DISS. ETH NO. 23157

Leveraging Geometric Priors and Measurements in 3D Modeling, Calibration and Registration

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Bernhard Zeisl

Dipl.-Ing. Univ., Technische Universität München

born 5. August 1982

citizen of Austria

accepted on the recommendation of

Prof. Marc Pollefeys
Prof. Konrad Schindler
Dr. Jürgen Sturm
Dr. Kevin Köser

2016

Abstract

Tow of the primary concepts in 3D computer vision are the reconstruction of 3D models and the subsequent estimation of the location and orientation of a camera therein. Due to a remarkable progress in the development of related algorithms in the last decade – for example, already modeling cities from images only – they nowadays form the basis of numerous and diverse applications, like digital map building for later virtual exploration, autonomous navigation, augmented reality, or the digital preservation of cultural heritage. However, algorithms are still error-prone in often occurring settings, *e.g.*, in texture-less scenes, in the presence of strong viewpoint changes between individual images, or they become computationally intractable for large-scale environments.

Therefore, this thesis examines approaches for 3D modeling and registration tasks and puts its focus on the exploitation of structural priors and available geometric measurements to achieve robust algorithms, while simultaneously aiming for computational speedup and improved precision.

In this regard, we consider computational stereo in texture-less environments, and bootstrap the extraction of a meaningful and visually pleasing geometry by enforcing a strong prior that favors vertical wall elements. The resulting parameterization allows us to transform the reconstruction problem into a dynamic programming optimization that computes regularized depth maps at interactive frame rates.

Since, RGB-D cameras can replace computational stereo in indoor settings nowadays, we consider such sensors and present a structure-based autocalibration method that jointly determines the extrinsic pose between the color camera and depth sensor, as well as the typically present distortions in the depth measurements. The obtained calibration allows for instant, accurate modeling without the need for any artificial calibration targets.

To efficiently compute a consistent model from individual RGB-D scans, automatic model registration is required. Especially terrestrial laser scans of-

ten exhibit strong viewpoint distortions, since the number of scan positions is minimized for reasons of efficiency. We illustrate how to leverage developable surfaces and salient directions, both extracted from the underlying scene geometry, to obtain rectified scene projections that enable discriminative feature matching and consequently fully automatic registration of scans with only limited overlap.

In a final step, we aim for the estimation of the camera pose with respect to a previously built reconstruction. For large-scale 3D models, *e.g.*, on the scale of a city, the 2D-3D correspondence search yields many tentative correspondences with a very low inlier ratio. We illustrate the use of simple, but efficient geometric filters to reject outliers and propose to formulate the camera pose estimation as a voting procedure. This results in a linear run-time, multi-modal pose estimates which are well suited to indicate repetitive structures, and an increased precision compared to state-of-the-art.

Zusammenfassung

Zwei der wichtigsten Konzepte im maschinellen, dreidimensionalen Bildverstehen sind die Rekonstruktion von 3D Modellen und die darauf folgende Schätzung der Position und Orientierung einer Kamera darin. Im letzten Jahrzehnt wurde ein bemerkenswerter Fortschritt in der Entwicklung von entsprechenden Algorithmen erzielt, z.B. erfolgte die 3D Modellierung von Städten ausschliesslich aus Bildmaterial. Deshalb bilden 3D Computer Vision Algorithmen heutzutage die Basis für unterschiedlichste Anwendungen, wie beispielsweise digitale Kartografie (inklusive der späteren virtuelle Erkundung der 3D Karte), autonome Navigation, Augmented Reality, oder auch die digitale Konservierung von Kulturerbe. Allerdings sind die zugrunde liegenden Algorithmen in regelmässig auftretenden Situationen weiterhin fehlerhaft, wie z.B. in wenig strukturierten Szenen, bei stark variierenden Ansichten zwischen einzelnen Bildern, oder werden zu rechenintensiv für grossäumige Umgebungen.

Aus diesem Grund untersucht diese Doktorarbeit neue Ansätze für die 3D Modellierung und Registrierung und setzt den Schwerpunkt auf die Behandlung und den Einbezug von bekannten strukturellen Beschränkungen und geometrischen Messungen mit dem Ziel der Entwicklung von robuster Algorithmen, bei gleichzeitig schnellerer Berechnungen und verbesserter Genauigkeit.

Zu Beginn betrachten wir die Tiefenschätzung aus Stereobildern in wenig strukturierten Umgebungen und ermöglichen die Extraktion einer aussagekräftigen und visuell ansprechenden 3D Geometrie mit Hilfe der Beschränkung auf vertikale Wandelemente. Die daraus resultierende Parameterisierung des Problems erlaubt es die Rekonstruktion als dynamische Programmierung aufzufassen, und ermöglicht die Berechnung von geglättete Tiefenkarten in Echtzeit.

Da heutzutage RGB-D Kameras die passive Stereoberechung in Innenräumen ersetzen können, widmen wir uns solchen Sensoren und präsentieren eine automatische struktur-basierte Kalibrationsmethode, welche die extrinsische Pose zwischen der Farbkamera und dem Tiefensensor sowie die typischen Verzerrungen in den Tiefenmessungen in einem Schritt ermittelt. Die erzielte

Kalibration ermöglicht eine sofortige und akkurate 3D Modellierung ohne auf künstliche Kalibrationsschablonen angewiesen zu sein.

Um ein konsistentes 3D Modell aus individuellen RGB-D Aufnahmen effizient zu berechnen, benötigt es einer automatischen Registrierung. Besonders bei der Arbeit mit terrestrischen Laser Scannern variieren die Standpunkten oft stark voneinander um die Anzahl der Messpositionen aus Effizienzgründen zu minimieren. In dieser Arbeit zeigen wir, wie abwickelbare Oberflächen und typisch auftretende Richtungen in der zugrunde liegenden Geometrie genutzt werden können um entzerrte, d.h. Standpunkt unabhängige, Szenenprojektionen zu erhalten, welche eine differenzierte Punktkorrespondenzbestimmung und infolgedessen eine vollkommen automatische Registrierung von Messungen mit nur geringer Überlappung ermöglichen.

In einem letzten Schritt zielen wir auf die Schätzung der Kamera Pose in Bezug auf eine zuvor berechnete Rekonstruktion. Für grosse 3D Modelle (z.B. in der Grösse einer gesamten Stadt) liefert die 2D-3D Punktkorrespondenzsuche viele mögliche Übereinstimmungen, allerdings nur mit einem geringen Prozentsatz an wirklich korrekten Korrespondenzen. Wir veranschaulichen die Verwendung von einfachen, jedoch effektiven geometrischen Filtern zur Detektion von falschen Korrespondenzen und formulieren die Schätzung der Kamerapose als Abstimmungsverfahren. Daraus resultiert eine lineare Laufzeit des Algorithmus, eine multimodale Posenschätzungen welche auf repetitive Szenenstrukturen anwendbar ist, sowie eine erhöhte Genauigkeit im Vergleich zu aktuellen Methoden.

Acknowledgments

This thesis would not have been possible without the support of numerous people – both in my professional as well as my private life. Thank you for all your continuous assistance, encouragement, critical feedback, and patience during the ups and downs involved in doing a PhD. This journey was not at all easy, but I am very glad that I never felt left alone.

I count myself lucky to have been given the great opportunity to work at the Institute for Visual Computing at ETH Zurich. My gratitude goes to my thesis advisor, Prof. Marc Pollefeys, for providing an exiting research environment, the freedom to explore ideas, and for his continuous trust in me. I am glad about our shared passion for skiing and will keep the yearly lab ski-trips all around Switzerland in fond memories.

I am deeply grateful to my (former) colleges and senior researchers, Christopher Zach, Kevin Köser, and Torsten Sattler at the Computer Vision and Geometry Group, whom I had the pleasure to work with. It is doubtless to say that their advice, explanations, ideas, and the many discussions we had, helped me the most, and that this thesis would not exist in its current form without these inspiring collaborations.

I would like to thank my long time office mate Jens Puwein for his companionship and friendship on our common journey. It was often him that helped me step back when computer vision did not work out as expected, for example during our regular "snack attacks" or exercise sessions. In the same manner, I am very thankful to my recent and former lab mates, in particular Olivier Saurer, Petri Tanskanen, Christian Häne, Aparna Taneja, Chris Sweeney, Amael Delaunoy, and Lubor Ladicky, that I shared numerous fruitful discussion with and who made research a joy – even during the crazy times when a deadline was around the corner.

Special thanks go to my co-examiners Prof. Konrad Schindler, and Jürgen Sturm for their interest in my work and the dedication of their precious time.

Already quite a time ago, but nothing short of importance, I was introduced to the field of computer vision by Pierre Fite-Georgel an Prof. Nassir Navab

at TU Munich, and Christian Leistner, Amir Saffari and Prof. Horst Bischof at TU Graz. Their excitement and vision for research has fascinated me and convinced me to follow a similar career.

Most of all, I would like to thank my parents for all their patience, understanding and love. It has been their continuous support in whatever matter that made it possible for me to follow and accomplish my dreams.

To all my friends, a thousands thanks for the much needed distraction, the fun times, and their assurance, that after all a PhD is just, and nothing more, than a PhD.

Last but not least, I am most thankful to Julia, who certainly has been affected the most by my research efforts during these last five years. Thank you for your love, your indispensable support and encouragement, and for being with me in this demanding time. I has been wonderful to have you by my side.

Zürich, 14. October 2015 Bernhard Zeisl

Contents

1 Introduction

The development of cheap digital cameras – nowadays available in mass market products such as mobile phones – has led to a wide-spread use and a large amount of image data available at hand. Not only for consumer products but also in industry their robustness and low price has motivated the usage of cameras in products, often for replacement of other sensors. As a result a high demand for systems and algorithms developed, which can make use of the present image data and analyze, evaluate, process and reason from it. In fact, over the last 50 years computer vision has majored from a (completely underestimated)"summer vision project" at MIT (Papert, 1966) to a dominant research direction that significantly is and will be shaping our future.

Across different industries, researchers, start-ups and established companies are currently working with ever increasing efforts to simplify our interaction with machines, improve our every-day life, and radically change the way how we access information by making computers "see" and understand our environment. Examples are as diverse as digital maps and 3D representation of the entire world like Google Maps and Street View (or similar products from competitors), the digital preservation of cultural and architectural heritage, vehicle safety systems, or gesture control for video games. Pioneering work in the area of autonomous driving requires accurate 3D vision and fault tolerant higher level scene interpretation to safely maneuver a car through our dynamic environments. Though, it doe not need expert systems like an upgraded car to experience the advance of 3D computer vision – already everyday consumer smart phones exhibit the ability to provide instant localization based on vision solely. In addition, we are currently witnessing the spread of augmented and virtual reality applications for tasks in our daily life, not to mention all the different advances in medical applications due to elaborate imaging algorithms.

At the bottom of all these applications are two fundamental tasks in 3D vision. First, there arises the need for the *reconstruction of 3D models* of real-world objects or scenes. Different approaches to 3D modeling have been

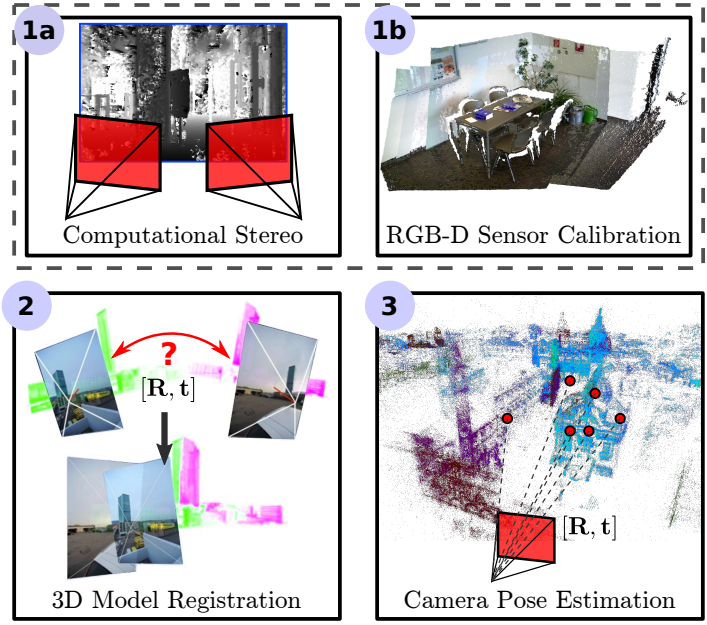


Figure 1.1: Overview of the 3D reconstruction, calibration and registration process considered in this thesis. 3D models are computed via computational stereo (1a) or captured with a (simultaneously) calibrated RGB-D sensor (1b). The registration (2) of these partial models results in a consistent reconstruction, with can then be used for image-based localization (3).

a very active and long-standing goal in computer vision and over the last decade tremendous progress has led to impressive results, even modeling whole cities from images only in a fully automated manner. Second, given a 3D model, we are often tasked with *estimating the 3D location and orientation of a camera* with respect to the model itself. Resulting image-based localization and tracking algorithms are essential for all kinds of navigation tasks, in augmented reality or also for the spatial ordering of image collections. Often the tracking of the camera pose needs to be performed in a large-scale setting (*e.g.*, on the scale of a whole city) and robust estimation methods are essential.

This thesis examines the consecutive steps typically present in such a 3D

reconstruction and localization system. As illustrated in Fig 1.1, the process starts with the initial estimate of scene depth from images or the registration of joint image and depth measurements, then considers the alignment of individual scans within the context of 3D model fusion, and finally leverages a previously built model for image-based localization. For each of these steps we will focus on the exploitation of a priori known structural constraints or already available geometric measurements to achieve computational speedup and improved accuracy, or automatic computation in cases where manual interaction has been required so far. The particular problems and challenges arising in these tasks – and that we aim to solve – are motivated in the following.

Stereo reconstruction: Dense 3D reconstruction becomes very challenging, if active sensors are not available and the 3D virtual model has to be generated solely from image data. Man-made, and in particular indoor environments are very demanding for automated image-based 3D modeling, as they are typically comprised of few visually salient objects and tend to have only weak textures. Sophisticated methods for computational stereo can overcome some of these difficulties, but those methods come at a high computation cost and are therefore less suitable, *e.g.*, for obstacle avoidance and on-line navigation of autonomous systems. In order to obtain a sufficiently accurate, but computationally cheap 3D map of the environment, suitable prior knowledge on the encountered surrounding, such as expected object shapes, is necessary.

3D modeling with RGB-D sensors: In recent years active depth sensing devices, such as Mircosoft Kinect or time of flight sensors, have gained tremendous popularity in the robotics and computer vision community, due to their ability to deliver densely sampled, accurate distance measurements in real-time and with low computational cost. Thus, they recommend themselves for 3D modeling and to replace passive stereo systems for close range sensing in indoor scenarios. In particular the combination of a high quality color camera – that is a standard on mobile devices nowadays – and a depth sensor already has and will continue to enable a plenitude of new and exciting applications, *e.g.*, accurate real-time 3D modeling and localization, interactive 3D gaming, tele-presence, or all kinds of machine learning driven reasoning and analysis. However, the different sensor modalities (denoted as RGB-D in this thesis) request for extrinsic and intrinsic calibration such that their full potential can

be exploited. For example, textured 3D models, correctly colored point clouds as well as features capturing both modalities are only possible by means of a correct calibration, *i.e.*, that (i) the mapping for corresponding image and depth pixels is known and (ii) the depth measurements represent undistorted, real world distances. In comparison, computational stereo methods naturally fulfill these requirements[1]. Typical calibration approaches makes use of artificial landmarks and special calibration patterns. However, the usage of calibration targets is a tedious process and will often not be applicable, *e.g.*, if on-line (re)-calibration is necessary or the sensor setup is inaccessible as it is the case for already captured datasets. Consequently, there is strong need for auto-calibration where the observed geometry is exploited instead of artificial targets.

Partial model registration: Once image and depth data has been captured, the registration of individual 2.5D scans constitutes a core challenge and aims to estimate the relative pose between the different model parts. Due to their high frame rates, consecutive measurements from commodity RGB-D sensors are relatively easy to relate to each other; however, the estimated visual odometry drifts over time, and thus requests for the detection and closing of loops in the camera path. Typically this requires pose estimation between model parts from significantly different view points. Moreover, in industrial applications laser scanning is the state-of-the-art technique to obtain accurate three-dimensional models. Usually a scanner is positioned at different places in order to minimize scan shadows and to obtain a model as complete as possible. Since scanning is a time-consuming and therefore expensive task the number of scans is kept as small as possible, again leading to a wide baseline setting between the scan positions. As a result, there is a need for automatic registration methods which do not rely on any artificial landmarks, but can generate accurate, wide-baseline registrations by exploiting the model data itself.

