additional justification for this choice of block size is motivated by the accuracy of consumer-grade GPS receivers available in mobile devices. The accuracy of GPS estimates resides in the range of 10 to 20 meters. Given a GPS prior, the correct feature block can be determined easily. In order to avert inaccuracies, the neighboring blocks are considered as well. With this choice, the environment around an initial GPS-based position estimate is represented in a sufficiently reliable way for computing 6DOF localization.

7.3 Localization from Panoramic Images

For performing self-localization from panoramic images, some considerations are necessary which will be explained in the following. (Note that for the rest of the paper we will refer to the FOV also as the *angular aperture* in horizontal direction: we will use these two terms interchangeably.) In optics, the angular aperture has a slightly different meaning. For our purpose, however, we assume the use of a cylindrical model for panorama creation. In this context, the FOV of a panoramic camera directly corresponds to the arc of the cylinder circle.

The PVS indoor experiments indicated that increasing the field of view also increases the robustness of localization. One possible way to increase the field of view without modifying the device itself is by letting the user capture panoramic images. Figure 7.5 illustrates the basic idea of how we use panoramic images to increase the field of view for better image-based localization. A partial or complete panorama can be used for querying the feature database. Features extracted from the panoramic image are converted into rays and used directly as input for standard 3-point pose estimation. An alternative approach would be to use the unmodified source images that were utilized to create the panorama, and feature point tracks. These tracks can be converted to rays in space using the relative orientation of the images. However, we chose to work directly with the panoramic image for reasons of simplicity and lower storage requirements.

7.4 Experiments

In the following, we present evaluation results illustrating several aspects of our work. One goal of our investigation is to gain insight into the relationship between the camera's FOV and the resulting localization accuracy. Another goal is to demonstrate the accuracy of the method and its applicability for high-quality AR.

7.4.1 Localization Database and Panoramic Images

As the raw material for the localization database, we collected a large set of images from the city center of Graz, Austria. A Canon EOS 5D SLR camera with a 20mm wide-angle lens was used, and 4303 images were captured at a resolution of 15M pixels. By using the reconstruction pipeline described in Section 7.2, a sparse reconstruction containing 800K feature points of the façades of several adjacent streets was created. As natural features we used a scale-space based approach similar to SURF [Bay et al., 2006]. The entire reconstruction was registered manually with respect to a global geographic coordinate system, and partitioned into 55 separate feature blocks. These blocks were combined

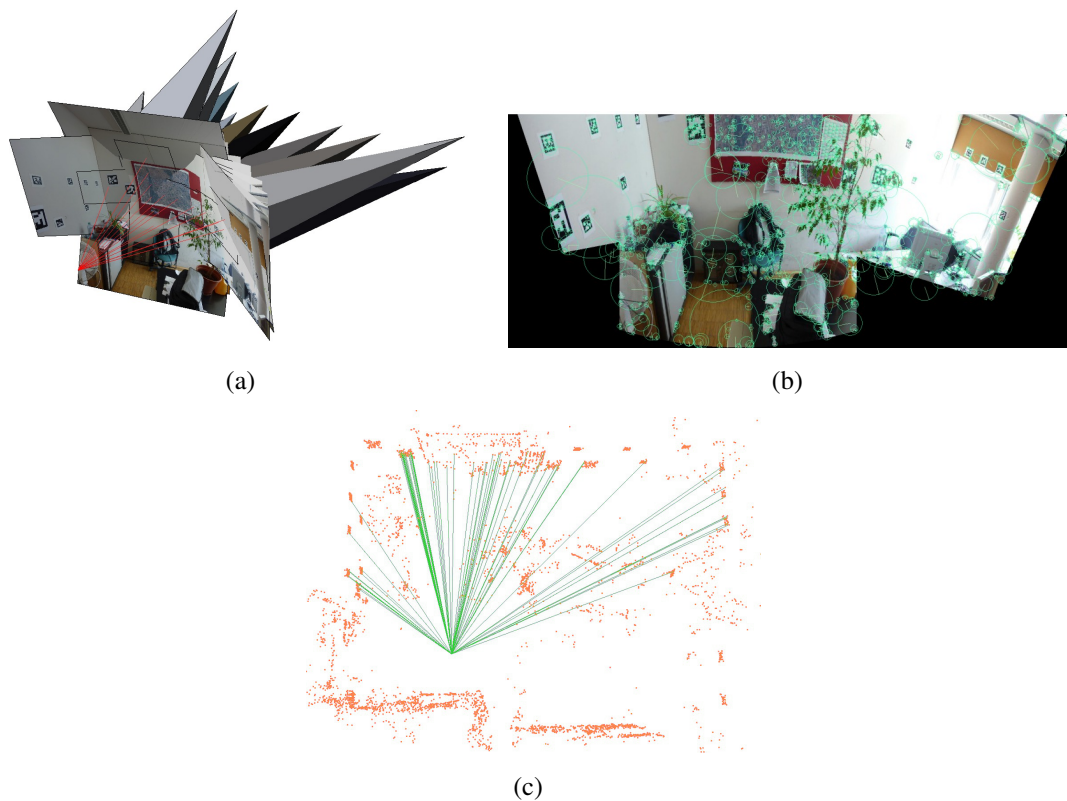


Figure 7.5: Extending the field of view illustration. (a) Relative orientation of images with the same projection center. (b) Feature points are extracted in the blended cylinder projection of the images. (c) Inlier feature matches for the panoramic image after robust pose estimation against a small office test database, the lines connect the center of projection with the matched database points.

again into 29 larger sections according to visibility considerations.

For studying our approach, we also created a set of reference panoramic images. We captured a set of 204 panoramas using a *Point Grey Ladybug 3* spherical camera. The images were captured along a walking path through the reconstructed area, and were resized to 2048x512 pixels to be compatible with our localization system. Note that we did not enforce any particular circumstances for the capturing of the reference panoramic images, they were rather resembling casual snapshots. The reference images and the images used for reconstruction were taken within a time period of about 6 weeks, while the imaging conditions were allowed to change slightly.

Since the spherical camera delivers ideal panoramic images, the results might not resemble realistic conditions for a user to capture a panorama. For this reason we additionally captured a set of 80 images using our panorama mapping application. These images

| Offline Reconstruction | |
|-------------------------------|---|
| Camera | Canon EOS 5D + 20mm wide angle lens |
| # of images | 4303 |
| image resolution | 5616x3744 pixels |
| # of sparse 3D points | 816,948 |
| total size of database | 122,769 kB |
| # of feature blocks | 55 |
| # of sections | 29 |
| Reference Images | |
| Camera | Point Grey Ladybug 3 |
| image resolution | 2048x512 pixels (resized for compatibility) |
| # of images | 204 |
| Test Images | |
| Camera Phone | Nokia N900 |
| image resolution | 2048x512 pixels |
| average aperture angle | $\sim 225^\circ$ |
| # of images | 80 |

Table 7.1: Details from our reconstruction of the Graz city center and the panoramic test images captured.

were taken about one year after the acquisition of both other datasets, while a significant amount of time had passed. The capturing conditions were almost the same, *i.e.* high noon and partially cloudy sky. The images expose a high amount of clutter and noise due to exposure changes of the camera¹. An important fact is that in almost all images only one side of the street could be mapped accurately. This results from the violated condition of pure rotation around the camera center during mapping.

Some details about the reconstruction and test images are summarized in Table 7.1.

7.4.2 Aperture Dependent Localization Performance

By using our panorama generation method, the handicap of the narrow FOV of mobile phone cameras can be managed. However, one remaining question is how the success of the localization procedure and the localization accuracy relates to the FOV of a camera in general.

We ran an exhaustive number of pose estimation tests given our set of panoramic images to measure the dependence of the localization success rate on the angular aperture. We modeled a varying FOV by choosing an arbitrary starting point along the horizontal

¹ Note that we simply used auto-exposure setting for acquisition.

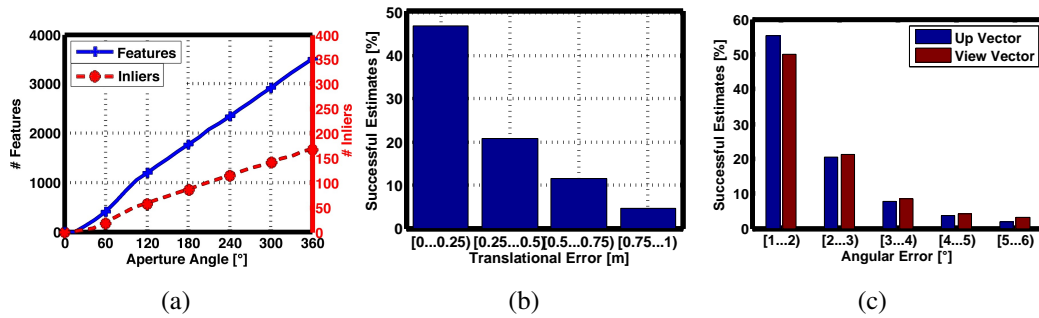


Figure 7.6: Localization Performance for a varying FOV. (a) The average number of inliers is approximately 5% of the features and increases linearly with the number of features detected in the entire image. (b) For around 84% of all successfully determined poses, the positional error is below 1m. (c) For around 88% of all successfully determined poses, the rotational error for both the view vector and the upvector is below 5 degrees.

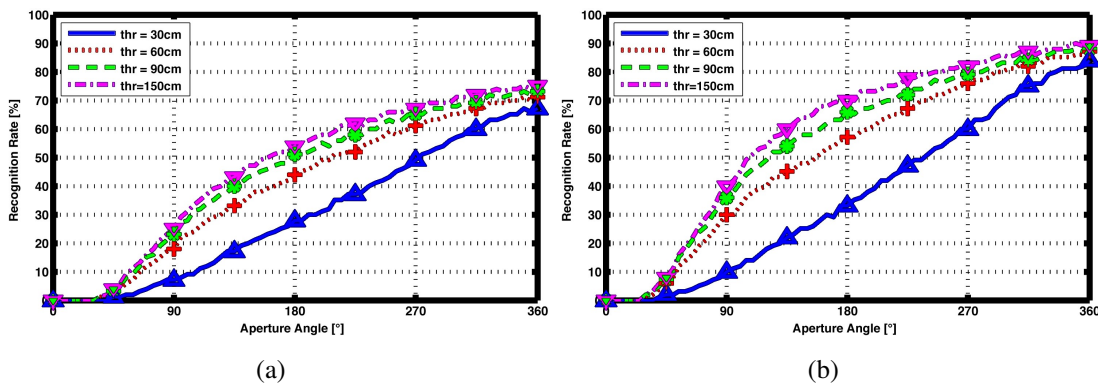


Figure 7.7: Success Rate for different thresholds applied on the translation error and random starting points. In (a) the tree-based matching is depicted, while in (b) the exhaustive matching based results are shown. The results of the tree-based matching are 5-10% worse than the results of the exhaustive matching approach.

axis in the panoramic image, and by limiting the actually visible area to a given slice on the panoramic cylinder around this starting point. In other words, only a small fraction of the panoramic image relating to a given FOV around the actual starting point is considered for pose estimation. The angular aperture was incrementally increased in steps of 5° from 30° to 360° .

We hypothesized that in urban scenarios, the localization procedure is likely to fail if the camera with a small FOV is pointing down a street at a steep angle. The same procedure is more likely to be successful if the camera is pointing towards a façade. Consequently, the choice of the starting point is crucial for the success or failure of the pose estimation procedure, especially for small angular apertures. To verify this assumption,

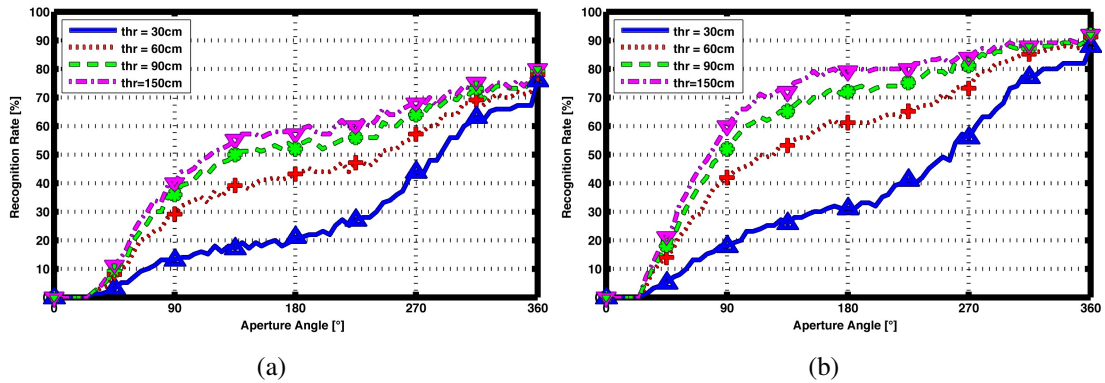


Figure 7.8: Success Rate for different thresholds applied on the translation error and manually selected starting points. The tree-based matching results are depicted in (a), the exhaustive matching based results are shown in (b). Compared to Figure 7.7, for small FOV higher success rates are achieved under a relaxed accuracy constraint.

we repeated the random starting point selection five times, leading to a total of 68,340 tests.