Image-based localization: Finally, let us consider the case where a 3D model is already present and our goal is to estimate the camera pose of a single query image wrt. the model. Image localization is typically performed by first establishing 2D-3D correspondences between 2D image observations and 3D

[1] Absolute scale is available if the baseline between images is known.

points and the consecutive application of a minimal pose solver combined with geometric verification via hypothesis evaluation from a minimal set of random samples (cf. RANSAC, Fischler and Bolles, 1981). Since the run-time of RANSAC has at least cubic growth[2] in the number of matches it is absolutely crucial to avoid finding and using too many wrong matches. At the same time, distinguishing between correct and incorrect correspondences only based on their local appearance is an ill-posed problem for large datasets, as they contain many visually similar points. This is especially true for urban scenes which often possess repetitive structures. Elaborate matching procedures together with guided correspondence search are able to overcome some of the difficulties; however, they still rely predominantly on filtering within the descriptor space and tend to reject too many correct matches. In order to achieve accurate localization in large-scale scenarios and in reasonable time, fast and efficient outlier filtering is necessary. Thereby, spatial relations between correspondences can remedy the problem that visual discriminative power often has reached its limits.

1.1 Problem Statement

3D reconstruction and registration algorithms in the spirit of aforementioned applications are already present in several products and automatically solved in increasing numbers on a daily basis. Algorithms often run on dedicated and constraint hardware, which renders their efficiency and computational complexity an important topic. At the same time, their suitability is dependent on the obtained solution quality and their robustness to failure cases. We address these subject by asking:

> *How can we decrease computation time while possibly simultaneously increasing the accuracy of 3D reconstruction or pose estimation?*

Admittedly, this is hard to achieve in general. Though, in many cases the particular application setting is known a priori, *e.g.*, in- or outdoors, man-made or natural environments, *etc.* Thus, we may well wonder:

> *Given knowledge about the (local) 3D geometry, what are the implications for the considered vision algorithms?*

[2] A 6 DoF pose solver requires 3 samples for a calibrated camera and more for (partially) uncalibrated cameras.

Especially for stereo reconstruction, available prior information on the observed geometry will help to constrain the solution space, addressed within the thesis:

What are appropriate geometric priors for efficient 3D modeling of textureless indoor environments?

Structure-from-motion is known to provide accurate triangulations for 3D points with many image observations. A corresponding sparse point cloud can be computed well from the image data captured by a RGB-D sensor, and the computed geometry should conform with the sensor's (calibrated) depth maps. Therefore, the natural question arises:

Can we leverage a sparse reconstruction for structure-based auto-calibration of a RGB-D sensor such that we obtain an accurate registration and 3D reconstruction?

The accuracy of registration results is known to depend primarily on the quality of established point correspondences. As a consequence, we focus on increasing the number of correct matches while limiting the number of outliers. As motivated before, the distinctiveness of visual features degrades under strong view point distortions and with the absolute number of possible candidate matches. Therefore, we are interested in eliminating perspective effects and efficient outlier filtering. This idea is subsumed in this thesis by the following question:

Do 3D-3D registration methods lend itself for joint exploitation of images and measured geometry information, and if so, can we thereby make the alignment process more robust?

Finally, large-scale models pose a particular challenge solely due to their sheer size of feasible candidate matches. In contrast to the current practice of aggressive outlier filtering during feature matching, we aim to consider 1-to-many matches in pose estimation and thus phrase the question:

How can we make geometric verification scalable to thousands of tentative correspondences?

Similar to before, our idea is to exploit the measured geometry; thus, we also aim to answer:

Which constraints can be extracted from the model and are applicable for spatial verification, and what do we gain from it?

1.2 Contributions and Thesis Outline

After we introduce notations, algorithms and conceptual foundations together with corresponding relevant related work in Chapter 2, we subsequently present our contributions. Motivated by the previously presented research questions, it is the aim of this thesis to develop extensions for existing algorithms in 3D modeling and pose estimation or identify alternative solution to overcome existing limitations. In particular we suggest the following approaches:

- We start with the classical 2-view case in Chapter 3 and consider computational stereo in texture-less environments. We incorporate the assumption that the scene predominantly consists of vertical wall elements located between a floor and ceiling plane, which is a setup that is often found in indoor environments. This particular configuration corresponds to a tiered labeling and we show how it allows us to transform the reconstruction task to a dynamic programming problem that is efficiently solved.
- In Chapter 4 we then account for the recent success of commodity RGB-D sensors. To jointly exploit image data and depth measurements from these sensors, or add-hoc camera-depth sensor setups, we present an auto-calibration method. Instead of using specific calibration targets, we propose to leverage the environment structure as the geometric prior for calibration. Structure-from-motion (or equivalently SLAM) can provide such a 3D scene model, and hence our approach allows for self-calibration without the need for any manual interaction. Obtained results demonstrate that we are able to compute an accurate calibration, which for example allows dense 3D modeling at improved precision.
- Next, we turn to the problem of wide-baseline registration of RGB-D scans in Chapter 5 and illustrate examples for both, scans from commodity sensors as well as high quality laser scans. For the alignment of several RGB-D scans in wide-baseline scenarios we rely on image based features, because they are plenty, localized well, repeatable among different views and much more discriminative than depth features. However, viewpoint distortions request for normalization and we propose to utilize the observed scene geometry for rectification. We show how the concept of developable surfaces allows to unfold textures for more discriminative matching between RGB-D scenes. In addition we present a fully automatic approach for the alignment of RGB-D scans with high viewpoint variations and only limited overlap.

Thereby, we rely on salient directions extracted from the scene geometry to generate orthographic scene projections that only differ by an euclidean transformation. As a result, our approach enables the generation of accurate model registrations in cases where other methods fail.

– Augmentation of a previously build 3D model with new image data requires to determine the camera pose. In large-scale scenarios the inlier ratio for tentative 2D-3D correspondences easily drops below 1% due to the increase of visual similarities in the model. In Chapter 6 we show that spatial verification can still be achieved via the extensive consideration of present geometric constraints between the camera setup and the 3D model. We formulate the camera localization as a voting procedure in the pose parameters space and by this achieve a linear run-time in the number of matches. As a result, our approach allows to evaluate 1-to-many correspondences – which are considerably more matches compared to other methods – and hence it is applicable for scenes containing repetitive structures. We demonstrate that our approach surpasses state-of-the-art on one of the currently most challenging datasets for camera pose estimation.

Finally, Chapter 7 summarizes our work and concludes with a discussion regarding topics for future work.

2 Foundations and Related Work

Several core concepts of computer vision, such as models, algorithms and specific solutions form the foundation for this thesis. These include camera models, appearance and geometry based image properties, correspondence search, registration problems and their solutions for pose estimation, and dense and sparse 3D reconstruction algorithms. In this chapter we will discuss them briefly, starting from the basic pinhole camera imaging process and gradually evolve to the construction and processing of full 3D models. For each concept we provide a broad overview of relevant work, while a more detailed discussion of related work appropriate to the particular application is then given in the following Chapters 3-6.

2.1 Notation

The mathematical notations are consistent throughout the thesis. Italic characters denote scalar values, while bold characters are used to represent vectors. Uppercase roman fonts are used for matrices.

To differentiate between 2D and 3D points, the former are written in lowercase while the latter appear uppercase. If not specified differently, the individual coordinates of points are $\mathbf{x} = (x, y)$ in the image domain and $\mathbf{X} = (X, Y, Z)$ for 3D points. In addition, points in $\mathbb{R}^2$ and $\mathbb{R}^3$ can be represented as homogeneous coordinates (Hartley and Zisserman, 2004, $p27$) by adding an additional dimension. They are indicated by an accent according to $\tilde{\mathbf{x}} = (w\mathbf{x}, w)^\intercal$ and $\tilde{\mathbf{X}} = (w\mathbf{X}, w)^\intercal$ with $w \in \mathbb{R}_{\setminus 0}$.

Finally a reference to an entire image or depth map is given as I and D, respectively.

2.2 Camera Model

All computer vision research starts by first taking an image; thus, we begin by discussing the imaging process. In general a camera performs a mapping between 3D world points and 2D locations in an image (Hartley and Zisserman, 2004). We will only cover the central projection and affine camera model, for generalized cameras the interested reader is referred to (Pless, 2003; Sturm, 2005; Lee et al., 2013).

While 3D points are represented in the world coordinate system, each camera has its own local camera coordinate system, where the z-axis is in direction of the optical axis. The transformation of a world point $\mathbf{X}_W$ into camera coordinates $\mathbf{X}_C$ is modeled by a rotation $\mathrm{R} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t} \in \mathbb{R}^3$ according to

$$\mathbf{X}_C = \mathrm{R}\mathbf{X} + \mathbf{t} = \begin{bmatrix}\mathrm{R} & \mathbf{t}\end{bmatrix} \tilde{\mathbf{X}}_W \ . \tag{2.1}$$

The expression $[\mathrm{R}, \mathbf{t}]$ is also referred to as the pose of a camera. The actual imaging process differs by the type of projection as follows.

Perspective Projection: A perspective camera, also known as the pinhole camera, has a single center of projection $\mathbf{C}$. In camera coordinates this coincides with the coordinate system origin $\mathbf{0}$, while in world coordinates it depends on the actual camera pose and computes to $\mathbf{C} = -\mathrm{R}^\mathrm{T}\mathbf{t}$.

If 3D points are represented as homogeneous coordinates, then a central projection corresponds to a simple linear mapping by means of the canonical projection matrix $\mathrm{P}_\mathrm{P} = [\mathrm{I} \quad \mathbf{0}]$. P_P effectively maps a point $\tilde{\mathbf{X}}_C$ to the homogeneous point $\tilde{\mathbf{x}}_n$; it corresponds to the intersection of the line through $\mathbf{C}$ and $\mathbf{X}_C$ with the $(Z=1)$-plane (referred to as image plane in the following). Coordinates obtained by this mapping are called normalized image coordinates.

Image coordinates (in pixels) are obtained via the mapping defined by the camera calibration matrix K according to

$$\tilde{\mathbf{x}} = \mathrm{K}\tilde{\mathbf{x}}_n = \begin{bmatrix} f & s & p_x \\ 0 & \alpha f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_n \ . \tag{2.2}$$

Here, f denotes the focal length and image coordinates can be scaled differently

according to the scaling factor α. Skewing of the pixel footprint is modeled by the parameter s; typically it is assumed to be 0. The entries $(p_x, p_y)^\mathsf{T}$ denote the offset of the coordinate system origin wrt. the intersection of the optical axis and image plane, it is called the principal point. We will later also use the notation $\mathbf{x} = \pi(\mathbf{X}_C)$ to refer to the combination of the projection of a 3D point in camera coordinates and the normalization into non-homogeneous pixel coordinates.

Finally, the full process of transforming a 3D world point into image coordinates is expressed by the projection matrix P as

$$\tilde{\mathbf{x}} = \mathrm{P}\tilde{\mathbf{X}}_W = \mathrm{K}\begin{bmatrix}\mathrm{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1\end{bmatrix}\tilde{\mathbf{X}}_W = \mathrm{KR}\begin{bmatrix}\mathrm{I} & -\mathbf{C}\end{bmatrix}\tilde{\mathbf{X}}_W \quad . \tag{2.3}$$

If the depth of the original 3D point is available, *e.g.*, via structure information captured in a depth map, we can compute the unprojected 3D point given its pixel location. First, by inversion of the imaging process we obtain a point ray

$$\mathbf{r} = \tilde{\mathbf{x}}_n = \mathrm{K}^{-1}\begin{pmatrix}\mathbf{x} \\ 1\end{pmatrix} \quad , \tag{2.4}$$

that corresponds to the normalize point coordinate. Second, the 3D point in camera coordinates is simply obtained by scaling to $\mathbf{X}_C = \mathbf{r}\,\mathrm{D}(\mathbf{x})$.

Parallel Projection: Compared to the perspective projection, a parallel or affine projection has its center of projection at infinity (Hartley and Zisserman, 2004, *p*171). As such, the parallel projection matrix

$$\mathrm{P}_\infty = \begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2.5}$$

maps coordinates X and Y, but neglects the depth of a point. In addition the principal point is not defined, *i.e.*, the camera calibration matrix only contains scaling and skew parameters in an upper triangular matrix $\mathrm{K}_{2\times2}$, and

$$\mathrm{K} = \begin{bmatrix}\mathrm{K}_{2\times2} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1\end{bmatrix} \quad . \tag{2.6}$$

A special case of an affine camera is the scaled orthographic projection where $\mathrm{K}_{2\times2} = s\mathrm{I}$ with the scale factor s. It will be used in Chapter 5 to generate virtual views of a scene. The full imaging process of 3D world point into image pixel locations is then described as

$$\tilde{\mathbf{x}} = \mathrm{P}\tilde{\mathbf{X}}_W = \mathrm{diag}(s, s, 1)\mathrm{P}_\infty \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathrm{T} & 1 \end{bmatrix} \tilde{\mathbf{X}}_W = \begin{bmatrix} \mathbf{r}_1^\mathrm{T} & t_1 \\ \mathbf{r}_2^\mathrm{T} & t_2 \\ \mathbf{0}^\mathrm{T} & 1/s \end{bmatrix} \tilde{\mathbf{X}}_W \ . \tag{2.7}$$

Compared to a finite camera with 11 degrees of freedom (5 for the intrinsics and 6 for the extrinsics), it has only 6 free parameters. These are 3 for the rotation, 2 for the offsets t_1, t_2, and 1 for scale s, while $t_3 = 1$.

For the un-projection of a point the corresponding ray $\mathbf{r}$ is computed with the affine camera calibration matrix and according to Eq. (2.4). Then, the original 3D point is obtained by scaling only the Z-coordinate by means of the measured depth, *i.e.*, $\mathbf{X}_C = \mathbf{r}\,\mathrm{diag}(1, 1, \mathrm{D}(\mathbf{x}))$.

Lens Distortion: Both the projective and parallel projection have been modeled as linear mappings via the projection matrix P so far. However, real world cameras usually do not follow a linear model, but exhibit distortions caused by the physical properties of a lens. In order to describe the image formation process more accurately, it suffices to establish the relationship between the physical coordinates of a pixel and the coordinates of the ideal perspective mapping. The Brown-Conrady distortion model (Brown, 1966) considers radial and tangential distortion components and is used in this thesis. Thereby, distortion is modeled on the image plane, *i.e.*, with normalized image coordinates $\mathbf{x}_n$ and distorted coordinates $\mathbf{x}_d$ are obtained by the mapping

$$\mathbf{x}_d = L(r^2)\mathbf{x}_n + \mathbf{d}_x(\mathbf{x}_n) \quad \text{where} \quad r^2 = \|\mathbf{x}_n\|_2^2 \ . \tag{2.8}$$

$L(r^2)$ models the radial distortion with the underlying assumption that the principal point coincides with the center of distortion. The most common functional approach for $L(r^2)$ is a polynomial, such that a second order Taylor expansion leads to

$$L(r^2) = 1 + k_1 r^2 + k_2 r^4 \ . \tag{2.9}$$

The tangential distortion model is contained in the term $\mathbf{d}_x$ and captures distortion due to imperfect centering of lens components. The offset is computed to

$$\mathbf{d}_x(\mathbf{x}) = \begin{bmatrix} 2xy & r^2 + 2x^2 \\ r^2 + 2y^2 & 2xy \end{bmatrix} \begin{pmatrix} k_4 \\ k_5 \end{pmatrix} \quad . \tag{2.10}$$

After distortion correction, the camera calibration matrix K maps the distorted coordinates to pixel locations $\tilde{\mathbf{x}} = \mathrm{K}\,(\mathbf{x}_d, 1)^\mathsf{T}$ as before. The compact notation $\mathbf{x} = \pi(\mathbf{X}_C)$ will also be used to include the non-linear dependency from lens distortion where appropriate; the particular meaning will be clear from the context.

The inverse of the lens distortion function of Eq. (2.8) can not be computed in closed form, due to the dependency on a higher order polynomial. Therefore, we need to resort to an iterative procedure to obtain the undistorted point $\mathbf{x}_n$ from the distorted location $\mathbf{x}_d$. However, if only the inverse mapping is required, it is valid to directly parameterize the lens distortion in the inverse direction, resulting in a closed form solution.

2.3 Image Properties

In computer vision we often look for an abstraction of the captured pixel intensity data, both to achieve better data interpretation and faster computation. In the following, appearance based information in the form of local image features and geometric information captured by vanishing point and line information are discussed.

2.3.1 Feature Detection and Description

Sparse local features define an abstraction layer and aim to summarize appearance information in an image in a consistent and repeatable way. This is in contrast to *dense* methods, which utilize all the intensity information in an image. While the latter recently regained popularity for example in deep learning scenarios (*e.g.*, Krizhevsky et al., 2012; Chatfield et al., 2014) or for real-time methods (*e.g.*, Forster et al., 2014; Engel et al., 2014), the former builds the basis for the majority of applications in 3D computer vision and also this thesis. Thereby, sparse feature extraction is a two step process, consisting

of the search for distinctive, repeatable image points by means of a detector and the subsequent description of their local image content via a descriptor. As our registration methods build upon image features, we will give a concise overview over relevant methods in the following.