In Figure 7.6(a), the total number of inliers and features is shown. The number of inliers is approximately 5% of the number of features detected in the entire image. In Figure 7.6(b) and (c), the translational and rotational errors of all successful pose estimates are depicted. Due to the robustness of the approach it is unlikely that a wrong pose estimate is computed; in ill-conditioned cases the pose estimation cannot establish successful matches and fails entirely. As ground truth, we consider the pose estimate with the most inliers calculated from a full 360° panoramic image. The translational error lies in the range of several centimeters, while the rotational error is below 5° . This indicates that the pose estimate were highly accurate if it is successful.

The success rate of our localization procedure with respect to the angular aperture is depicted in Figure 7.7. We measured the localization performance considering different thresholds for the translation error to accept or reject a pose as being valid. In order to measure the difference between brute-force based feature matching and our tree-based matching approach, in Figure 7.7 (a) and (b) the results for both approaches are depicted. The tree-based approach has an approximately 5-10% lower success rate. Since building façades expose a high amount of redundancy, the tree-based matching is more likely to establish wrong correspondences. Thus, a lower success rate is reasonable. For a small threshold the performance is almost linearly dependent on the angular aperture. This is an important result since it proves that for solving the localization task the FOV should be as wide as possible, *i.e.* a full 360° panoramic image in the ideal case. An additional result is that for a small FOV and an arbitrarily chosen starting point (corresponding to

an arbitrarily camera snapshot), the localization procedure is only successful in a small number of cases. Even if the snapshot is chosen to contain parts of building façades, the localization approach is still likely to fail due to the relatively small number of matches and even smaller number of supporting inliers. With increasing aperture values all curves converge which is an indication that the pose estimates get increasingly accurate.

By now we only considered random starting points for capturing the panorama. However, a reasonable assumption is that the user starts capturing a panoramic snapshot while pointing the camera towards building façades intentionally, rather than somewhere else. Thus, we defined a set of starting points manually for all our reference images and conducted the previous experiment again. In Figure 7.8, the success rate for both matching approaches is depicted given different thresholds and our manually chosen starting points. For small aperture values the success rate is between 5 and 15% higher than for randomly chosen starting points, if the threshold on pose accuracy is relaxed (compared to Figure 7.7 (a) and (b) respectively). This result implies that successful pose estimates can be established more easily, but at the expense of a loss of accuracy. Since the features are not equally distributed in the panoramic image, the curves become saturated in the mid range of aperture values, mostly due to insufficient new information being added at these angles. For full 360° panoramic images, the results are identical to the ones achieved in the experiments before.

The entire path of panoramic images is shown in Figure 7.11, starting from the lower left area (red cylinders) and ending in the upper left area (cyan cylinders). In the upper right part the localization fails due to missing parts in the reconstruction. Localization again succeeds in areas where enough texture is visible (violet cylinders).

7.4.3 Pose Accuracy

For measuring the pose accuracy depending on the angular aperture, we ran a Monte-Carlo simulation on one sample panoramic image. Again, we simulated different angular apertures from 40° to 360° in steps of 5°. For each setting, we conducted 100,000 runs with random starting points, perturbing the set of image measurements with Gaussian noise of 2σ . This corresponds to a measurement error for features in horizontal and vertical direction of at most ± 5 pixels.

In Table 7.3, the resulting pose estimates are shown for different settings of the aperture angle. All poses are considered with a translational error of at most 1m. The camera pose uncertainty follows an unimodal distribution for uniformly distributed 3D points. In our real test environment 3D points are distributed more systematically. This is well reflected in our results, the centers follow a multimodal distribution. The pose estimates cluster visibly in multiple centers, increasing values of the aperture angle decreases the

variances around the centers. For a full panoramic image, all pose estimates converge into a single pose with minimal variance and the number of mixture components is reduced to one, i.e. an increasing FOV decreases the variance and the complexity of the pose uncertainty.

There are multiple reasons for this behavior. First, for a small FOV, only a small part of the environment is visible and can be used for pose estimation. A small field of view mainly affects the estimation of object distance, which, in turn, reduces the accuracy of the pose estimate in the depth dimension. A second reason for inaccurate results is that the actual view direction influences the quality of features used for estimating the pose, especially for a small FOV. Since the features are non-uniformly distributed for viewing directions towards façades, the estimation problem can be constrained better due to a higher number of matches. In contrast, for a camera pointing down a street at a steep angle, the number of features for pose estimation is considerably lower, and the pose estimation problem gets harder. Finally, due to the least squares formulation of the pose estimation algorithm, random noise present in the feature measurements gets less influential for increasing aperture angles. As a consequence, the pose estimates converge to multiple isolated positions. These images already cover large parts of the panoramic view (50-75% of the panorama). A single common estimate is maintained for full 360° panoramic images.

7.4.4 Runtime Estimation

To prove the usability of our approach on mobile devices, we took runtime measurements of the most important parts of our algorithm on a *Nokia N900* smartphone featuring an ARM Cortex A8 CPU with 600MHz and 1GB of RAM. The results were averaged over a localization run involving 10 different panoramic images. The results of this evaluation are given in Table 7.2.

The feature extraction process consumes the largest fraction of the overall runtime. Since the panoramic image is filled incrementally in an online run, the feature extraction process can be split up to run on small image patches (*i.e.*, the newly finished tile in the panorama caption process). Given a tile size of 64x64 pixels, the average time for feature extraction per tile is around 11.75 ms. As features are calculated incrementally, the time for feature matching is split up accordingly to around 0.92 ms per cell. To improve the accuracy of the pose estimate, the estimation procedure can be run multiple times as new matches are accumulated over time.

Given an input image size of 320x240 pixels and a tile size of 64x64 pixels, the estimated time for the first frame being mapped is around 230 ms. This results from the maximum number of tiles finished at once (15), plus the time for matching and pose es-

| Test Results | | Algorithm | Time [ms] |
|--------------------|------|------------------------|-----------------------|
| # of images | 10 | Feature extraction | 3201.1 (11.75 / tile) |
| avg. # of features | 3008 | Matching | 235.9 (0.92 / tile) |
| avg. # of matches | 160 | Robust pose estimation | 39.0 |
| avg. # of inliers | 76 | First frame (15 tiles) | < 230 |

Table 7.2: Results of our runtime estimation for different parts of our algorithm on the *Nokia N900* smartphone.

timination. The average time spent for localization throughout all following frames can be estimated similarly by considering the number of newly finished tiles. However, this amount of time remains in the range of a few milliseconds.

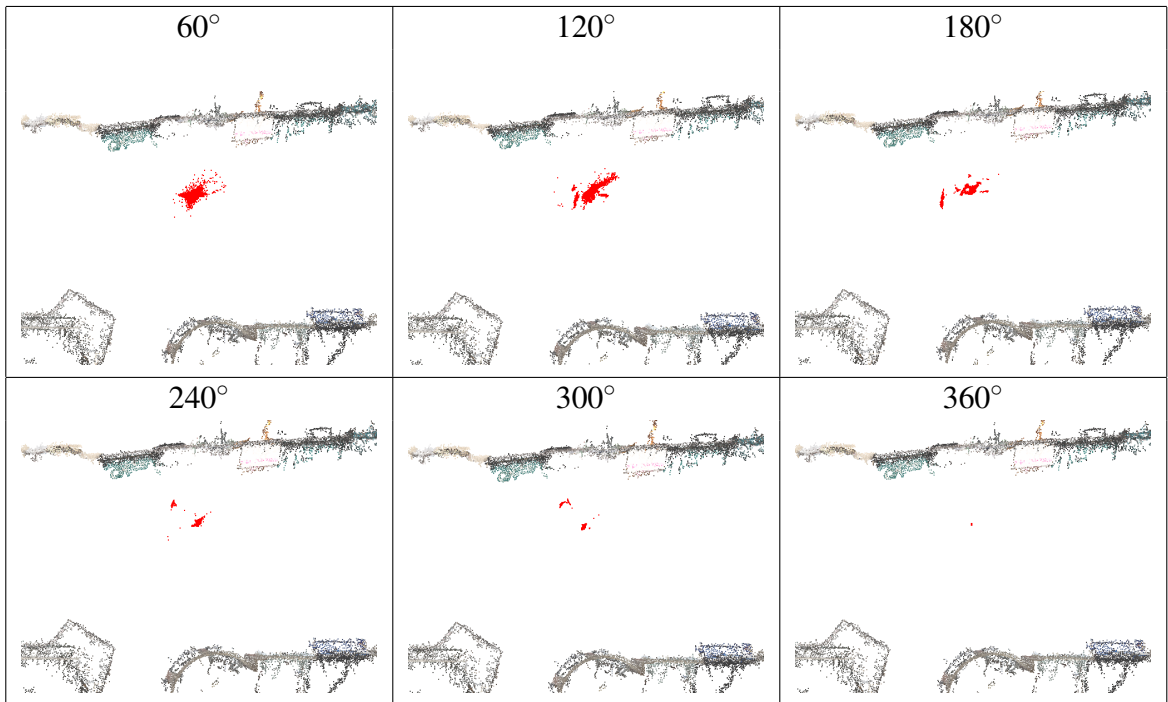


Table 7.3: Monte Carlo Localization Simulation Performance for a varying FOV. With an increasing aperture angle, the pose estimates converge into smaller clusters. Finally, for a full panoramic image all pose estimates converge to a single position.

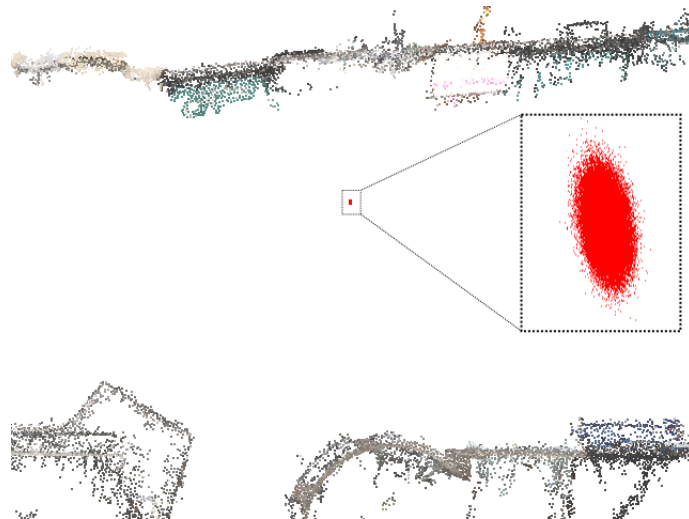


Figure 7.9: Zoomed in Monte Carlo Localization Simulation for FOV 360° . The influence of the unequal distribution of scene structure can be clearly seen. The pose is much better localized in directions parallel to the facades.

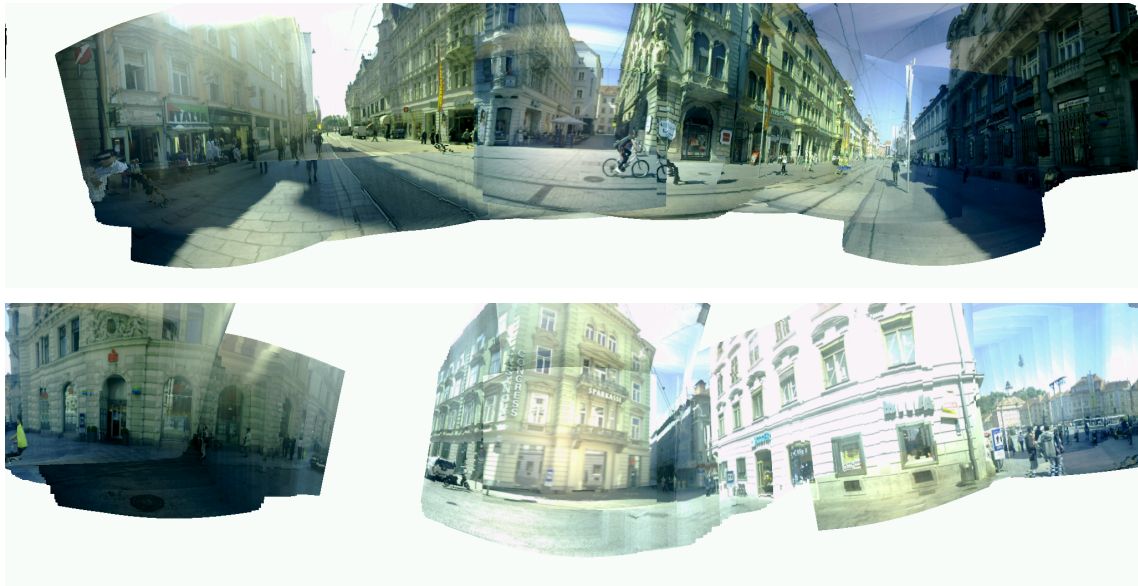


Figure 7.10: Panoramic images captured with our mapping approach. No exposure adjustment was used to increase the contrast or the visual appeal of the images.