In their fundamental work, Harris and Stephens (1988) and later Shi and Tomasi (1994) propose one of the first corner detectors which is widely known as Harris or Shi-Tomasi corner detector nowadays. It finds (constant size) interest points by evaluating statistics about local image gradients. To efficiently detect stable keypoint locations also at different sizes, a detector is employed on the scale space (Lindeberg, 1994) representation of an image. The SIFT (scale invariant feature transform) detector proposed by Lowe (2004) hereby is the most prominent example. It employs a difference of Gaussians filter, approximating the second order image derivative, to extract distinctive image blobs as extrema in scale space. Together with its proposed SIFT descriptor, summarizing local image gradient information, it has been proven to work reliably in real world applications and thus is also used as the feature of choice in this thesis. Inspired by SIFT, Bay et al. (2008) developed SURF (speeded up robust features) which utilize the determinant of the Hessian as a blob detector and build upon integral images to achieve faster computation. While both detectors have been shown to be robust to viewpoint changes of up to 30° degrees (Mikolajczyk et al., 2005), Matas et al. (2004) explicitly target the wide-baseline setting and propose MSER (maximally stable extremal regions). Their blob detector searches for stable image components among several binary segmentations and by this obtains invariance to affine image transformations. Finally, for real-time applications, *e.g.*, in robotics, binary corner detectors such as FAST (features from accelerated segment test) (Rosten et al., 2010) got popular recently.

In terms of descriptors, the SIFT descriptor (Lowe, 2004) is still one of the most precise and thus also widely used ones. It was extended to the GLOH (gradient location and orientation histogram) by Mikolajczyk and Schmid (2005) to consider more spatial regions, while the higher dimensionality is reduced using PCA. Following a similar concept, the HOG (histogram of oriented gradients) (Dalal and Triggs, 2005) efficiently captures image data on a regular grid and has been shown to significantly outperform existing feature sets for human detection. To facilitate dense, wide-baseline matching, Tola et al. (2010) propose DAISY, which is robust against many photometric and geometric transformations. To achieve fast computation, BRIEF (binary

robust independent elementary features) (Calonder et al., 2010) build upon binary strings as efficient, highly discriminative descriptor which are computed via intensity differences resulting in simple binary tests.

Besides the aforementioned, well-known features, a multitude of variants thereof exist. Prominent examples are ORB (Oriented FAST and rotated BRIEF) (Rublee et al., 2011) or BRISK (binary robust invariant scalable keypoints) (Leutenegger et al., 2011). The former – compared to its underlying basis feature – is invariant to in-plane rotations and resistant to noise. The latter employs the FAST based detector in scale space and forms its descriptor as an assembly of bit-strings from binary tests.

Especially the introduction of scale and rotation invariant detectors and descriptor has had tremendous influence on the robustness and performance in various algorithms in 3D reconstruction, camera tracking, object detection, or scene understanding and alike and thus is a standard building block nowadays. For an overview and evaluation of invariant detectors and descriptors, the interested reader is referred to *e.g.* (Schmid et al., 2000; Mikolajczyk and Schmid, 2004; Mikolajczyk et al., 2005) and (Mikolajczyk and Schmid, 2005), respectively. In addition Kaneva et al. (2011) present a comparison of invariant image features wrt. robustness to scene changes and image transformations using a photorealistic virtual world. Finally, a comparison of binary features is presented by Heinly et al. (2012).

2.3.2 Vanishing Points and Lines

With the knowledge of the actual camera model, also geometric information can be obtained from a single image. Most prominent in this respect is the extraction and interpretation of vanishing points and lines. Under a perspective projection, objects that stretch to infinity can have finite extent. Therefore, an infinite scene line is imaged as a line terminating in a vanishing point – or parallel world lines are imaged as converging lines, intersection in the vanishing point (Hartley and Zisserman, 2004).

Geometrically a vanishing point $\mathbf{v}$ of a world line with direction $\mathbf{d}$ is obtained by intersecting the image plane with a ray parallel to $\mathbf{d}$ and passing through the camera center $\mathbf{C}$; *i.e.*, $\mathbf{v}$ only depends on the direction $\mathbf{d}$ and not its position. The computation from image measurements is achieved by intersecting line segments in the image. Under the assumption of Gaussian noise, the maximum likelihood estimate of a vanishing point and line segments is obtained as

the best least squares fit wrt. orthogonal distances between lines and the vanishing point (Kosecka and Zhang, 2002; Zhang and Kosecka, 2002; Hartley and Zisserman, 2004).

Especially human build environments follow strong geometric properties and the majority of lines is aligned with the principal orthogonal directions of the world coordinate frame. This so called Manhattan style world is a widely use prior assumption in reconstruction algorithms (Furukawa et al., 2009a; Lee et al., 2009; Flint et al., 2010a,b; Ramalingam et al., 2013). The observation can be exploited to efficiently compute vanishing points and further to estimate the relative orientation of the camera wrt. the scene (Kosecka and Zhang, 2002). Within this thesis, we leverage this property to rotate the camera coordinate system into an upright position. Let $\mathbf{v}_v$ correspond to the vertical vanishing point and $\mathbf{r}_v = \mathrm{K}^{-1}\mathbf{v}_v$ be the ray in the vertical vanishing direction $\mathbf{d}$, then the needed transformation is represented via a rotation aligning $(0, 1, 0)^\intercal$ with $\mathbf{r}_v$. The remaining axes of the new coordinate system are chosen to be orthogonal to $\mathbf{r}_v$. Assuming that the original reference camera system was the identity matrix $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$, then the new vertically aligned coordinate system coincides with the 3D rotation

$$\mathrm{R}_v = \begin{bmatrix} \mathbf{r}_v \times \mathbf{e}_z \\ \mathbf{r}_v \\ \mathbf{r}_v \times \mathbf{e}_z \times \mathbf{r}_v \end{bmatrix} . \tag{2.11}$$

A homography which warps the image itself to an upright representation is given via

$$\mathrm{H}_v = \mathrm{K}\mathrm{R}_v\mathrm{K}^{-1} . \tag{2.12}$$

2.4 Correspondence Search

So far we have considered a single image and its projection of the 3D world only. Given two or more cameras imaging (parts of) the same scene from different viewpoints, the projection imposes constraints on the positions of projected 3D points, such that the relative pose between cameras and the actual 3D structure can be computed (cf. following Sec. 2.5-2.6). At the basis of these algorithms is the need to establish point-to-point correspondences between images, or an image and the 3D model. This search is typically

based upon the sparse features and their descriptors discussed previously. A simple solution is to perform brute force nearest neighbor search in descriptor space; however, its complexity grows quadratically in the number of features to match. Fast matching via approximate nearest neighbor search (Muja and Lowe, 2013) provides a tractable approach and is used for matching tasks in this thesis. It consists of first building a space partitioning kd-tree in descriptor space and then evaluating the (decision) tree on a query feature. Thereby the computationally complex part is the tree modeling itself, while matching exhibits a complexity of $\mathcal{O}(\log n)$ in the number of features. Constant time complexity can be achieved by clustering image features into visual words, since retrieval then amounts to indexing an inverted file. Typically the quality of the correspondence search degrades (*i.e.*, more correct matches are missed) the more abstraction is introduced. Recently, Cheng et al. (2014) have proposed a promising matching approach based on cascaded hashing, which shows comparable performance to kd-tree based matching, but is 10 times faster.

The majority of 3D computer vision algorithms relies on accurate correspondences, such that the generation of wrong matches presents a major problem. However, there is no guarantee that the closest point in descriptor space also represents the correct match. This problem gets even more severe in case of high descriptor space density, *e.g.*, for databases containing thousands of pictures or large-scale 3D models. Consequently an efficient outlier rejection method is required. The widely used ratio test (Lowe, 2004) enforces that the distance ratio between the closest and second closest match is below a certain threshold, such that only discriminative matches in descriptor space survive. Though, it rejects many correct matches in case of repetitions and high descriptor density. In Chapter 6 we present an alternative filtering strategy incorporating prior known geometric information from the camera setup and the 3D model itself within a pose estimation setting. Besides strong filtering, another options is to aim for the extraction of more similar descriptors right from the beginning. Especially for setups with considerably varying viewpoints, keypoint repeatability decreases and descriptors get distorted due to perspective effects. In this regard, Wu et al. (2008) and Cao and McDonald (2012) proposed to extract viewpoint invariant patches by means of known planar structures in the 3D geometry. We extend this idea to developable surfaces and general 3D scenes in Chapter 5. Finally, Lepetit and Fua (2006) and Ozuysal et al. (2010) represent wide-baseline matching as a classification problem. For training an object is modeled from all possible viewpoints; then

the classifier uses hundreds of simple binary features and models class posterior probabilities related to the learned camera poses.

2.5 Pose Estimation

Pose estimation comprises solutions to the problem of finding the relative pose parameters [R, $\mathbf{t}$] and, in the case of camera pose estimation, optionally (parts of) the camera intrinsics K and lens distortion parameters, given a set of tentative correspondences. First, closed form solutions for different geometric setups will be discussed. We also point out the minimal solutions, *i.e.*, an algorithm is provided with the minimal number of required correspondences, all of which are assumed to be correct. Second, we briefly mention robust estimation methods including geometric verification, which allow to find a good inlier set within numerous established, but potentially noisy correspondences. In the rest of this thesis we will resort to several of these methods.

2.5.1 Closed Form and Minimal Solutions

Relative Pose for 2D - 2D Correspondences

Estimating the relative motion between two images is a classic problem in stereo vision. Thereby, epipolar geometry (Hartley and Zisserman, 2004, *p*239ff) describes a relationship between two views which expresses the condition that corresponding point rays must intersect in space. As such, it is independent of the scene structure and solely depends on the intrinsic and extrinsic camera parameters. According to epipolar geometry, for each point $\mathbf{x}$ in one image, there exists a corresponding epipolar line $\mathbf{l}'$ in the other image, *i.e.*, the matching point $\mathbf{x}'$ in the second image must lie on $\mathbf{l}'$ and vice versa. The fundamental matrix $\mathrm{F} \in \mathbb{R}^{3\times3}$ is the algebraic representation of this property of epipolar geometry (Faugeras, 1992; Hartley, 1992; Luong and Faugeras, 1996):

$$\mathbf{x}'^{\mathrm{T}}\mathbf{l}' = \mathbf{x}'^{\mathrm{T}}\mathrm{F}\mathbf{x} = \mathbf{x}\,\mathrm{F}^{\mathrm{T}}\mathbf{x}' = \mathbf{x}^{\mathrm{T}}\mathbf{l} = 0 \ . \tag{2.13}$$

Above constraints can be written in linear form $\mathbf{a}^{\mathrm{T}}\mathbf{f} = 0$, where $\mathbf{a} = \mathbf{x}\otimes\mathbf{x}'$ and $\mathbf{f}$ is the stacked fundamental matrix. Since the common scaling is not significant, 8 (normalized) pairs of corresponding points are sufficient to compute a solution for F in linear form, *e.g.*, via a Singular Value Decomposition (Hartley, 1997).

In addition, F is a homogeneous matrix of rank 2, such that the constraint $\det \mathrm{F} = 0$ needs to be satisfied. Alternative methods for computing the fundamental matrix, both relying on sparse and dense correspondences, are discussed for example in Valgaerts et al. (2011).

In case the intrinsic camera calibration is known, only the relative pose $[\mathrm{R}, \mathbf{t}]$ between images remains unknown and the fundamental matrix is specialized to the essential matrix E. It is expressed in terms of normalized image coordinates $\mathbf{x}_n = \mathrm{K}^{-1}\mathbf{x}$, leading to the relation

$$\mathbf{x}'^{\mathsf{T}}\mathrm{F}\mathbf{x} = \mathbf{x}_n'^{\mathsf{T}}\mathrm{K}'^{\mathsf{T}}\,\mathrm{F}\,\mathrm{K}\mathbf{x}_n = \mathbf{x}_n'^{\mathsf{T}}\mathrm{E}\mathbf{x}_n = 0 \ . \tag{2.14}$$

As before, E is a 3×3 homogeneous matrix, but now only has five degrees of freedom – three for the rotation and two for the translation, since there is an overall scale ambiguity. In his seminal work Longuet-Higgins (1981) established the additional relationship $\mathrm{E} = [\mathbf{t}]_\times \mathrm{R} = -\mathrm{R}[\mathbf{C}]_\times$, which reveals that the essential matrix can be factored into its rotational and translational components (Hartley and Zisserman, 2004, *p*258f). To obtain a solution for E, the 8-point-algorithm can be applied equivalently, followed by a projection onto the space of essential matrices. However, already Kruppa (1913) showed, that due to the additional constraints only 5 point correspondences are sufficient to obtain a valid solution. In this regard, efficient solutions (*e.g.* Nistér, 2004; Stewénius et al., 2006) have been proposed more recently. Including a priori geometric information about the camera setup, such as the knowledge of the vertical camera orientation (Kalantari et al., 2011; Fraundorfer et al., 2010) or constraint motion (Li et al., 2013; Lee et al., 2013) lead to even more efficient solutions due to the reduced number of required correspondences.

Absolute Pose for 2D - 3D Correspondences

In case scene structure is known a priori, *e.g.*, from a sparse reconstruction, and correspondences are established between image and 3D points we search for the absolute camera pose wrt. the model. This setup typically arises in image based localization and tracking scenarios and setups differ by the number of required point correspondences. Hence, it is also referred to as the perspective-n-point (PnP) problem in the computer vision literature or as space resection in the photogrammetry community.

Known internal calibration: Let us first consider the case where the camera intrinsics are known. The P3P problem exhibits the smallest subset of control points that yields a finite number of solutions. It was first investigated by Grunert (1841) and brought to the computer vision community by Fischler and Bolles (1981). There exist a multitude of approaches to solve the problem and the majority of proposed methods represent non-iterative, multi-stage solutions. As such, they first estimate 3D coordinates of the image points in the local camera frame, and then solve the remaining 3D-3D alignment problem (see the following paragraph for a discussion of suitable algorithms). The algorithms differ in the way they solve the polynomial equations modeling the constraints between image and 3D points. In general one obtains up to 4 solutions for 3 points[1], which can be disambiguated using additional information, *e.g.*, a forth point correspondence. A detailed review and analysis of different algebraic solutions can be found in the work of Haralick et al. (1994), including an examination of their numeric stabilities. One of the most popular and robust P3P solvers was proposed by Gao et al. (2003). The authors use both, an algebraic and geometric approach, to provide a solution classification of the P3P equation system. In comparison to these methods, Kneip et al. (2011) was the first to use a parameterization, that computes the aligning transformation directly in a single stage. The algorithm shows lower computational cost and improved numerical stability. Moreover, Nistér and Stewénius (2007) discuss a P3P algorithm for the generalized camera setting, *i.e.*, for 3 points with different (but known) centers of projection and Albl et al. (2015) explicitly address rolling shutter effects.

The over-constrained case $n \geq 4$ results in pose estimation from redundant data with the advantage of improved accuracy under noisy data. This PnP setting includes P3P as special case. Quan and Lan (1999) and Ansar and Daniilidis (2003) propose linear solutions using 4 and 5 points, but their methods exhibit a high computational burden with a complexity of at least $\mathcal{O}(n^5)$. In comparison, both Lepetit et al. (2009) and Li et al. (2012a) show that there exist closed form solutions with $\mathcal{O}(n)$ time complexity. A natural alternative to previous non-iterative methods are iterative ones, which rely on the minimization of an appropriate criterion, *e.g.*, a geometric distance in the image or 3D domain. Such a formulation will lead to a non-convex optimization

[1] For degenerate 3D point configurations, such as co-planar points and camera center, the solution will also be degenerate .

problem, but can deal with an arbitrary number of correspondences and achieve excellent precision (in case of convergence). In this regard, the method of Lu et al. (2000) is one of the fastest and most accurate. Its objective is an error in 3D space which is minimized via alternating optimization on the rotation matrix and translation vector. The provably optimal solution by Olsson et al. (2009) leverages a branch and bound framework, but its high computational cost makes the algorithm impractical. An optimal solution in $\mathcal{O}(n^5)$ time that is also independent of the outlier rate has been present by Enqvist et al. (2012) and is used in Svärm et al. (2014) (see also Sec. 6). Recently, Zheng et al. (2013) have proposed a non-iterative, globally optimal algorithm with a computational complexity of $\mathcal{O}(n)$. This is achieved by formulating the PnP problem as an unconstrained minimization problem using a non-unit quaternion rotation parameterization. The method leverages the Gröbner basis in order to retrieve all possible solutions and thus is universally applicable.

Unknown internal calibration: In practice, the camera calibration is not always given, especially the focal length changes under zoom or auto-focus. In the classical camera calibration problem we are looking for all parameters of the projection matrix $\mathrm{P} = \mathrm{K}[\mathrm{R}, \mathbf{t}]$. The idea of the direct linear transform (DLT) algorithm (Abdel-Aziz and Karara, 1971; Ganapathy, 1984; Hartley and Zisserman, 2004) is to obtain the entries of P as the solution to a linear system of equations. P has 12 entries but due to an overall scale ambiguity, only 11 degrees of freedom. Therefore, at minimum 6 point correspondences are sufficient to obtain a solution.