7.4.5 Panoramas captured under Realistic Conditions

To test the performance of our algorithm on images captured under realistic conditions our localization approach was run on the second test set of 80 panoramas captured by our mapping application. Although a significant amount of time had passed between the initial reconstruction and the acquisition of the test dataset, using exhaustive feature matching our approach was successful in 51 out of 80 cases (63.75%). The tree-based matching approach was only successful in 22 of 80 cases (27.5%), however. A pose estimate was considered successful if the translational error was below $1m$ and the angular error was below 5° . These results mainly align with the results discussed in Section 7.4.2. The tree-based matching approach is more sensitive to changes of the environment and the increasing amount of noise respectively, which directly results in inferior performance.

We consider these results to be exceptionally good. We did not compensate for exposure changing artefacts in our panoramic images (see Figure 7.10). Given an aperture angle of $180 - 270^\circ$ the results correspond to our previous observations. The fact that the localization procedure still delivers reasonable results even a year after reconstruction is remarkable, however.

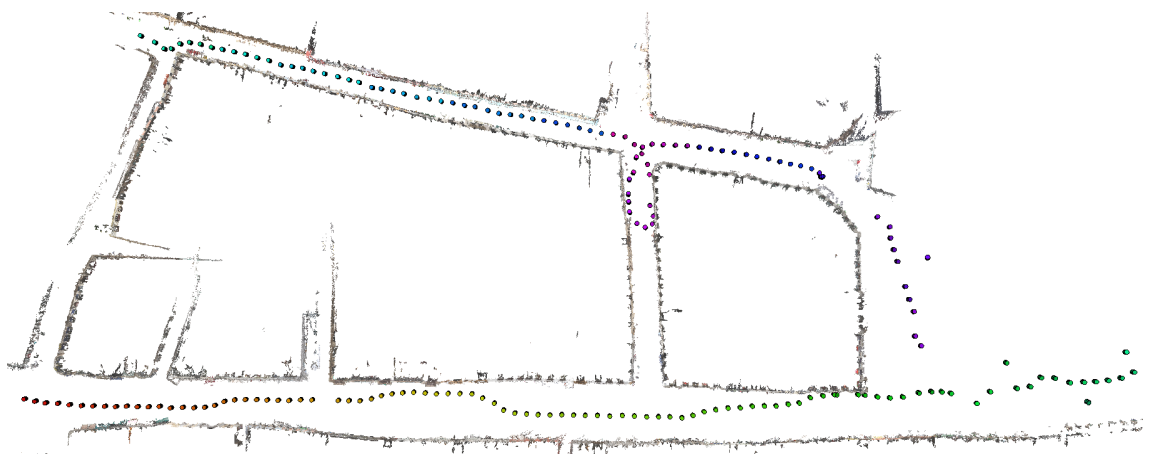


Figure 7.11: The path of panoramic images through the reconstruction. The camera positions are drawn as color coded cylinders, starting from the bottom left area and ending in the upper left area. Some localization results fail in the upper right area due to missing parts in our reconstruction.



Figure 7.12: Five sample snapshots from the augmented video sequence. After initialization, the panoramic preview on the bottom is removed to improve the visibility of the augmentations.

7.4.6 Live Augmentation

To demonstrate the capability of our system to field real-time AR experiences on current mobile phone hardware, we are presenting an application scenario involving two dwarfs running through the city of Graz. Snapshots from the video sequence accompanying this paper are depicted in Figure 7.12. The entire video is available as supplementary material.

The screenshots show that 3D models are accurately registered with the real world environment. Some minor errors are still visible, which are mainly caused by parallax effects. These effects result from the incorrect assumption of pure rotational movement around the center of projection. This assumption cannot be hold permanently all the time, so small errors become apparent especially for close-by objects.

7.4.7 Discussion

From our evaluation we concluded that the pose estimation algorithm gives highly accurate results in the case of successful completion. Giving details about the reprojection error, the traditional indicator for accuracy was omitted, since we considered it as an unsuitable measure in our case. The panoramic mapping procedure introduces a certain degree of discretization error, while the assumption of pure rotational movement during mapping creates another source of inaccuracy. The pose estimation procedure internally minimizes the distance between the projection (*i.e.*, mapping) of 3D points and the corresponding measurements. This has more algorithmic relevance than direct implications on the perceived quality of the augmentation.

The number of inliers for pose estimation lies in the range of 5-10% of the features matched, and in the range of about 1-2% of the entire number of features detected. On the one hand, this means that our pose estimation algorithm is very robust against outliers. On the other hand, the use of alternative, additional feature types should be considered to make the entire localization procedure more reliable.

The real-time applicability of our approach makes it highly suitable for the use on mobile phones. Although the size of the feature database as a whole is still prohibitive, holding the entire database on external storage on the device or downloading feature blocks occasionally, is a reasonable option. The smartphone used in our evaluation is a mid-range device in terms of computational power. More recent devices with fast multi-core CPUs are more likely to be capable of dealing with the computational demands of the method presented in this paper. Note that we did not investigate the use of exhaustive matching on the mobile device. Given a reasonable amount of a few thousand features, realistically brute-force matching cannot be applied under real-time constraints at present.

In this work, we do not include any information from non-visual sensors like compasses or gyroscopes. Obviously, the use of such sensors can help with multiple steps of our approach. For example, through compass information a guided matching scheme could be employed, pre-filtering features based on the current view direction and visibility constraints. An example of combining multiple heterogeneous sensors and visual tracking has for example been described in the work of Schall *et al.* [Schall et al., 2010], for instance.

7.5 Conclusion

We demonstrated a highly accurate outdoor localization system using panoramic images which can be used in real-time on current mobile devices. The most important characteristics of our approach are its high degree of accuracy and its low computational demands

that make it suitable to run on off-the-shelf mobile phones. We managed to overcome the problem of narrow FOV of current cameras on mobile devices by employing a novel technique for image capturing. Our approach can be used intuitively and in a very straight forward way, as the user is simply asked to capture the environment as a visually pleasing panoramic snapshot, and our approach allows for augmentations of considerably higher quality than possible with previous outdoor approaches.