For the standard pinhole camera model it is valid to assume that the principal point coincides with the image center and that the pixel footprint is squared, *i.e.*, the skew factor s can be neglected. Consequently, the focal length remains as the only unknown parameter, leading to 7 degrees of freedom and requiring a minimum of 3.5 image points for 4 known 3D points. Triggs (1999) and Bujnak et al. (2008) introduce a P4Pf method that takes 4 image points to estimates the focal length together with the external camera parameters. Sattler et al. (2014) propose to rely on the efficient P3P solver and calibrate the camera by sampling focal length values. Their sampling strategy models the probability of finding a pose better than the current best estimate, which enables to efficiently guide the sampling process. Recently, the true P3.5P problem was solved by Wu (2015). The minimal solutions allows them to

use the remaining image coordinate to filter the 10 candidate solutions which creates a significant efficiency improvement.

Especially for wide field of view lenses, radial distortion has a strong effect on the imaging process. Rather than assuming a linear projection model, Josephson and Byröd (2009) show that the unknown radial distortion and focal length can be estimated jointly, also requiring only 4 correspondences. In contrast, the P4Pfr method of Bujnak et al. (2011) estimates pose, focal length and radial distortion separately. While both approach leverage Gröbner basis solvers to obtain a solution, the latter method was shown to be more than one order of magnitude faster at comparable or better accuracy.

Partially known rotation: Often partial information about the orientation of the camera is known upfront. For example, the gravity direction of a camera can be obtained from inertial sensors in modern smart phones or extracted from the vertical vanishing point. Fixing the vertical camera orientation restricts the camera degrees of freedom to only in-plane rotations (besides translation). Therefore, two points are sufficient for pose estimation of a calibrated camera and three points, if radial distortion is considered as well (Kukelova et al., 2011). A typical practical example where the vertical direction is known is given by cameras mounted on a car which is moving on a plane. Lee et al. (2014) show that one can calibrate such a multi-camera system jointly using only 4 points in the minimal case and that 8 points are required to obtain a linear solver. If non-holonomic constraints (cf. Ackermann steering principle for wheel vehicles (Siegwart et al., 2011)) are taken into consideration, a single point correspondence suffices to solve for the camera pose (Scaramuzza, 2011).

The interested reader is referred to the website `http://cmp.felk.cvut.cz/minimal/`, which represents an excellent source of information about minimal pose solvers.

Relative Pose for 3D - 3D Correspondences

Finally, let us consider the case where all points are given in 3D space. Thereby, we aim for the alignment between two point sets $\mathcal{X}, \mathcal{Y}$ with $N \geq 3$ given correspondences. This setup for example arises in the registration of point clouds, but can also be established between depth maps. In general, the

objective function to minimize is stated as

$$E(s, \mathrm{R}, \mathbf{t}) = \sum_i \rho\left(\|\mathbf{Y}_i - (s\mathrm{R}\mathbf{X}_i + \mathbf{t})\|_2^2\right), \quad \mathbf{X}_i \in \mathcal{X}, \mathbf{Y}_i \in \mathcal{Y} \ , \tag{2.15}$$

where $\rho(\cdot)$ is a cost function of choice. If the error minimization is performed in the least squares sense, *i.e.*, $\rho(e) = e$ depicts the trivial loss, a closed form solution can be derived. According to Arun et al. (1987); Horn et al. (1988), the centroids of the point clouds $\boldsymbol{\mu}_X = \sum_i \mathbf{X}_i/N$, $\boldsymbol{\mu}_Y = \sum_i \mathbf{Y}_i/N$ will align under Gaussian noise. This property allows to decouple R and $\mathbf{t}$, such that $\mathbf{t} = \boldsymbol{\mu}_Y - \mathrm{R}\boldsymbol{\mu}_X$ and the transformed least squares problem reads as

$$E(s, \mathrm{R}) = \sum_i \left\|\bar{\mathbf{Y}}_i - s\mathrm{R}\bar{\mathbf{X}}_i\right\|_2^2 = \sum_i \bar{\mathbf{Y}}_i^\mathsf{T}\bar{\mathbf{Y}}_i + s^2\bar{\mathbf{X}}_i^\mathsf{T}\bar{\mathbf{X}}_i - 2s\bar{\mathbf{Y}}_i^\mathsf{T}\mathrm{R}\bar{\mathbf{X}}_i$$
$$\text{with} \quad \bar{\mathbf{Y}}_i = \mathbf{Y}_i - \boldsymbol{\mu}_Y, \bar{\mathbf{X}}_i = \mathbf{X} - \boldsymbol{\mu}_X. \tag{2.16}$$

Above decoupled problem formulation is common among different solution methods, which distinguish themselves in the way the rotation R is parameterized (cf. Eggert et al., 1997; Kanatani, 1994).

The so called orthogonal procrustes problem (Schönemann, 1966) relies on a matrix representation and the constraint that $\mathrm{R}^\mathsf{T}\mathrm{R} = \mathrm{I}$, but doesn't consider scaling, *i.e.*, $s = 1$. Thus, the goal is to maximize the term $\sum_i \bar{\mathbf{Y}}_i^\mathsf{T}\mathrm{R}\bar{\mathbf{X}}_i = \mathrm{tr}(\mathrm{R}\mathrm{X}\mathrm{Y}^\mathsf{T})$ wrt. the rotation[2], where X and Y are the concatenated 3D points. With the SVD of the 3×3 correlation matrix $\mathrm{X}\mathrm{Y}^\mathsf{T} = \mathrm{U}\Sigma\mathrm{V}^\mathsf{T}$ the closed form solution is $\mathrm{R} = \mathrm{V}\mathrm{U}^\mathsf{T}$. Considering isotropic scaling, the result for R remains as above and it holds $s = \mathrm{tr}(\mathrm{R}\mathrm{X}\mathrm{Y}^\mathsf{T})/\mathrm{tr}(\mathrm{X}^\mathsf{T}\mathrm{X})$ (Schönemann and Carroll, 1970). The orthogonal procrustes problem does not consider the fact, that R needs to belong to the special orthogonal group SO(3) (*i.e.*, $\det \mathrm{R} = 1$). Therefore, obtained solutions might also give a reflection ($\det \mathrm{R} = -1$), if point sets are planar or severely corrupted. In contrast, Horn et al. (1988) and Umeyama (1991) also consider the orthonormal characteristic of R to guarantee that the result will always be a rotation. A good overview over procrustes problems is given in Gower and Dijksterhuis (2004). Alternatively a rotation parameterization via quarternions (Faugeras and Hebert, 1986; Horn, 1987) or dual quarternions (Walker et al., 1991) prohibits reflections by design, still providing

[2]Note, that $\sum_{i,j} \mathrm{A}_{ij}\mathrm{B}_{ij} = \mathrm{vec}(\mathrm{A})^\mathsf{T}\mathrm{vec}(\mathrm{B}) = \mathrm{tr}(\mathrm{A}^\mathsf{T}\mathrm{B}) = \mathrm{tr}(\mathrm{A}\mathrm{B}^\mathsf{T})$.

a closed form solution.

If $\rho(\cdot)$ takes the form of a robust cost function, such as the popular L1 or Huber norm, no closed form solution can be derived anymore and we need to rely on an iterative method as briefly discussed in the next section.

2.5.2 Pose Estimation from Noisy Data

In the previous section we have assumed that point correspondences are correct. For real-world applications this is an unrealistic setting, *e.g.*, automatically matching features from appearance only is difficult and errors are frequent. Therefore, established correspondences typically contain a significant amount of outliers, which creates the need for robust estimation methods. In this regard, RANSAC exploits geometric consistency to remove incorrect correspondences, while other estimation methods rely on robust cost functions or optimal search strategies.

Geometric Verification with RANSAC

Random sample consensus (RANSAC) (Fischler and Bolles, 1981) is an efficient method for outlier rejection and probably the most common strategy to estimate a transformation in the presence of noisy data. Because of its random and non-deterministic nature, it does not fulfill consensus set maximization exactly and optimally, but provides an effective search method for finding a good geometric model explaining the relation between correspondences. At a glance, RANSAC iteratively selects a random, minimal subset from all correspondences to generate a model hypothesis by means of one of the previously presented closed form solvers. It then verifies the estimated transformation against all other matches and counts the number of inliers. A correspondence is define to be an inliers if its error distance $d_\Theta(c)$ (*e.g.*, the reprojection error or the euclidean distance between points in 3D) is smaller than a user defined threshold. The geometric model that reaches the largest consensus on the correspondences set is finally taken as the true underlying pose. Alg. 1 presents a prototypical implementation of the approach.

The number of performed iterations is crucial for the algorithm to provide some guarantees on the returned solution. Let us assume that the ground truth inlier set has size $N < |\mathcal{C}|$, where $\mathcal{C}$ denotes all tentative correspondences. Then, for a single iteration the probability that all m sampled points are inliers and

Algorithm 1 Random Sample Consensus (Fischler and Bolles, 1981)

Input: Correspondences set $\mathcal{C}$, minimal sample number m,
inlier threshold Γ, failure probability η
Initialization: $I_{best} = \{\}, \quad w_{best} = m/|\mathcal{C}|$
$k_{max} = \lfloor \log \eta / \log(1 - w_{best}^m) \rfloor + 1$
for $i = k$ **to** k_{max} **do**
  Randomly select m correspondences from $\mathcal{C}$
  Compute model hypothesis Θ from samples
  $I = \{\}$
  for $c \in \mathcal{C}$ **do**
    if $d_\Theta(c) < \Gamma$ **then**
      Add c to inlier set I
  if $|I| > |I_{best}|$ **then**
    $\Theta_{best} \leftarrow \Theta, \quad I_{best} \leftarrow I$
    $w_{best} = |I_{best}|/|\mathcal{C}|, \quad k_{max} = \lfloor \log \eta / \log(1 - w_{best}^m) \rfloor + 1$
Return: Tuple $(\Theta_{best}, I_{best})$ consisting of best model and inlier set

we get a correct model is

$$P_{inliers} = \prod_{i=0}^{m} \frac{N-j}{|\mathcal{C}|-j} \leq \left(\frac{N}{|\mathcal{C}|}\right)^m = w^m \ . \tag{2.17}$$

Consequently, $(1-w^m)$ denotes the probability that at least one of the samples is an outlier and a wrong model is estimated. Over the sequence of k iterations the failure probability, *i.e.*, the probability that the algorithm never selects a valid sample set, is $\eta = (1-w^m)^k$. Hence, RANSAC needs to take at least

$$k_{max}(w, m) = \left\lfloor \frac{\log \eta}{\log(1-w^m)} \right\rfloor + 1 \tag{2.18}$$

iterations to ensure that a correct model is found with probability $1-\eta$. Both, the number of samples m and the inlier ratio w, have a strong impact on the required number of iterations. It is seen immediately from this context why minimal pose solvers are such important – any solver that needs less sample points for hypothesis generation will speed up the overall computation

significantly. Additionally, w is updated during the search, if the current best solution reflects a higher inlier ratio than initially expected, allowing an earlier stop.

One typically refines the solution obtained by RANSAC via a non-linear optimization on the found inlier set. This guarantees that the final estimated model best explains the relation between matches that have been identified as correct.

Robust Parameter Estimation

RANSAC is the method of choice, if the correspondence set is contaminated by a large amount of incorrect matches. If the outlier ratio is known to be rather small, robust parameter estimation methods present an alternative. In order to do so, a registration task is formulated as non-linear least squares problem. The objective function which we seek to minimize in dependence of parameters θ then is

$$E(\theta) = \sum_i \rho\left(\mathbf{r}_i(\theta)^\mathrm{T}\mathbf{r}_i(\theta)\right) \quad . \tag{2.19}$$

The residual term $\mathbf{r}_i$ for the absolute camera pose estimation and 3D point cloud alignment (cf. Eq. (2.15)) problem is

$$\mathbf{r}_i = \mathbf{x}_i - \pi(\mathrm{R}\mathbf{X}_i + \mathbf{t}) \quad \text{and} \quad \mathbf{r}_i = \mathbf{Y}_i - s\mathrm{R}\mathbf{X}_i - \mathbf{t} \ , \tag{2.20}$$

respectively. This formulation has the advantage that lens distortion is naturally modeled in the camera projection $\pi(\cdot)$ as well. An equivalent formulation can also be established for epipolar geometry expressed in terms of the fundamental or essential matrix (Hartley and Zisserman, 2004, p284f). ρ is chosen to be a robust cost functional like the convex L1 or Huber norm, or non-linear Cauchy and Tukey m-estimators which down weight the influence of outliers even more (Zhang, 1997; Huber and Ronchetti, 2009). Due to the nonlinearity in the rotation parameterization (and in the projection for the absolute pose problem), iterative methods are used for optimization. Prominent examples are the Gauss-Newton or Levenberg-Marquardt (LM) algorithm.

Recently, interesting alternative methods for robust pose estimation have been proposed, that perform a complete search in the parameter space to retrieve the optimal inlier set. The solutions differ in the used approxima-

tions that make the search problem tractable. While Enqvist and Kahl (2008) decouple the search for rotation and translation and derive approximate constrains for L_∞ optimality (Kahl and Hartley, 2008), Enqvist et al. (2009) show that joint inlier detection and pose estimation corresponds to the (NP complete) vertex cover problem and suggest approximate solutions for it. Li (2009) proposes a reformulation of the consensus set maximization as a mixed integer programming problem which provides convex under-estimates. The approximations from all these approaches provide bounds on the solution and thus allow to leverage a branch and bound (Land and Doig, 1960) style algorithm for exploiting the search space efficiently. In contrast, Chin et al. (2015) depict that consensus maximization for a wide variety of vision tasks can be posed as a tree search problem, which can be solved very efficiently. Finally, Enqvist et al. (2012) show that the number of outliers can be minimized in $O(n)$ time. Their algorithm obtains the optimal inlier set by extracting all Karush-Kuhn-Tucker (KKT) points to a constructed optimization problem. In many cases this is too slow to be practical. Therefore, Ask et al. (2013); Svärm et al. (2014); Fredriksson et al. (2014) propose simple and fast outlier rejection methods to be used as preprocessing step to the optimal estimation. Our absolute pose estimation procedure proposed in Chapter 6 also goes in this direction and formulates the outlier filtering as an efficient, linear-time voting problem.

2.6 Multiview Sparse 3D Reconstruction

So far we have targeted the problem of relating views to each other and estimating the underlying relative motion. In this section we will use the relation between local features from several images to recover both, the camera motions and the scene structure. Therefore, the problem is termed *Structure-from-Motion* (SfM) and much research has been devoted to it in the computer vision literature over the last decades. The thesis does not contain contributions to core SfM algorithms, but sparse SfM reconstructions are leveraged in our proposed approaches in Chapters 3, 4 and 6. In particular the auto-calibration algorithm in Chapter 4 can be seen as an augmented bundle adjustment problem. Thus, we would like to give a brief overview.

SfM can be structured into a three step process. First, 2-view matching and geometric verification is performed to estimate the epipolar geometry and by

this the relative motion from matched image features (cf. Sec. 2.5.1). Second, the different camera poses together with the scene points are recovered in a global camera coordinate system. Previous calibrated two-views allow matches to be reconstructed through triangulation. If camera intrinsics are unknown, self calibration (*e.g.*, Pollefeys et al., 1999) needs to be employed. Third, a global non-linear optimization refines the estimated structure and motion.

The reconstruction is strictly projective, since all measurements are carried out in the image domain. As a result, there remains a scale ambiguity in the model. To get a metric reconstruction, constraints are needed either on the structure or the motion. In addition, no structure can be recovered in case the camera undergoes a pure rotation, or if the scene is planar. The former is missing parallax between views (*i.e.*, the cameras have zero baseline), while for the latter the epipolar geometry is not unique. In both cases the mapping between image points is fully described by a homography, rather than the actual scene depth.

While there exist optimal strategies for the first and last processing step, algorithms mainly differ in how they perform the second, structure recovery step as follows.

Incremental SfM: In incremental SfM new views are added in an iterative fashion by first establishing 2D-3D correspondences between the current reconstruction and the new image features, then estimating and evaluating the view's absolute pose (against the initially estimated relative poses), and finally updating the structure. Intermediate bundle adjustment is necessary to rigidify local structure and motion and ensure successful reconstruction. Many well known SfM systems such as Pollefeys (1999); Pollefeys et al. (2004); Snavely et al. (2007); Pollefeys et al. (2008); Agarwal et al. (2009); Wu (2013) have been built in this way over the last years. Due to the incremental nature of the algorithm, it suggests itself for sequential image sequences or real-time applications, where immediate reconstruction is desirable. In the robotics community this iterative reconstruction setting is known as monocular simultaneous localization and mapping (SLAM) (Davison, 2003; Davison et al., 2007). However, all approaches are known to suffer from drift due to the accumulation of errors, have difficulties in handling loop closures efficiently, and the success of the reconstruction is likely to depend on the image order.

Global SfM: In contrast, global SfM methods consider the entire view graph at once. Hence, they are particularly applicable to unordered image collections where no spatial or temporal order can be assumed. Algorithms start by first estimating global rotations and then refining the relative translations. To disambiguate the reconstruction and detect incorrect epipolar geometry (*e.g.*, from wrong matches or repetitive structures), the view graph is filtered by removing any two view constraints that do not conform with the estimated global rotations (*e.g.* Zach et al., 2008, 2010; Roberts et al., 2011; Jiang et al., 2012). Finally, triangulation of points leads to an initial structure and motion. There exist established algorithms for global rotation estimation (*e.g.* Hartley et al., 2013; Moulon et al., 2013), which exploit loop constraints in the graph; however, the calculation of the translation is more evolved, due to the present scale ambiguity. Possible solutions (Govindu, 2001; Jiang et al., 2013; Crandall et al., 2013; Wilson and Snavely, 2014) differ in how the parameterize the problem and compute the global camera translations.

Local Refinement via Bundle Adjustment: Both, incremental and global SfM methods create an initial sparse point cloud together with camera poses that are refined in the third processing step by means of a global non-linear optimization. In *bundle adjustment* (Brown, 1958; Triggs et al., 2000; Ni et al., 2007; Zach, 2014) the objective function models the reprojection error between projected 3D points and their image observations. The optimization is then carried out within a potentially robust, non-linear least squares framework, according to

$$\theta^* = \min_{\theta \in \mathbb{R}^n} \sum_{i,j} \rho\left(\mathbf{r}_{ij}(\theta)^\mathsf{T} \Sigma\, \mathbf{r}_{ij}(\theta)\right) = \min_{\theta \in \mathbb{R}^n} \sum_{i,j} \rho\left(\left\|\Sigma^{1/2} \mathbf{r}_{ij}(\theta)\right\|_2^2\right) \quad , \tag{2.21}$$

where $\mathbf{r}_{ij}$ corresponds to the residual term belonging to the i^{th} 3D point and j^{th} camera. Therefore, the reprojection error computes to

$$\mathbf{r}_{ij}(\pi_j, \mathbf{w}_j, \mathbf{t}_j, \mathbf{X}_i) = \mathbf{x}_{ij} - \pi_j\left(\mathrm{R}(\mathbf{w}_j)\mathbf{X}_i + \mathbf{t}_j\right) \quad . \tag{2.22}$$

The matrix Σ induces a norm on the residual, resulting in a prior on the normal equations of the linearized objective function. For example, we have incorporated the individual location uncertainties of feature points via Σ in an earlier work (Zeisl et al., 2009). π models the non-linear camera projection

and depends on the camera intrinsics and lens distortion parameters. The rotational part of the camera pose [R, **t**] is often parameterized via a 3 DoF angle-axis representation **w**. The corresponding rotation matrix is computed via the so call Rodrigues formula (exponential map in $SO(3)$) to

$$\mathrm{R}(\mathbf{w}) = \exp([\mathbf{w}]_\times) = \mathrm{I} + [\bar{\mathbf{w}}]_\times \sin\phi + [\bar{\mathbf{w}}]_\times^2 (1 - \cos\phi) \tag{2.23}$$

with $\bar{\mathbf{w}} = \mathbf{w}/\|\mathbf{w}\|_2$ denoting the rotation axis and $\phi = \|\mathbf{w}\|_2$ the rotation angle.

Solving Eq. (2.21) involves the first oder linearizion of the residual term, leading to a least-squares problem at the current working point, which is characterized by the Jacobian $\mathrm{J} = \partial\mathbf{r}(\theta)/\partial\theta$. Solving the normal equations includes a factorization of $\mathrm{J}^\mathsf{T}\mathrm{J}$. Sophisticated methods like the Schur complement decomposition exist to exploit the sparsity of the problem and significantly reduce computational complexity. Typically the update step is then performed within a Levenberg-Marquardt framework. Efficient solvers like Ceres solver (Agarwal et al., 2015), g2o (Kummerle et al., 2011), or incremental smoothing and mapping (iSAM) (Kaess et al., 2008, 2012) exist which foster efficient implementation.

2.7 3D Vision with RGB-D Data

Recently introduced RGB-D sensors like Microsoft Kinect, Asus Xtion Pro Live, or time of flight sensor jointly capture image and depth data. Laser scanners or structured light sensors provide this capability for years already; however, it is the real-time processing and low cost of commodity RGB-D sensors that has fostered their the widespread use in the computer vision community. 3D modeling becomes a lot easier then, since the scene geometry is already recovered, which circumvents stereo reconstruction and leaves the alignment of individual scans as the key problem. With the presence of depth data, the estimation of the relative pose between two images is solvable in closed from with only three point correspondences via 2D-3D or 3D-3D matching (cf. Sec 2.5.1), compared to 5 points if only image data is considered. We will cover this topic of point correspondence based registration in more detail in the following Sec 2.8.

The availability of dense geometry allows for image warping between views, which is in favor of visual odometry methods estimating the camera trajec-

tory directly from image intensities without the need for expensive feature extraction and matching (*e.g.*, Newcombe et al., 2011a; Steinbrücker et al., 2011; Kerl et al., 2013). Once camera poses are recovered, immediate dense modeling is straight forward to achieve, *e.g.*, via volumetric fusion and an implicit surface representation (Chen and Medioni, 1991; Curless and Levoy, 1996). Recent research therefore concentrates on the modeling of larger scale scenes (Henry et al., 2012, 2013), combined tracking and mapping (Newcombe et al., 2011b,a), or even the reconstruction of articulated (Schmidt et al., 2014) or dynamic environments (Newcombe et al., 2015). The generation of high quality reconstructions is generally limited by the accuracy of the sensor input itself. Especially commodity depth sensors exhibit substantial distortion. Our RGB-D sensor auto-calibration procedure in in Chapter 4 targets this problem. Finally, the ability to easily capture large amounts of (training) data also inspired the use of regression methods for the classical problem of (re-)localization in 3D models (Gee and Mayol-Cuevas, 2012; Shotton et al., 2013; Glocker et al., 2013).

2.8 3D Model Registration

At last, let us look at the problem of 3D model alignment, where we aim for the fully automatic registration of different model parts to obtain a consistent, fused model. Partial models may have been acquired from different sources such as commodity depth sensors or laser scanners, but (sub)models built via SfM techniques may serve as data source likewise. The unknowns in this process are the relative and global registrations among the sub-models. The algorithms for alignment addressed in Sec. 2.5.1 already provide closed form and robust solutions from a set of given point correspondences. Thus, we are left with the task to establish those correspondences and following methods primarily differentiate amongst each other by the way the perform model matching.

2.8.1 Implicit correspondence generation

One of the most famous approach for point cloud alignment is the iterative closed point (ICP) algorithm (Besl and McKay, 1992). It does not require any given relationship between two models, but iteratively establishes and updates correspondence hypothesis during optimization. As the name of the algorithm

suggests, the maintained correspondence set is typically defined by the nearest point neighbors between models in euclidean space, but applicable alternatives are the point to plane (Chen and Medioni, 1991) or point to triangle distance (see Rusinkiewicz and Levoy (2001) for a comparison of distance metrics). The original ICP algorithm requires that one of the models is a subset of the other; therefore, it can not handle outliers or models which only share partial overlap. In contrast Zhang (1994) consider local point statistics, *i.e.*, their method differs in how point matching is performed, and thus can handle outliers. While ICP is proven to converge, it will only do so to the closest local minimum. Hence, initialization of the algorithm with a good initial transformation is essential for successful registration.

While ICP updates direct point correspondences and uses a closed form solver on them in each iterations, Fitzgibbon (2003) utilizes the champfer distance transform to represent the 3D model on a discrete voxel grid. This allows for a consistent error functional with immediate distance error evaluation and enables faster, non-linear least-squares based optimization.

In Sec 2.5.2 we already briefly mentioned globally optimal algorithms for robust pose estimation; however, they still require a set of tentative correspondences. Li and Hartley (2007) go one step further and assume point correspondences to be unknown as well. In contrast to ICP, their method guarantees a globally optimal solution without any initialization. This is achieved by jointly solving the correspondence and alignment problem and performing a global search in the rotation space $SO(3)$, again relying on efficient search via branch and bound.

2.8.2 Explicit correspondence generation

Normally the modeled scene captures enough geometric or appearance variation such that the identification of distinctive salient locations allows for explicit correspondence generation between overlapping model parts. Since ICP requires a good initialization, feature based matching suggests itself as a preprocessing step (Pandey et al., 2011). Thereby features may either be extracted directly on the geometry data (*e.g.* Johnson and Hebert, 1999; Rusu et al., 2009a; Yamany and Farag, 2002) or from texture information (cf. Sec 2.3.1) such as reflectance or color images.

Considering depth only, Stamos and Leordeanu (2003) build upon segmentation based features and exploit the fact, that the inter-model lines between

points from two matches need to conform, for an early hypothesis rejection. Huber and Hebert (2003) utilize a surface mesh matching engine and show that a graph based optimization allows to reject wrong correspondences. While corner like features are applicable to depth scans as well, if they are customized to address depth related phenomena (Barnea and Filin, 2008), Aiger et al. (2008) have shown that no keypoint detection is needed at all due to the geometric relation of points. If one selects a random coplanar point quadruple in one model, there exists a finite set of congruent quadruples in the second model, which can be evaluated quickly via RANSAC (cf. Sec 2.5.2); though, the efficiency degenerates with less overlap between scans. Sub-sampling of point clouds, *e.g.*, retaining 3D keypoints (Theiler et al., 2014), speeds up the process but does not require any descriptors for matching.

Geometry based alignment methods share the common problem that shape variation is essential, but often limited, especially for urban scenes. Texture or reflectance information on the other hand can help significantly to resolve ambiguities. However, the imaging process is strongly viewpoint dependent an distorts local descriptors, which creates the need for appearance normalization in wide-baseline matching scenarios. Invariant image regions have been obtain for example by intersection of the surface with a 3D sphere (Wyngaerd and van Gool, 2003), via a local planar shape approximation from surface normals and subsequent texture warping (Köser and Koch, 2007), by means of plane detection in point clouds (Wu et al., 2008) or from vanishing points (Cao and McDonald, 2012; Srajer et al., 2014), or by the introduction of virtual views for location recognition (Irschara et al., 2009). In Chapter 5 we will extend the idea of planar feature normalization first to general developable surfaces and then also free-form scenes.

Besides correspondences based on local features, also global geometric constraints have been used for the alignment tasks; *e.g.*, extended Gaussian images or surface orientation histograms enable alignment, if they are analyzed in the spherical Fourier domain (Makadia et al., 2006).

2.8.3 Other Variants of Alignment

So far, we have only considered the pairwise alignment of two 3D models, but most of the time a final model, consisting of multiple subparts or augmented with image data is of interest.

Multiple scans: For the registration of multiple scans (*e.g.* Pulli, 1999; Theiler et al., 2015), the goal is to minimize the global registration error. If the individual models are represented as nodes in a connectivity graph built from available pairwise relations, then the distribution of the alignment error through the graph is the optimal strategy. This type of optimization got popular in the literature as pose graph optimization (Lu and Milios, 1997; Borrmann et al., 2008; Kummerle et al., 2011; Kaess et al., 2012). However, it considers the established underlying pairwise constraints as fixed. In contrast, Nishino and Ikeuchi (2002) integrate the various relative pose estimations between scans in one common least squares framework for joint optimization with a robust cost function, which is computationally more complex, but also achieves better accuracy. The work is part of the Great Buddha Project, with which Ikeuchi et al. (2007) impressively demonstrate how model registration can be used to digitally preserve cultural heritage for later generations.

Image to model registration: To fully capture the details of a model, it is often desirable to also integrate texture information. This raises the questions of how images can be registered wrt. an established model. Local keypoints are typically not common between both modalities, and higher level descriptions such as line based features (Liu and Stamos, 2005, 2007; Stamos et al., 2008) are used for image to model matching. The maximization of mutual information (Mastin and Kepner, 2009; Pandey et al., 2012) represents and alternative. It exploits the statistical dependence between optical appearance and measured LIDAR elevation, eliminating the need for specific calibration targets (*e.g.*, artificial landmarks, line features, etc.).

If a collection of images or video data capturing the same scene is available, a two step process for alignment is possible. One first resorts to SfM techniques (cf. Sec 2.6) to build a sparse 3D model and then solves the remaining 3D-3D registration problem between the reconstructed point cloud and the initial model (Zhao et al., 2005; Liu et al., 2006; Stamos et al., 2008; Novak and Schindler, 2013). By this the images are successfully mapped to the model. To obtain an optimal color projection, Corsini et al. (2013) rely on maximizing the global mutual information between overlapping images.

Non-rigid registration: The assumption that model parts only differ by a rigid transformation is not met in case the modeling process is effected by noise

or other distortions such as device nonlinearities or calibration inaccuracy. To still obtain one global consistent model, there is the need to allows for a local deformation of model fragments while registration is performed. Such non-rigid alignments have been used for the fusion of individual scans (Brown and Rusinkiewicz, 2007), only partial overlapping range data (Li et al., 2008), or smaller local reconstruction from depth data (Zhou and Miller, 2013). The result are improved registrations with preserved surface details; though, due to the local shape deformation guarantees about the accuracy of the reconstruction are lost.

3 Stereo Reconstruction of Texture-less Building Interiors

In this chapter, we consider the first step in a reconstruction pipeline, *i.e.*, the estimation of scene depth from images only. The proposed method resorts to classical stereo vision and illustrates an efficient computational solution for depth map estimation in indoor-scenarios.

Man-made environments constitute a very difficult setup for passive stereo vision, since they typically contain only a few visually salient objects predominantly exhibiting homogeneous areas with weak textures. Office-like indoor environments also often contain specular or even transparent objects violating the Lambertian surface assumption, thus making image-based reconstruction even harder. Therefore, the Manhattan-world assumption – *i.e.*, that major surfaces are parallel with either the ground plane, or with one of two orthogonal planes – is recently utilized as a strong prior in several approaches for image-based modeling. We propose to replace the Manhattan-world assumption by a related, but somewhat different prior for indoor environments:

> *The open/maneuverable space is bounded by parallel ground and ceiling planes, and by purely vertical structures (i.e., mostly walls).*

Further, vertical elements are assumed to be (piecewise) smooth in 3D. Under this assumption our method is able to *hallucinate* the most probable vertical structure whenever it is obscured by non-vertical elements (*e.g.*, people or furniture), or alternatively it can detect non-vertical objects and insert depth measurements from a different source (*e.g.*, local stereo).

We illustrate the difficulties of obtaining a meaningful depth map in an indoor environment in Figure 3.1, where a stereo pair depicting a hallway is shown (Figures (a) and (b)). Columns in the images already correspond to vertical structures in 3D. The floor and the ceiling have significant view-dependent highlights, and the scene is partially weakly textured. These properties result

in poor depth maps using local (best cost) and scanline optimization (Figures (c) and (d)). Incorporating a strong piece-wise planarity prior (Figure (e)) or even global optimization for stereo (Figure (f)) returns visually appealing disparity maps, but both methods have major difficulties in the ground region (due to the specularities). Explicit incorporation of a vertical world assumption significantly stabilizes depth estimation with and even without vertical smoothness (Figures (g) and (h)).

In this work we explore the utility of the vertical structure prior for challenging indoor environments. Our contributions are as follows:

- We derive a minimal parameterization for the vertical structure prior suitable for depth extraction from pixel-wise matching costs.
- The obtained depth map is demonstrated to represent a tiered labeling, substantially simplifying the overall stereo problem.
- We formulate the final optimization as an efficient dynamic programming problem that enforces smooth depth changes.
- We introduce an optional model selection stage to detect image columns violating the vertical assumption.

The resulting algorithm exhibits a low computation cost, such that all processing steps run at interactive frame rates, which makes it suitable *e.g.*, for autonomous system navigation.

In the following Section 3.1 we cover related work, whereas Section 3.2 explains the underlying idea of a vertical structure prior and required preprocessing steps. Next, the utilization of vertical structures in the algorithm is outlined in Section 3.3, followed by the incorporation of smoothness assumptions via dynamic programming in Section 3.4. Experiments, illustrating the effectiveness of our approach, are presented in Section 3.5. Finally Sections 3.6 conclude with a summarizing and prospective discussion.

3.1 Review

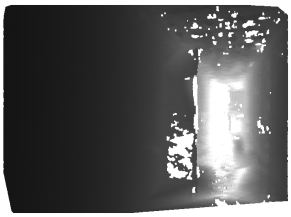
Stereo matching has been and continues to be one of the most investigated topics in computer vision. We will give a brief introduction and discuss the various possibilities for the incorporation of geometric priors in the stereo reconstruction process. The interested reader is also referred to the works of

(a) Left image

(b) Right image

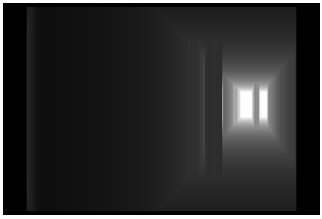
(c) Per pixel best cost depth

(d) Scanline opt. (Scharstein and Szeliski, 2002)

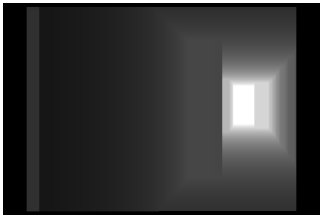


(e) libELAS (Geiger et al., 2010)

(f) Global opt. (Zach et al., 2009)



(g) Vertical aggregation



(h) Our result (with DP)

Figure 3.1: Difficulties in dense stereo computation in indoor environments arise from little texture and specular surfaces (a-b). Local (c) and scanline optimization (d) lead to poor depth maps, while stronger regularizations (e-f) return bettern results. Explicit incorporation of a vertical prior (g-h) clearly stabilizes the depth estimation.