Building a localization system with low computational demands and a high degree of robustness and accuracy is a challenging task. A significant unresolved issue is the acquisition and maintenance of the vast amounts of data needed for highly accurate reconstruction and subsequent localization. Databases have to contain information covering the visual variations for different times of the day and have to capture the appearance of the environment throughout the year. Building suitable and maintainable representations is an open issue for all localization tasks employing visual sensors.

Companies like Google or Microsoft tend to off-load the localization task to the *cloud*, mainly due to computational and maintenance reasons. However, we do not consider this as a reasonable option given the constraints of limited bandwidth and real-time operation.

For future work, we mainly consider the fusion of multiple heterogeneous sensors, such as recent micro-gyroscopes, to further enhance our localization approach. Moreover, we want to explore the use of more powerful mobile device hardware to make our approach more robust. Finally, upcoming stereo or depth cameras in mobile phones could be used to create a tight feedback loop between self-localization, image capturing, updating and maintaining the database representations used for the localization task. However, such a solution will be highly dependent on the characteristics of the actual sensor hardware.

Chapter 8

Conclusion

This thesis has addressed the 6DOF vision based AR localization problem from the data acquisition and localization perspective. We showed that image-based 6DOF localization and structure from motion are highly interconnected problems. The 3D data that is needed for an absolute pose computation is exactly what structure from motion methods compute. A robust incremental hierarchical structure from motion approach was used that allows to create large data sets for localization applications. Robust sequential image matching techniques were introduced that are suitable for our reconstruction problems. This makes it possible to detect robust overlaps of individual 3D reconstructions in ambiguous environments. A method to filter the epipolar matching graph was presented to enhance pair-wise image matching results. This makes structure from motion even more robust. Potentially visible sets were used to partition the SfM point cloud results more efficiently for the re-localization task. This idea was demonstrated in an indoor localization scenario. This idea was extended further with user interaction to enhance localization accuracy and robustness by letting the user capture panoramic images. Since panoramic images can be created online on mobile devices, this gives us the opportunity to let the user increase the field of view of the physical camera.

8.1 Outlook

Environment changes over time should be investigated more thoroughly. There are long term and short term aspects of a changing environment. Short time appearance changes like shadows cast by the moving sun can have equally dramatic effects on feature points as long term changes like seasons and changing window decorations in shopping windows. Especially for short term lighting changes more robust feature detectors would be interesting. Creating more generic visual words, for example by strong descriptor quantization would be an obvious approach. But such changes of feature description would still suffer

from repeatability problems of the detector itself.

Another relevant research topic is to reduce the complexity of the data acquisition step. The absolute pose algorithm based localization methods in this thesis need a fully triangulated and globally consistent SfM point cloud and therefore camera positions. These requirements can be relaxed for example to known camera positions without known orientations and a fully triangulated model by computing the query image position from a set of pair-wise relative orientations to images with known position. A suitable source of known positions is for example differential GPS. On the other hand, globally consistent point clouds make it easy to increase the field of view to enhance localization properties and use all available data transparently. This is not so straightforward when the 6DOF is computed from relative orientations.

Bibliography

- [Addlesee et al., 2001] Addlesee, M., Curwen, R. W., Hodges, S., Newman, J., Steggles, P., Ward, A., and Hopper, A. (2001). Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56.
- [Agarwala et al., 2006] Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., and Szeliski, R. (2006). Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 853–861, New York, NY, USA. ACM.
- [Agrawal et al., 2008] Agrawal, M., Konolige, K., and Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. In Forsyth, D. A., Torr, P. H. S., and Zisserman, A., editors, *ECCV (4)*, volume 5305, pages 102–115. Springer.
- [Airey et al., 1990] Airey, J. M., Rohlf, J. H., and Brooks, Jr., F. P. (1990). Towards image realism with interactive update rates in complex virtual building environments. In *Proc. Symposium on Interactive 3D Graphics*, pages 41–50, New York, NY, USA. ACM.
- [Akbarzadeh et. al, 2006] Akbarzadeh et. al, A. (2006). Towards urban 3d reconstruction from video. In *Proc. 3DPVT*, pages 1–8.
- [Baatz et al., 2010] Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., and Pollefeys, M. (2010). Handling urban location recognition as a 2d homothetic problem. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV, pages 266–279.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Proc. ECCV*, pages 404–417.

- [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 *DAGM*, pages 657–666.
- [Bibby and Reid, 2007] Bibby, C. and Reid, I. (2007). Simultaneous localisation and mapping in dynamic environments (slamde) with reversible data association. *Matrix*.
- [Brand et al., 2004a] Brand, M., Antone, M. E., and Teller, S. J. (2004a). Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In Pajdla, T. and Matas, J., editors, *ECCV*, volume 3022, pages 262–273.
- [Brand et al., 2004b] Brand, M., Antone, M. E., and Teller, S. J. (2004b). Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *ECCV*, pages 262–273.
- [Brown and Lowe, 2003] Brown, M. and Lowe, D. G. (2003). Recognising Panoramas. In *ICCV*.
- [Brown and Lowe, 2007] Brown, M. and Lowe, D. G. (2007). Automatic Panoramic Image Stitching using Invariant Features. 74(1):59–73.
- [C. Engels and Nister, 2006] C. Engels, H. S. and Nister, D. (2006). Bundle adjustment rules. In *Photogrammetric Computer Vision*.
- [C. Estrada and Tardós, 2005] C. Estrada, J. N. and Tardós, J. D. (2005). Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596.
- [C. Estrada and Tardós, 2009] C. Estrada, J. N. and Tardós, J. D. (2009). Finding good cycle constraints for large scale multi-robot slam. In *IEEE Int. Conf. Robotics and Automation*, pages 395–402, Kobe, Japan.
- [Chum and Matas, 2005] Chum, O. and Matas, J. (2005). Matching with PROSAC - progressive sample consensus. In Schmid, C., Soatto, S., and Tomasi, C., editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 220–226, Los Alamitos, USA. IEEE Computer Society.
- [Chum and Matas, 2008] Chum, O. and Matas, J. (2008). Optimal randomized ransac. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1472–1482.
- [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 *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*.
- [Davis, 2008] Davis, T. (2008). *User Guide for CHOLMOD: a sparse Cholesky factorization and modification package*.
- [Davison, 2003] Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proc. ICCV*, page 1403.
- [Davison et al., 2007] Davison, A. J., Reid, I., Molton, N., and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *TPAMI*, 29(6):1052–1067.
- [Eade and Drummond, 2006] Eade, E. and Drummond, T. (2006). Scalable monocular SLAM. In *Proc. 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 *BMVC*.
- [Engels et al., 2006] Engels, C., Stewénius, H., and Nistér, D. (2006). Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*. ISPRS.
- [Faugeras, 1993] Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA.
- [Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient belief propagation for early vision. In *Proc. CVPR*, pages 261–268.
- [Fischler and Bolles, 1981] Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Association for Computing Machinery*, 24(6):381–395.
- [Fitzgibbon and Zisserman, 1998] Fitzgibbon, A. W. and Zisserman, A. (1998). Automatic camera recovery for closed or open omage sequences. In *ECCV*, page I: 311.
- [Fritz et al., 2006] Fritz, G., Seifert, C., and Paletta, L. (2006). A mobile vision system for urban detection with informative local descriptors. page 30.
- [Furukawa et al.,] Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. Clustering Views for Multi-View Stereo. <http://grail.cs.washington.edu/software/cmvs>.
- [Gausemeier et al., 2003] Gausemeier, J., Fründ, J., Matysczok, C., Bruederlin, B., and Beier, D. (2003). Development of a real time image based object recognition method for mobile AR-devices. In *Afrigraph*, pages 133–139. ACM.