Scharstein and Szeliski (2002) and Seitz et al. (2006), which give an excellent taxonomy of dense, two-frame and also multi-view stereo methods.

Stereo reconstruction constitutes a dense correspondence problem. Given two views, epipolar rectification (Ma et al., 2004, p404ff) defines an image transformation that maps the respective epipoles at infinity, resulting in all epipolar lines being parallel. Matching between the warped images then reduces to a search for corresponding points along horizontal scan lines, rather than the whole image. Multiple images can not be handled at once in this way; however, plane sweeping (Collins, 1996) provides and alternative. It does not require a priori rectification, but measures the agreement between frames by projecting images onto a virtual plane that is moved through the scene. The algorithm was later adapted to work in real-time on graphics hardware (Yang and Pollefeys, 2003) and extended to multiple sweeping directions (Gallup et al., 2007). In general the resolution of computed disparity maps depends on the spatial configuration of views. Best results are achieved if the baseline is kept variable and adapted to the actual scene depth (Gallup et al., 2008). The most common used pixel-based matching costs in the literature are the sum of absolute difference (SAD), the sum of squared intensity differences (SSD), and normalize cross correlation NCC which can handle illumination changes (Scharstein and Szeliski, 2002; Heo et al., 2011).

Independent, per pixel depth extraction is generally ambiguous and additional constraints are added that support smoothness by penalizing changes of neighboring disparities. Recently, Hosni et al. (2013) proposed a local, edge preserving cost volume filtering that aligns label transitions with color image edges. On the contrary, global optimization methods aim to recover smooth surfaces via the incorporation of additional geometric constraints. Usually, this prior is very generic and formulated in terms of pairwise (sometimes higher order) clique potentials in a Markov random field (Felzenszwalb and Huttenlocher, 2006) favoring small depth discontinuities. The resulting discrete multi-label problem is then solved via graph cut (Boykov et al., 2001; Kolmogorov and Zabih, 2004; Felzenszwalb and Zabih, 2011). Since the computational complexity of these methods is high, researchers have been looking for faster ways of smoothing the matching cost volume. Optimization along a scan line only, is efficiently performed via dynamic programming (DP) Scharstein and Szeliski (2002), but vertical consistency between scanlines is not enforced, which leads to streaking artifacts. Therefore, Hirschmüller (2008) approximate the global 2D smoothness constraint by combining many 1D constraints, each

of them solved via DP. Veksler (2005) showed that a single tree structure gives a sufficient approximation to the 2D grid which can also be solved via DP. This paradigm of DP on a tree was extend to nodes consisting of scan line segments (Deng and Lin, 2006) and image segmentations (Mei et al., 2013).

Let us now turn to the specific problem of stereo estimation in man-made environments. There the reconstruction typically requires very strong assumptions, *i.e.*, scene priors, to be able to handle texture-less regions successfully. In particular this applies to indoor environments where weakly textured, homogeneous surfaces (*e.g.*, uniform walls) are dominant in the image. Consequently, several strong priors for reconstructing urban environments and building interiors are proposed in the literature.

Man-made outdoor environments are usually composed of mainly *piece-wise planar* surfaces. This strong assumption can be incorporated at different steps in the image-based reconstruction process: first, computation of the matching costs between images can be improved by considering several surface orientations (derived *e.g.*, from dominant vanishing directions or from a sparse 3D point model (Gallup et al., 2007; Mičušík and Košecká, 2010)). Further, the robustness of depth map extraction and the efficiency of 3D model representation can be significantly enhanced (Furukawa et al., 2009a; Sinha et al., 2009). A different, but usually even stronger model for outdoor urban environments assumes purely vertical facades emerging from a ground plane (Cornelis et al., 2008). The corresponding depth map representation is extremely efficient: after image alignment with the vertical direction only one depth value per image column needs to be determined and stored in the depth map. Further, depth map computation is very robust, since the matching costs along an image column can be (robustly) fused to determine the single required depth value. We leverage these advantages in our work.

Reconstructing indoor environments, *e.g.*, office spaces or corridors, from images even poses a more challenging task, since texture-less or only weakly textured surfaces are predominant. In many cases line structures corresponding to (orthogonal) vanishing directions allow the inference of simple planar, Manhattan-like models from single images (Lee et al., 2009; Flint et al., 2010a). Unfortunately, these methods are not suitable for (near) real-time applications due to their expensive inference stage to determine the most likely 3D configuration. Fusing several depth maps, generated under the Manhattan-world assumption, can give impressive results (Furukawa et al., 2009b). Since our application is targeted towards real-time usage, such a high-quality approach is

not feasible because of run-time constraints, and the potential lack of required redundancy in the captured image data.

In order to handle weakly textured regions, dense correspondence methods typically utilize some prior model on the resulting depth map as motivated before. The assumption of piecewise planarity of the imaged environment can be explicitly incorporated by assigning locally planar depth hypotheses to image regions induced by super-pixel segmentation (*e.g.*, Sun et al., 2005; Klaus et al., 2006; Wang and Zheng, 2008). A fast stereo method strongly using the piecewise planar assumption was proposed by Geiger et al. (2010). This approach first determines a sparse set of very confident correspondences, and uses the induced Delaunay triangulated surface model as strong prior for the generation of a complete depth map. Gallup et al. (2010) extends the piecewise planar model and explicitly introduces an additional label for non-planar surfaces. Images are segmented into planar and non-planar regions by means of photoconsistency and learned appearance, and finally non-planar regions are modeled with a standard stereo approach.

Continuous algorithms often rely on a convex formulation (eg, Chambolle and Pock, 2010) that aim to minimize the total variation of the reconstructed geometry. These algorithms suffer from stair-casing artifacts, since first oder total variation favors piece-wise constant solutions. To handle arbitrary surface orientations, total generalized variation (TGV) (Bredies et al., 2010) based denoising was applied for stereo estimation by Ranftl et al. (2012), resulting in piece-wise planar depth maps.

3.2 Vertically Aligned Stereo Representations

We want to start our explanation by motivating for the layout of a vertical structure in the scene and its representation in the image domain. According to the illustration in Figure 3.2 let us assume for the left camera that

(i) the optical axis is parallel to the ground plane,

(ii) it has extrinsic parameters [I, **0**] and thus defines the reference coordinate system,

(iii) vertical structures in the scene posses a vertical layout in the image domain, and

(iv) that the height of ceiling and ground plane are known.

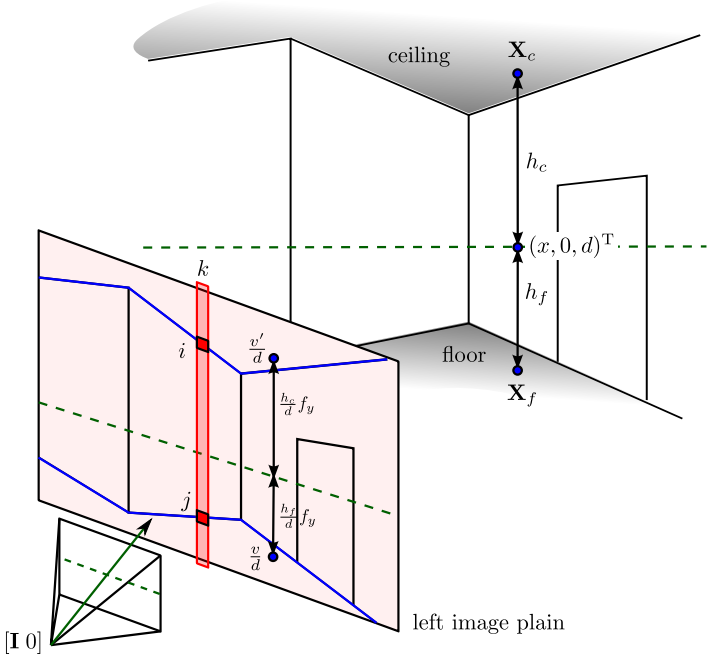


Figure 3.2: Layout of a vertical structure in 3D and its projection in the image domain. Corresponding ceiling and floor points $(\mathbf{X}_c, \mathbf{X}_f)$ are coupled via their common depth d, resulting in a general coupling of image points v', v and finally of boundary points i, j. Consequently a vertical structure is fully described either by its depth d or the tuple (i, j). See the text for a derivation of the relationship.

We will explain how we can guarantee these requirements, but first it is important to note that under these assumptions the intersection points of a purely vertical element (which can be seen as an upright line in 3D) with the ground and ceiling plane share the same depth. Hence, a mapping from points on the floor to corresponding points on the ceiling (and vice versa) has just one degree of freedom, namely the ratio of floor and ceiling heights as derived in the following.

Given heights h_c and h_f for ceiling and floor plane with plane normal $\mathbf{e}_y = (0, 1, 0)^\intercal$, two corresponding points are $\mathbf{X}_f = (x, h_f, d)^\intercal$ and $\mathbf{X}_c = (x, h_c, d)^\intercal$. Thus, we obtain for the respective (homogenous) image positions

$$(u, v, d)^\intercal = \mathrm{K}\mathbf{X}_f \quad \text{and} \quad (u', v', d)^\intercal = \mathrm{K}\mathbf{X}_c \tag{3.1}$$

$$\tag{3.2}$$

with K describing the camera intrinsics. Since $\mathbf{X}_c = \mathbf{X}_f + (0, h_c - h_f, 0)^\intercal$ we are only interested in the parameter change in the y direction. Due to the upper triangle structure of K (and also K^{-1}) such a coordinate change is independent from the horizontal location in the image domain. In the remainder of the paper we will denote vertical image positions by indices

$$i = \frac{v'}{d} = \frac{h_c}{d} f_y + p_y \quad \text{and} \quad j = \frac{v}{d} = \frac{h_f}{d} f_y + p_y \ , \tag{3.3}$$

and use the pair (i, j) to specify the ceiling and floor boundary, *i.e.*, the start an end point of a vertical structure in the image. These quantities are only dependent on the depth d and thus define the mapping

$$i = \frac{h_c}{h_f} j + \left(1 - \frac{h_c}{h_f}\right) p_y \tag{3.4}$$

by eliminating the factor f_y/d from both equations in Eq. (3.3). As a result, with known camera intrinsics and heights h_c, h_f *either* d, i or j allows to fully specify a vertical structure. As a consequence, the scene geometry depicts a tiered structure (cf. Fig. 3.3), where depth estimation is solely restricted to the vertical elements.

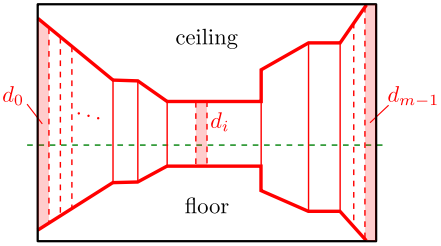


Figure 3.3: Tiered structure labeling between ceiling, floor and the vertical elements (which itself posses varying labels wrt. to the scene depths.)

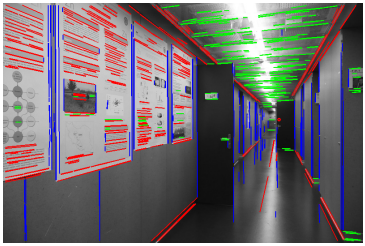


Figure 3.4: Vanishing lines corresponding to the three dominant directions as found in indoor environments.

Image Alignment with Vertical Direction This simple relation between image projections of corresponding points on the floor and the ceiling plane only holds for cameras aligned with the vertical direction. This constraint is not fulfilled a priori, but can be met by warping images by an appropriate homography. We utilize the vertical vanishing point to determine the upright direction by first detecting edges, followed by a line growing and clustering step, and finally by rejecting outliers via a RANSAC approach as descriped in the work of Kosecka and Zhang (2002). See Fig. 3.4 for an example of extracted orthogonal vanishing lines in a typical indoor scene.

Vertical scene structures match with image columns if the corresponding vertical vanishing point $\mathbf{v}_v$ lies at infinity, *i.e.*, at $(0, 1, 0)^\intercal$. The underlying homography that warps the image to an upright orientation was introduced in Eq. (2.12) of Sec. 2.3.2. The rotation R_v (cf. Eq. (2.11)) describing the aligned camera coordinate system also influences the initial pose $[\mathrm{R}, \mathbf{t}]$ of the second stereo image, which changes accordingly to $[\mathrm{R}_v^{-1}\mathrm{R}, \mathbf{t}]$.

Identification of Floor and Ceiling plane Knowledge of the ceiling and floor heights h_f, h_c is important, since they define the relation between hypothesis depths and vertical structures. We take a data driven approach, where the mapping of boundary points, *i.e.*, the ratio h_c/h_f in Eq. (3.4), is determined by robustly voting for corresponding points on edges above and below horizon (similar to Flint et al., 2010a), thereby relying on strong edges at structural

boundaries. Next, we fit vertical structures (see following Section 3.3) with random boundary pairs (i, j) in the matching cost volume and determine the depth with minimum vertical cost. Implicitly we retrieve corresponding ceiling and floor heights and in this way vote for the most likely ground and ceiling configuration. Alternatively, for robotic applications it is likely that the height of the camera(s) above ground is fixed and known. A sampling of ground contact points similar to Cornelis et al. (2008) will give a stable estimate of the ground plane over time. A line based reconstruction scheme (inspired for example by Taylor and Kriegman (1995); Smith et al. (2006); Schindler et al. (2007c); Micusik and Wildenauer (2014)) may also be utilized for the estimation of corresponding plane heights. It is applicable if at least two boundary points lie on the same edge.

Calibration and Stereo Image Matching Calculation of matching costs for various depths requires the knowledge of camera poses and intrinsics. For our application setting we assume that either a calibrated stereo camera pair is used, or that SfM or visual SLAM is applied for self localization. Therefore, we consider camera poses and intrinsics as given.

Matching costs for different depth hypotheses may be calculated along scan lines for a rectified image pair. In general, aligning the cameras with vertical elements in the images usually vitiate the rectified setup; hence, we employ a plane sweep approach (Gallup et al., 2007) to calculate the matching cost volume. Sweeping directions are set along the optical axis (*i.e.*, fronto-parallel and thus aligned with column-wise vertical structures) and in direction of the vertical axis to match ceiling and ground plane. Planes are chosen such that depth hypothesis exhibit a linear spacing in the disparity domain. In the following the resulting cost volume is denoted by $\mathrm{C}(x, y, \mathbf{p})$, where the tuple $\mathbf{p} = (\mathbf{e}, d)$ encodes the sweeping direction $\mathbf{e}$ and distance (depth) d from the reference camera center $[\mathrm{I}, \mathbf{0}]$.

3.3 Vertical Structure Hypothesis

Assuming a vertical structure along an image column k, its start point i, end point j, and depth d can be used synonymously for parametrization as was described in previous Sec. 3.2. In this way all possible depth hypotheses d relate to index pairs (i, j), *i.e.*, $d \mapsto (i, j)$ according to Eq. (3.3), and encode

the cost table D_k for column k.

The cost for an assumed vertical structure in image column k is given by the sum over individual matching costs at its depth hypothesis d via

$$D_k^V(d) = \sum_{r=i}^{j} \mathrm{C}(k, r, (\mathbf{e}_z, d)) \ . \tag{3.5}$$

If boundary points (i, j) lie within the image, the support of accumulated matching costs along the fixed ceiling and floor plane can be facilitated with

$$D_k^C(d) = \sum_{r=0}^{i-1} \mathrm{C}(k, r, (\mathbf{e}_y, h_c)) \ , \tag{3.6}$$

$$D_k^F(d) = \sum_{r=j+1}^{m-1} \mathrm{C}(k, r, (\mathbf{e}_y, h_f)) \ . \tag{3.7}$$

For the calculation of D_k^C and D_k^F we make use of the cumulative structure along the ceiling and floor plane, *i.e.*, we calculate running sums of matching costs. For D_k^V cost accumulation is not possible, since each depth d in the cost volume is just considered only once. Fig. 3.5a visualizes the cost aggregation within the matching cost volume for a certain depth hypothesis.

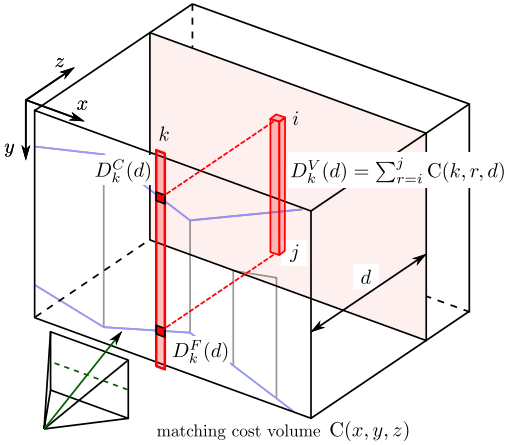
The combined cost D_k for a vertical structure at depth d and image column k is described by the aggregation of previous three terms by

$$D_k(d) = D_k^C(d) + D_k^V(d) + D_k^F(d) \ . \tag{3.8}$$

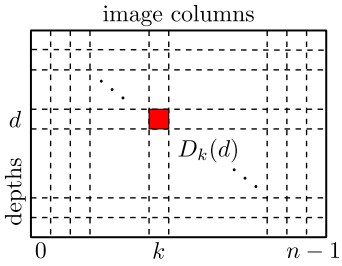
The most suitable combination of vertical structures simplifies to solving for the column-wise minimum over possible depths (see Fig. 3.5b for the cost table):

$$d_k^* = \arg\min_d D_k(d) \quad \forall k \in \{0, \ldots, m-1\} \ . \tag{3.9}$$

Images in Figures 3.8(c) and 3.9(c) illustrate the depth maps obtained by this local optimization.



(a) Matching cost volume



(b) Cost table, consisting of the accumulated costs for different depths.

Figure 3.5: (a) For a constant depth, vertical costs are accumulated between bounding indices (i, j) in the matching cost volume. The final costs $D_k(d)$ for a column k and depth d include the additional aggregated costs along floor and ceiling plane according to Eq. (3.8). (b) Hypothesizing several depths for each image column results in the final cost table, where we search for a smooth transition between vertical structures via dynamic programming.

3.4 Optimization via Dynamic Programming

Given the best cost solution obtained via Eq. (3.9) one can observe undesired depth discontinuities, especially at locations where the solution is ambiguous. In this section we will present how smoothness between neighboring vertical structures can be enforced and how the optimization problem can be solved efficiently via dynamic programming.

In general dynamic programming guarantees to find the global optimum for an energy function like

$$E = D_0(l_0) + \sum_{k=1}^{n-1} D_k(l_k) + V(l_k, l_{k-1}) \quad , \tag{3.10}$$

with labels $l_k \in \mathcal{L}$, unary terms $D_k(l_k)$ and binary terms $V(l_k, l_{k-1})$. In the simplest setting $\mathcal{L}$ contains the set of possible depths and we have $l_k = d_k$. Then $D_k(d_k)$ is the cost for a vertical structure at depth d_k as computed in Sec. 3.3. The resulting smoothness term $V(d_k, d_{k-1})$ constitutes a penalty for large label changes, *i.e.*, it penalizes deviations in depth. It could be spatially varying with location k as well, but we did not make use of this generalization. Note that smoothness along columns is already encoded in the data terms, because the vertical structure prior only allows one single depth. In our setting we use a linear cost model for $V(\cdot, \cdot)$ with slope λ_d and truncated by t to allow for large depth changes, if the data term indicates so. The penalty for a depth change reads as

$$V(d_k, d_{k-1}) = \lambda_d \min\left(|d_k - d_{k-1}|, t\right) \quad . \tag{3.11}$$

The dynamic programming algorithm is traversing over image columns, left to right, and accumulates costs up to the current position. In column k for label d_k it searches over all previous depths d_{k-1} and selects the one with minimum accumulated costs and regularization penalties. The related dynamic programming cost table, denoted as $C_k(d)$, is written as (for better readability we will drop the index from labels and depths in the following)

$$\begin{aligned} C_k(d) &= D_k(d) + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d}) + V(d, \hat{d}) \right\} \\ &= D_k(d) + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d}) + \lambda_d \min\left(|d - \hat{d}|, t\right) \right\}, \end{aligned} \tag{3.12}$$

with the initialization $C_0(d) = D_0(d)$. We use the fast min-convolution proposed by Felzenszwalb and Huttenlocher (2006) to update $C_k(d)$ for all depths d in linear time. The optimal solution is found by backtracking over C_k for $k = m - 1 \ldots 0$.

3.4.1 First Extension: Slope based Smoothness Term

The regularization term in Eq. (3.11) prefers structures with constant depths, which is not always suitable for the often observed piecewise linear assembly of vertical structures (recall Fig. 3.1). Directly adding a curvature regularization as proposed in Amini et al. (1990) via ternary cliques is expensive due to the quadratic complexity in the number of labels. The alternative is to extend the

labels by a slope value, hence a label $l_k = (d_k, s_k)$ consists of a depth and a respective local slope value. Thus, the binary cliques for the smoothness are sufficient. We obtain a speed-up by limiting the values of s_k to a small range. The smoothness term now reads as

$$V(l_k, l_{k-1}) = \lambda_s |s_k - s_{k-1}| + \lambda_d |d_k - d_{k-1} - s_{k-1}| \ . \tag{3.13}$$

Consequently, changes in direction and depth discontinuities (compensated by the local slope value) are penalized.

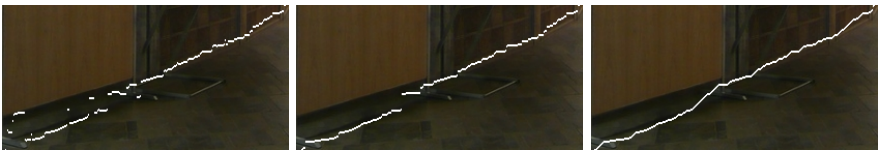
Similar to Eq. (3.12) the search for the best previous state in a dynamic programming step now has to consider both, previous depths and slopes:

$$\begin{aligned} C_k(d,s) &= D_k(d) + \min_{\hat{l}=(\hat{d},\hat{s})} \left\{ C_{k-1}(\hat{d},\hat{s}) + V(l,\hat{l}) \right\} \\ &= D_k(d) + \min_{\hat{s}} \left\{ \lambda_s |s-\hat{s}| + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d},\hat{s}) + \lambda_d |d-\hat{d}-\hat{s}| \right\} \right\}. \end{aligned} \tag{3.14}$$

The minimization over $\hat{d}$ has the same structure as beforehand and thus can be sped up again by efficient computation of the lower envelope (Felzenszwalb and Huttenlocher, 2006). Due to the introduction of the slope variable we face a two-dimensional minimization problem. However, the number of possible slopes is quite small, *e.g.*, $\mathcal{S} = \{-2, -1, 0, 1, 2\}$. For each slope the min-convolution can be executed separately, which increases complexity by a factor of $|\mathcal{S}|$. Fig. 3.6 shows the improved recovery of depths with the slope-based regularization (by means of a smoother and more accurate intersection boundary between a vertical structure and floor).

3.4.2 Second Extension: Model Selection

Given that a scene contains a non-vertical structure, the algorithm tries to fit the best vertical model in terms of matching costs. Fig. 3.7 illustrates such a case and shows the result for the vertical approximation in subfigure (b). In (c) we are comparing pixel-wise best matching costs (minimum in cost volume over all depth hypotheses) with the matching costs at depths described by the optimal fitted vertical structure. The result highlights exactly these areas where non-vertical structures (and also occlusions) are present. We can make use of this property by adding a new label to the optimization problem



(a) Per column best cost (b) DP without slopes (c) DP with 5 slopes

Figure 3.6: Improvement in the estimation of the vertical structure boundary by the introduction of slope-based regularization (images are a cutout from the lower left part of the scene in the last row of Fig. 3.9)

describing a non-vertical structure. The goal in the optimization then is to decide for a certain depth (assuming a vertical structure) or for a non-vertical structure.

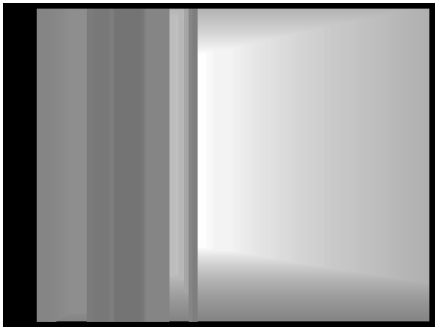
The cost for a non-vertical structure along a column is the sum over per pixel minimal matching costs as motivated before. This sum will always be smaller than any cost aggregation over vertical structures; therefore, we add a constant scalar bias b leading to

$$D_k(l = \text{non-vertical}) = b + \sum_{r=0}^{m-1} \min_d \mathbf{C}(k, r, (\mathbf{e}_z\ d)) \quad . \tag{3.15}$$

In the regularization a constant penalty t_2 is added for a label change between a vertical an a non-vertical structure and vice versa. A label transition within non-vertical structures is not directly penalized, but implicitly covered via the bias B. The final smoothness term results in

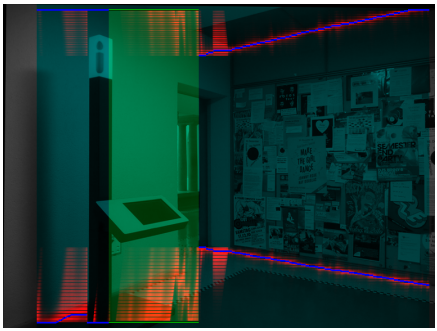
$$\bar{V}(l_k, l_{k-1}) = \begin{cases} V(l_k, l_{k-1}) & \text{if } l_k, l_{k-1} \text{ are vertical} \\ 0 & \text{if } l_k, l_{k-1} \text{ are non-vertical} \\ t_2 & \text{otherwise} \end{cases} \quad . \tag{3.16}$$

(a) Left stereo image



(b) Hallucinated vertical structures

(c) Difference in matching costs



(d) Non-vertical parts (in green)

Figure 3.7: Model selection between vertical and non-vertical structures. (c) illustrates the matching cost difference between the pixel-wise best cost solution and the fitted vertical structures from (b); clearly visible is the error in columns containing the info screen.

3.5 Experimental Evaluation

In our experimental setup we capture a scene from several view points. First, we run a SfM pipeline (Zach, 2014)[1] to estimate camera poses. A pair of images is chosen from the sequence and aligned with the vertical direction. Second, we execute a plane sweep stereo matching to generate the cost volume; thereby 256 plane hypotheses are tested. Intensity differences are measured via SAD in a 7×7 matching window. Finally, costs for vertical structures are calculated and an optimal sequence is retrieved via dynamic programming. We optimize over 256 discrete depth values, corresponding to the number of sweeping planes. Costs for vertical structures are normalized to lie in the range $[0, 1]$; the same applies for depths values. With that $\lambda_d = 3$ and smoothness terms are truncated above $t = 0.2$. The bias for costs supporting a non-vertical structure was set to $b = 1200$ (before normalization) and the penalty for a change between vertical and non-vertical models was set to $t_2 = 0.2$. We incorporate 5 possible slopes with $\lambda_s = 0.5$.

In Fig. 3.8 results are illustrated for scenes predominantly featuring vertical structures. Computed depth maps are not absolutely accurate due to the strong vertical structure presumption, but provide dense depth estimates without artifacts for texture-less regions. Results for scenes were vertical and non-vertical structures coexist are shown in Fig. 3.9. Occlusions and non-vertical structures correctly cause a model change. Finally, Fig. 3.10 exhibits scenes were our depth estimation fails. It mainly occurs if the vertical assumption is clearly hurt or matching costs are inaccurate because of specular, transparent environments.

In terms of speed our plane sweep algorithm (GPU implementation) requires 160ms to generate the cost volume for images of sizes 768×576. Cost calculation for vertical and non-vertical structures takes 50ms (on a single CPU core). Finally, basic dynamic programming is executed in 5ms; using 3 and 5 slope values execution times are 46ms and 120ms, respectively. Based on this measurements our approach is well suited for real time applications. Consequently, it is conceivable to enable live, dense reconstruction for indoor environments in the spirit of Newcombe and Davison (2010). Global stereo optimization (Zach et al., 2009) in comparison takes 2 seconds on a GPU processing down-sampled images of size 384×256. By comparison ELAS (Geiger

[1] We use the open source implementation of Simple Sparse Bundle Adjustment (SSBA) available from `https://github.com/chzach/SSBA`

et al., 2010) is also very fast and has a run-time of 320ms (but takes rectified images as input). Full scanline optimization for stereo (Scharstein and Szeliski, 2002) (with GPU accelerated matching cost calculation) requires about 1.4s.

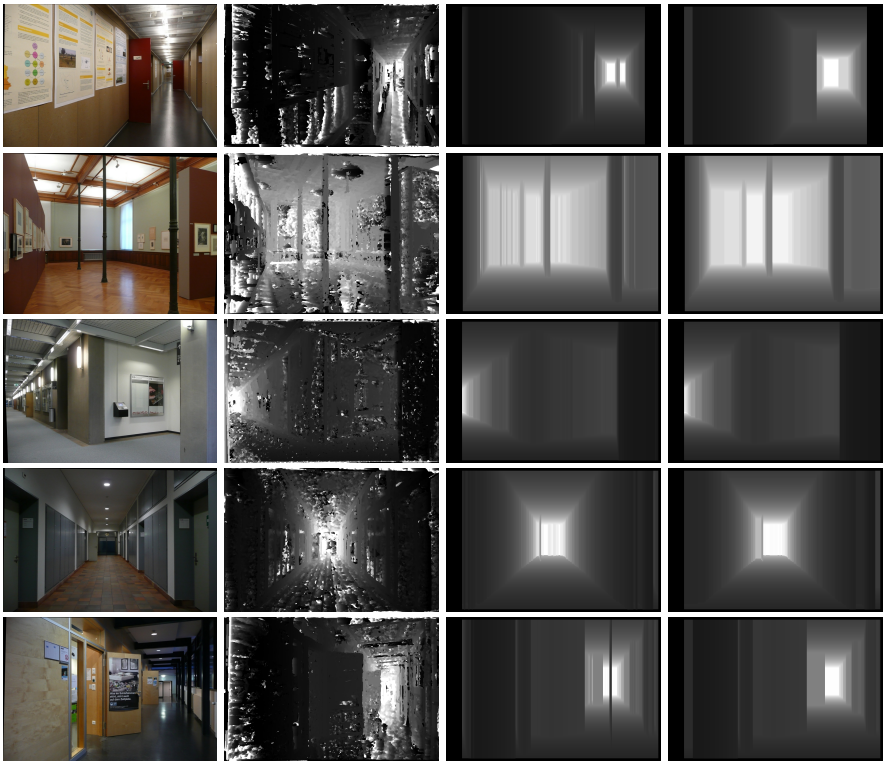
3.6 Discussion

In addition to the approach presented in Sec 3.4 we explored additional, potentially more powerful methods.

First, for an image aligned with the vertical direction the semantic layout of floor, middle, and ceiling regions is a tiered one (Felzenszwalb and Veksler, 2010). Hence the *simultaneous* determination of floor and ceiling boundaries in the image, and deciding whether the pixel column in between has either a vertical or a general depth structure is in principle possible in one dynamic programming pass. We initially considered using a label set consisting of depth values (for vertical columns) and index pairs indicating the floor and ceiling boundaries (for general columns). Using similar acceleration techniques as presented in Felzenszwalb and Huttenlocher (2006); Felzenszwalb and Veksler (2010) the complexity of dynamic programming is $O(nm^2L)$ for an $m \times n$ image and considering L depth values, which we decided is too expensive for our target application. As reference, the presented implementation has a complexity of $O(nL)$.

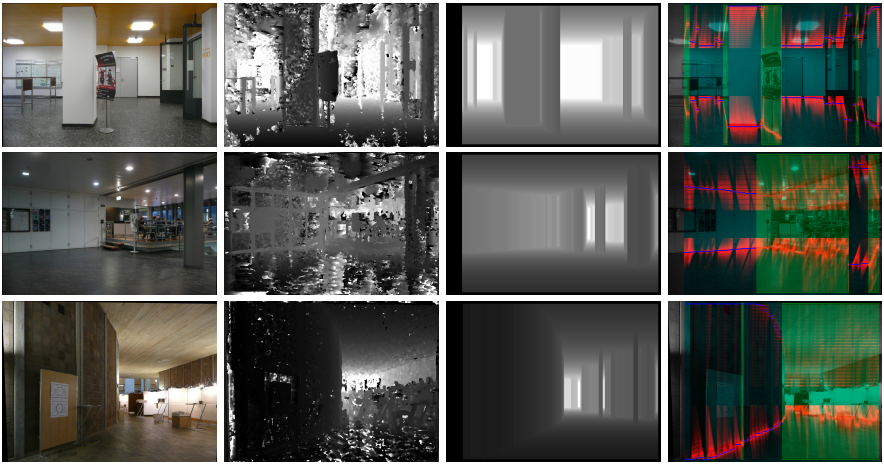
Second, since the computationally most expensive step is the matching cost calculation, one aims on replacing the general, expensive plane-sweep approach by a cheaper method. The plane-sweep method is only fast, if hardware support (*e.g.*, a GPU providing fast texture sampling) is available. Otherwise, a standard stereo setup with aligned scanlines is preferable. This can be achieved, but only if the baseline between the cameras is parallel to the ground plane (or is very close of being parallel). In such a setting changing the depth of a fronto-parallel 3D plane amounts to shifting the image in horizontal direction. With the appropriate samples of depth values (corresponding to integral disparities), subpixel access can be avoided. This simplification is only available, *e.g.*, for driving robots, but not for humanoid (walking) ones or micro aerial vehicles.