- [Govindu, 2001] Govindu, V. M. (2001). Combining two-view constraints for motion estimation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:218.
- [Govindu, 2004] Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:684–691.
- [Govindu, 2006] Govindu, V. M. (2006). Robustness in motion averaging. In *ACCV (2)*, pages 457–466.
- [Grabner et al., 2006] Grabner, M., Grabner, H., and Bischof, H. (2006). "fast approximated sift". In *ACCV*, pages 918–927.
- [Gupta and Hartley, 1997] Gupta, R. and Hartley, R. I. (1997). Linear pushbroom cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):963–975.
- [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 *Proc. CVPR*, pages 592–598.
- [Haralick et al., 1994] Haralick, R. M., Lee, C.-N., Ottenberg, K., and Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision*, 13:331–356.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). *A combined corner and edge detector*, page 50. Manchester, UK.
- [Hartley, 1997] Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593.
- [Hartley and Sturm, 1997] Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding: CVIU*, 68(2):146–157.
- [Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.
- [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 *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

- [Heikkilä and Silven, 1997] Heikkilä, J. and Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1106, Washington, DC, USA. IEEE Computer Society.
- [Hile and Borriello, 2007] Hile, H. and Borriello, G. (2007). Information overlay for camera phones in indoor environments. In *Location- and Context-Awareness, Third International Symposium, LoCA*, volume 4718, pages 68–84. Springer.
- [Hirschmüller, 2005] Hirschmüller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. CVPR*, pages 807–814.
- [Ho and Newman, 2007] Ho, K. L. and Newman, P. (2007). Detecting loop closure with scene sequences. *IJCV*, 74:261–286.
- [Horn et al., 1988] Horn, B., Hilden, H., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5:1127–1135.
- [Horn, 1989] Horn, B. K. P. (1989). Relative orientation. *Int. J. Comput. Vision*, 4:59–78.
- [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)*.
- [Irschara et al., 2009] Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition.
- [Jegou et al., 2007] Jegou, H., Harzallah, H., and Schmid, C. (2007). A contextual dissimilarity measure for accurate and efficient image search. In *Proc. CVPR*.
- [Kai Ni Steedly, 2007] Kai Ni Steedly, D. Dellaert, F. (2007). Out-of-core bundle adjustment for large-scale 3d reconstruction. In *Proc. ICCV*, pages 1–8.
- [Kalkusch et al., 2002] Kalkusch, M., Lidy, T., Reitmayr, G., Kaufmann, H., and Schmalstieg, D. (2002). Structured visual markers for indoor pathfinding.
- [Kaminsky et al., 2009] Kaminsky, R., Snavely, N., Seitz, S., and Szeliski, R. (2009). Alignment of 3D Point Clouds to Overhead Images. In *IEEE Workshop on Internet Vision (held in conjunction with CVPR)*, pages 63–70.
- [Kavitha et al., 2009] Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., Rizzi, R., Ueckerdt, T., and Zweig, K. A. (2009). Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review*, 3(4):199–243.

- [Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.
- [Klein and Murray, 2009] Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *Proc. Eighth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*, Orlando.
- [Kneip et al., 2011] Kneip, L., Scaramuzza, D., and Siegwart, S. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Kschischang et al., 2001] Kschischang, F., Frey, B., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519.
- [Kukelova et al., 2011] Kukelova, Z., Bujnak, M., and Pajdla, T. (2011). Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part II, ACCV'10*, pages 216–229, Berlin, Heidelberg. Springer-Verlag.
- [Lepetit et al., 2005] Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, pages II: 775–781.
- [Li and Hartley, 2006] Li, H. and Hartley, R. I. (2006). Five-point motion estimation made easy. In *ICPR*, pages 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 *Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV*, pages 427–440, Berlin, Heidelberg. Springer-Verlag.
- [Li et al., 2010] Li, Y., Snavely, N., and Huttenlocher, D. P. (2010). Location Recognition using Prioritized Feature Matching. *ECCV'10*, pages 791–804, Berlin, Heidelberg. Springer-Verlag.
- [Lindeberg, 1994] Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA.
- [Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

- [Longuet-Higgins, 1987] Longuet-Higgins, H. C. (1987). A computer algorithm for reconstructing a scene from two projections. pages 61–62.
- [Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA. IEEE Computer Society.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110.
- [Martinec and Pajdla, 2007] Martinec, D. and Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *Proc. CVPR*.
- [Mikolajczyk and Schmid, 2003] Mikolajczyk, K. and Schmid, C. (2003). A performance evaluation of local descriptors. *Proc. CVPR*, 2:257–263.
- [Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *IJCV*, 65(1/2):43–72.
- [Mooij, 2010] Mooij, J. M. (2010). libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173.
- [Mouragnon et al., 2006a] Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., and Dhome, M. (2006a). Real time localization and 3d reconstruction. In *Proc. CVPR*, pages 363–370.
- [Mouragnon et al., 2006b] Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006b). 3d reconstruction of complex structures with bundle adjustment: an incremental approach. In *Proc. ICRA*, pages 3055–3061.
- [Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09*, pages 331–340. INSTICC Press.
- [Naimark and Foxlin, 2002] Naimark, L. and Foxlin, E. (2002). Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. IEEE Computer Society.
- [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. pages 415–426.

- [Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Bio.*, 48(3):443–453.
- [Nistér, 2003] Nistér, D. (2003). Preemptive ransac for live structure and motion estimation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03*, pages 199–, Washington, DC, USA. IEEE Computer Society.
- [Nistér, 2004] Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:756–777.
- [Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Proc. CVPR*, pages 652–659.
- [Nister and Stewenius, 2006] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06*, pages 2161–2168, Washington, DC, USA. IEEE Computer Society.
- [Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.
- [Otsason et al., 2005] Otsason, V., Varshavsky, A., LaMarca, A., and de Lara, E. (2005). Accurate GSM indoor localization. In *UbiComp*, pages 141–158.
- [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. *IJCV*, 59(3):207–232.
- [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 *Proceedings of the 10th European Conference on Computer Vision: Part II, ECCV '08*, pages 500–513, Berlin, Heidelberg. Springer-Verlag.
- [Reitmayr and Drummond, 2006] Reitmayr, G. and Drummond, T. W. (2006). Going Out: Robust Model-Based Tracking for Outdoor Augmented Reality. pages 109–118.
- [Reitmayr and Drummond, 2007] Reitmayr, G. and Drummond, T. W. (2007). Initialisation for visual tracking in urban environments. In *ISMAR*, pages 1–9, Washington, DC, USA. IEEE Computer Society.
- [Robertson and Cipolla, 2004] Robertson, D. and Cipolla, R. (2004). An image-based system for urban navigation. pages 819–828.

- [Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443.
- [Schaffalitzky and Zisserman, 2002] Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, pages 414–431.
- [Schall et al., 2010] Schall, G., Mulloni, A., and Reitmayr, G. (2010). North-centred Orientation Tracking on Mobile Phones. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Seoul, South Korea,.
- [Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47(1-3):7–42.
- [Schindler et al., 2007a] Schindler, G., Brown, M., and Szeliski, R. (2007a). City-scale location recognition. In *Proc. CVPR*.
- [Schindler et al., 2007b] Schindler, G., Brown, M., and Szeliski, R. (2007b). City-scale location recognition. volume 0, pages 1–7, Los Alamitos, CA, USA. IEEE Computer Society.
- [Schmid et al., 2000] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, pages 151–172.
- [Shum et al., 1999] Shum, H.-Y., Zhang, Z., and Ke, Q. (1999). Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. *Proc. CVPR*, 2:2538.
- [Sim and Hartley, 2006] Sim, K. and Hartley, R. (2006). Removing outliers using the l_∞ norm. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 485–494, Washington, DC, USA. IEEE Computer Society.
- [Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477.
- [Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Bio.*, 147:195–197.

- [Snavely, 2008] Snavely, N. (2008). Scene reconstruction and visualization from internet photo collections.
- [Snavely et al., 2006] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA. ACM Press.
- [Snavely et al., 2008a] Snavely, N., Seitz, S. M., and Szeliski, R. (2008a). Skeletal graphs for efficient structure from motion. In *CVPR*.
- [Snavely et al., 2008b] Snavely, N., Seitz, S. M., and Szeliski, R. S. (2008b). Skeletal graphs for efficient structure from motion. In *CVPR*, pages 1–8.
- [Steedly et al., 2003] Steedly, D., Essa, I., and Delleart, F. (2003). Spectral partitioning for structure from motion. In *Proc. ICCV*, page 996, Washington, DC, USA. IEEE Computer Society.
- [Steele and Egbert, 2006] Steele, K. L. and Egbert, P. K. (2006). Minimum spanning tree pose estimation. In *Proc. 3DPVT*, pages 440–447.
- [Szeliski, 2005] Szeliski, R. (2005). Image Alignment and Stitching: A Tutorial. Technical report, MSR-TR-2004-92, Microsoft Research, 2004.
- [Takacs et al., 2008] Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.-C., Bismpiagiannis, 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 walkthroughs. *SIGGRAPH Comput. Graph.*, 25(4):61–70.
- [Triggs et al., 2000] Triggs, B., Mclauchlan, P. F., Hartley, R., and Fitzgibbon, A. W. (2000). Bundle adjustment—a modern synthesis. *Vision algorithms theory and practice*, 34099:298–372.
- [Verbiest and Gool, 2004] Verbiest, F. and Gool, L. V. (2004). Drift detection and removal for sequential structure from motion algorithms. *TPAMI*, 26(10):1249–1259. Member-Kurt Cornelis.
- [Wagner et al., 2010] Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. (2010). Real-Time Panoramic Mapping and Tracking on Mobile Phones. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 211–218.

- [Wagner et al., 2008] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008). Pose Tracking from Natural Features on Mobile Phones. pages 125–134.
- [Wagner et al., 2009] Wagner, D., Schmalstieg, D., and Bischof, H. (2009). Multiple target detection and tracking with guaranteed framerates on mobile phones. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando, FL, USA. IEEE Computer Society.
- [Wang et al., 2006] Wang, J., Zhai, S., and Canny, J. F. (2006). Camera phone based motion sensing: interaction techniques, applications and performance study. In *UIST*, pages 101–110. ACM.
- [Want et al., 1992] Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102.
- [Willson, 1994] Willson, R. (1994). Modeling and calibration of automated zoom lenses. In *Proceedings of the SPIE 2350: Videometrics III*, pages 170 – 186.
- [Willson and Shafer, 1994] Willson, R. and Shafer, S. (1994). What is the center of the image? *Journal of the Optical Society of America A*, 11(11):2946 – 2955.
- [Zach et al., 2008] Zach, C., Irschara, A., and Bischof, H. (2008). What can missing correspondences tell us about 3d structure and motion? In *CVPR*. IEEE Computer Society.
- [Zach and Pollefeys, 2010] Zach, C. and Pollefeys, M. (2010). Practical methods for convex multi-view reconstruction. In *Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10*, pages 354–367, Berlin, Heidelberg. Springer-Verlag.
- [Zhang and Kosecka, 2006a] Zhang, W. and Kosecka, J. (2006a). Image based localization in urban environments. pages 33–40, Washington, DC, USA. IEEE Computer Society.
- [Zhang and Kosecka, 2006b] Zhang, W. and Kosecka, J. (2006b). Image Based Localization in Urban Environments. *3DPVT '06*, pages 33–40, Washington, DC, USA. IEEE Computer Society.
- [Zhu et al., 2008] Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., and Sawhney, H. (2008). Real-time Global Localization with a Pre-built Visual Landmark Database. pages 1 –8.