For future research it is interesting to incorporate the vertical structure prior into a broader variety of reconstruction algorithms. A relatively straight forward extension is its application for depth map fusion algorithms, since it



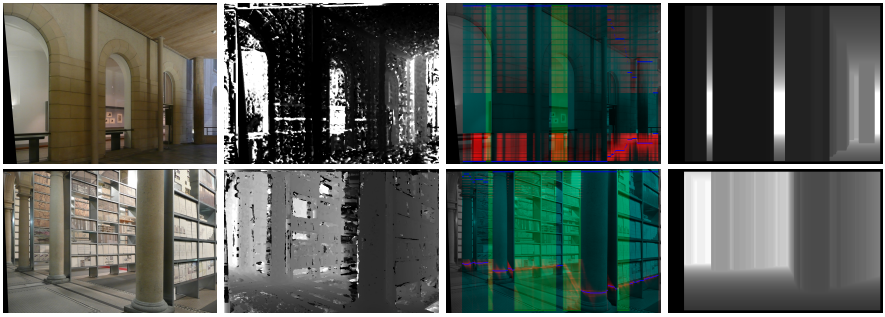
(a) Upright images (b) Best cost depth (c) Vertical aggregation (d) Our result with DP

Figure 3.8: Depth maps for less textured indoor environments. Images also exhibit small non-vertical parts, e.g. ceiling, open doors and structured walls, for which our depth maps in column (d) show a visually pleasing fit of vertical structure to the scenes.



(a) Upright images (b) Best cost depth (c) Best vertical depth after DP (d) Labeling of non-vertical structures

Figure 3.9: Depth calculation for scenes containing non-vertical, general structures. Column (d) illustrated detected non-vertical areas and occlusions with a green overlay. The blue line indicates the best sequence of indices (i, j) after DP. In red the likelihood (based on an exponential mapping of costs $D_k(d)$) for an index (i, j) is shown. (best viewed in color)



(a) Upright images (b) Best cost depth (c) Detected non-vertical structures (d) Depth, best vertical assumption

Figure 3.10: Failure cases. First row: Reflections on the glass and structure behind the arches violate the vertical structure assumption. Second row: The floor boundary of the vertical structures is set too high, since book shelves posses holes at their bottom resulting in less support for a vertical structure continuation. As a consequence DP wrongly estimates large parts of the scene as non-vertical. (best viewed in color)

only requires altering the data term such that it measures the data fidelity wrt. the input depth maps. More challenging is the utilization within a SLAM system, which relies on good local features and thus typically fails in textureless environments. Given our two-view stereo result as input, the strong prior on the observed scene depth is expected to considerably stabilize the tracking performance.

In summary, the proposed algorithm works exceptional well for scenes that are largely consistent with our underlying assumption of planar vertical surfaces, even for poorly textured images. In addition, a remarkable complexity reduction is achieved compared to regularization methods relying on global optimization (*e.g.*, Zach et al., 2009). The assumption of the dominant presence of vertical structures in a scene is strong prior, but is obviously violated as soon as one aims to reconstruct more general scenes. Thus, an alternative direction for further work is the relaxation of the prior to allow for depth changes along a vertical element, *e.g.*, modeled via a mixture of active shape

models leading to a linear combination of basis shapes. We investigated in a similar direction, but leveraged local patches instead of full image columns as the prior entity.

3.6.1 Local Patch-Based Priors

In the following we illustrate obtained results and the applicability of local, patch-based priors, that are also directly modeling the expected surface structure. They point out an interesting different direction to model and incorporate priors for local surface shape in stereo.

The particular choice of a patch prior is inspired by the successful application of equivalent representations in image processing (*e.g.* Aharon et al., 2006; Elad and Aharon, 2006; Wright et al., 2009; Mairal, 2010). It enables to model higher-order priors by means of a dictionary based approach and effectively regularization beyond triple cliques and second order priors. Due to the limited variation in surface shape (for our setting of reconstructing man-made environments) the dictionaries can be small, leading to fast inference.

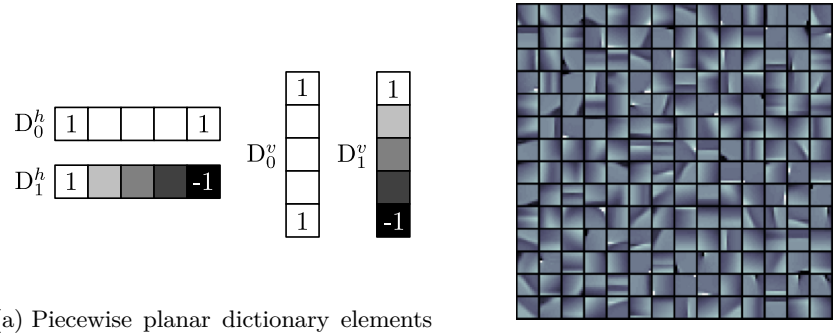
The basic energy functional measures the local deviation of the reconstructed surface from both, (a) the underlying measured data and (b) the local patch prior, modeled via the dictionary:

$$E(\mathbf{u},\mathbf{a}) = \int \underbrace{\phi_p(u_p)}_{\text{data term}} + \underbrace{\eta\, \varphi_p(\mathbf{u},\mathrm{D},\mathbf{a}) + \lambda\|\nabla\mathbf{a}_p\|_2}_{\text{patch based regularization}} \,\mathrm{d}p \ . \tag{3.17}$$

For stereo computation the data term directly measures the agreement of image intensities. Since relating image pixels between two images includes a non-linear warping, a relaxation of matching costs is employed to obtain a convex function according to

$$\phi_p(u_p) = |I_1(u_p) - I_0| \approx |I_1(u_p^0) + (u_p - u_p^0)\nabla_u I_1 - I_0| \ . \tag{3.18}$$

Within the same framework, also the fusion of several depth maps can be performed. Thereby the data term models the data fidelity for $k \geq 1$ inputs. The reconstruction error wrt. an input depth value f_p^k is weighted by w_p^k, *e.g.*,



(a) Piecewise planar dictionary elements of length 5

(b) Learned over-complete dictionary

Figure 3.11: Dictionary elements modeling the local shape of disparity maps.

modeling the data precision, leading to the data term

$$\phi_p(u_p) = \sum_k w_p^k |u_p - f_p^k| \ . \tag{3.19}$$

Different choices for the patch based regularization term $\varphi_p(\cdot)$ will be given in the following. The additional penalization of the coefficient variation via $\|\nabla \mathbf{a}\|$ is common among them and enforces a constant prior (*e.g.*, the same surface orientation) for neighboring pixels.

Linear 1D Patches In our work published in Haene et al. (2012) we aim for locally planar surfaces. In this case the dictionary can be modeled as two one-dimensional sets of patches that are orthogonal to each other, with allows much faster inference compared to squared patches. The vertical dictionary elements are given as $\mathrm{D}_0 = \mathbf{1}_w$, and $\mathrm{D}_1 = \{2i/(w-1) - 1\}_{i=0}^{w-1}$. The horizontal prior is just their transpose (also see Fig. 3.11a). The regularization term then is defined as

$$\varphi_p(\cdot) := \sum_{d \in \{h,v\}} \left\| \mathrm{R}_p^d \mathbf{u} - \mathrm{D}^d \mathbf{a}_p^d \right\|_1 \ , \tag{3.20}$$

where the linear operator R_p^d extracts horizontal and vertical 1D patches of length w from the depth map u. With this formulation $\mathbf{a}_p$ is a two vector, modeling a constant depth and slope at pixel p in $\mathbf{u}$. Exemplary stereo results with this piece-wise planar prior are depicted in Fig. 3.12. As expected, using a TV regularizer leads to visible staircaising artifacts, in particular in texture-less areas. Contrary, our patch-based formulation models planar surfaces well, also overcoming the constraints imposed by a pure vertical prior. Using a GPU implementation the approach runs at interactive frame rates of 7Hz for image of size 512×384 pixels.

Learned Patches Inspired by the good results obtained with a hand-crafted patch based prior, we also invested in learning a dictionary from training data[2]. The regularization term then resembles the sparse coding formulation

$$\varphi_p(\cdot) := \|\mathrm{R}_p\mathbf{u} - \mathrm{D}\mathbf{a}\|_2 + \mu\|\mathbf{a}\|_1 \ , \tag{3.21}$$

where D is an over-complete dictionary of size $w^2 \times N$ ($N > w^2$) and R_p extracts squared patches of size $w \times w$ from the depth map u. The additional regularization on $\mathbf{a}$ constraints the otherwise infinite many solutions and promotes sparse solutions, *i.e.*, only a small set of dictionary elements are active to form the actual prior.

Dictionary learning utilizes an equivalent objective function, and the solution is found via alternating optimization between D and $\mathbf{a}$. For training data we first considered using depth maps from active depth sensors like Kinect, but found them to be too noisy for our task. We thus turned to renderings from synthetic scenes, capturing the shape variety.

Results are twofold: While the increase in shape prior variety results in a more accurate modeling of details in the scene, it fails to suppress noise adequately at the same time. Consequently, more compact dictionaries, such as the piece-wise planar one, seem to be more appropriate for shape regularization. This is remarkable, considering the huge success of dictionary based methods in image denoising or in-painting. Though, texture alters much more frequently than shape – especially for man-made environments and when only small, local patches are considered. Thus, we argue that a general over-complete dictionary is inappropriate as a structure prior for depth map regularization, and stronger

[2]Most of this work was done by David Samuelsson and is available in his Master thesis entitled "Learned patch priors for depth map fusion", ETH Zurich, 2013.

Input image

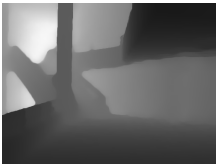
TV prior

Piecewise planar patch prior

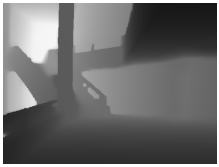
(a) Stereo computation

Input image

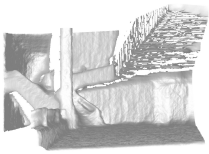
Input depth map



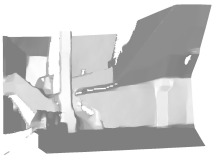
Huber-TV fusion

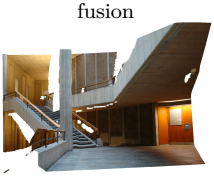


Piecewise planar fusion



Huber-TV fusion





Piecewise planar fusion

(b) Depth map fusion

Figure 3.12: Reconstruction results for indoor scenes computed with our patch based regularization. While our method obtains piece-wise planar surfaces, a total variation based regularization favors piece-wise constant solutions and thus exhibits the typical stair-casing artifacts.

constraints are required. An interesting direction for future research is to model the prior dependent on the underlying semantic class. By this the intra class variance can be restricted considerably, while the variance between classes is able to remain high.

4 Structure-Based Calibration of RGB-D Sensors

In the previous Chapter 3 we have seen how strong geometric priors can help to obtain visually pleasing and geometrically simplified but correct 3D representations from images with little texture. In this chapter we account for the recent success of commodity depth sensors. Among different applications areas they are also well suited for 3D modeling, but require an initial registration and un-distortion to unfold their full potential. Our proposed method does not depend on artificial targets, but is designed to leverage the environment geometry for calibration.

In fact, commodity depth sensors have enabled remarkable progress in diverse applications like camera tracking (Kerl et al., 2013) and re-localization (Steinbrücker et al., 2011; Shotton et al., 2013), obstacle detection, 3D modeling (Newcombe et al., 2011b; Zhou and Miller, 2013), object detection and semantic segmentation (Silberman et al., 2012), or human pose estimation (Shotton et al., 2011), that would not have been possible solely with passive stereo cameras or from image data.

Several of these works jointly leverage color and depth information and assume that the sensor pair is calibrated. Usually, the extrinsic pose between the color camera and depth sensors is estimated by means of a checkerboard pattern for structure recovery and planarity constraints on the observed scene, or from explicit point correspondences between both modalities. The computation of depth sensor intrinsics and distortion is then targeted in a subsequent estimation step (*i.e.*, it assumes the relative pose to be known) and relies on the a priori known geometry of the calibration target. Although self-calibration is a well studied problem for cameras (*e.g.*, Faugeras et al., 1992; Pollefeys et al., 1999), we are unaware of any approach that aims to fully calibrate a RGB-D camera setup, *i.e.*, that jointly estimates both, the relative RGB-to-depth transformation and the depth distortion field without the use of artificial

calibration targets.

Consequently, in this work we aim for the target-less extrinsic and intrinsic calibration of a color camera - depth sensor pair. Our approach builds on the hypothesis, that

> *a model built via structure-from-motion (or equivalently SLAM) offers an accurate enough reconstruction to determine the full calibration of a RGB-D sensor as well as the present distortions in the depth measurements.*

This means, that we aim to leverage a sparse point cloud as a geometric prior int the calibration step. To that end we introduce a calibration approach that aims to jointly minimize the alignment error between the sparse reconstruction and all measured scene depths. The accuracy of these measurements is known to degrade considerably with increasing distance from the camera (as for any stereo device). We account for it via a spatially varying correction term representing the depth deviation. Fig. 4.1 exemplifies results of an obtained reconstruction and registration from an initially uncalibrated RGB-D sensor. Our contributions are as follows:

- We formulate the registration and calibration as a joint minimization over image reprojection and depth alignment error and by this obtain a calibration that allows for accurate registration and 3D modeling.
- Our approach determines the extrinsic pose without the need for any artificial calibration targets.
- We actively compensate for errors and deviations in the measured depth, which is shown to undistort captured models especially in the far range.

As will be shown in Sec 4.4, the resulting method enables accurate modeling and texturing from uncalibrated, ad-hoc RGB-D camera pairs.

The rest of this chapter is structured as follows. Sec. 4.1 will provide a review of RGB-D sensor calibration and cover related work, while in Sec. 4.2 we provide an overview over our approach and the employed calibration model. Sec. 4.3 then explains the optimization and required initialization steps. Finally in Sec. 4.4 we provide results to our experiments and conclude with a discussion in Sec. 4.5.

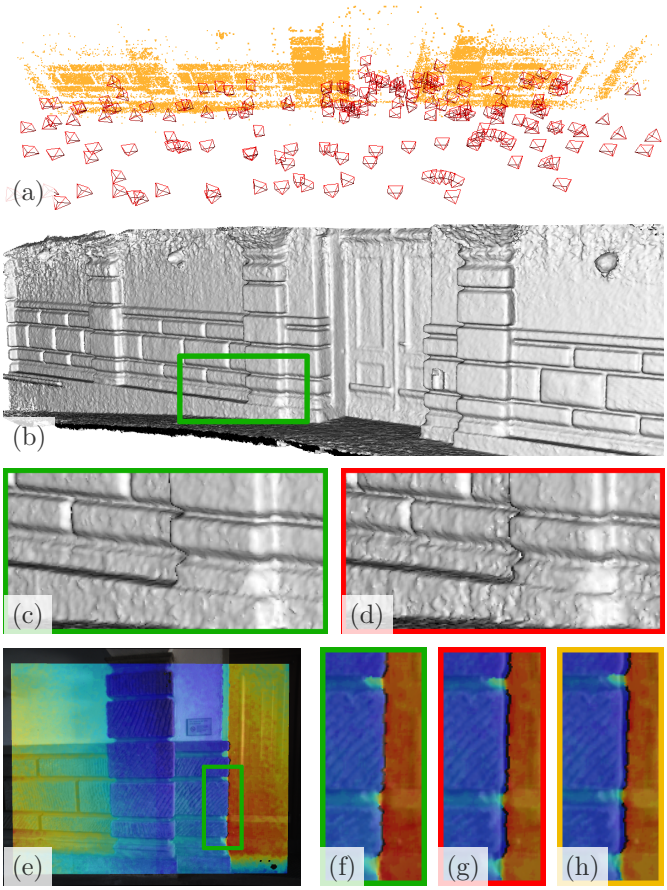


Figure 4.1: An un-calibrated RGB-D sensor (Asus Xtion Pro Live) leads to noisy reconstructions (d) and misalignment between the color images and depth measurements ((g): factory calibration, (h): calibrated IR sensor). Our approach performs self-calibration without any manual interaction or the need for special calibration patterns. Instead we build upon a sparse 3D model (a) reconstructed from the sensor's images and use the obtained geometry as a constraint in our optimization for the extrinsic and intrinsic sensor parameters. As result we obtain a more accurate reconstruction (b,c) together with a correct alignment of depth and image data (e,f).

4.1 Review

Most previous work relies on a planar calibration target in the form of a checkerboard pattern to estimate the extrinsic pose between a color camera and depth sensor. Zhang and Zhang (2011) and Herrera and Kannala (2012) incorporate the available planarity constraint in their solution. In particular the measured depth values are optimized to have their z-coordinate vanish in the coordinate frame of the respective plane. Conceptually this is equivalent to the alignment residual r^A, which we introduce in the next section. Opposed to the planar geometry prior we rely on a sparsely sampled arbitrary surface; though, deviation of depth measurements to this surface are equally forced to vanish. The initialization step either assumes that the relative pose is small, or an initial estimate is provided via manual establishment of correspondences. The approach of Herrera and Kannala (2012) also considers the depth camera intrinsics and extends the calibration by a spatially varying correction term. Our calibration model of Sec. 4.2.1 is inspired by the exponential parameterization of the depth correction in Herrera and Kannala (2012).

A multi-camera, range sensor setup is considered by Geiger et al. (2012). The authors achieve calibration in a single shot via segmentation of present planar targets, followed by the enforcement of plane coherence during optimization. Vasconcelos et al. (2012) augment the constraints provided by a checkerboard pattern by line correspondences. The authors demonstrate that the alignment of 3 plane-line matches has at most 8 solutions, which leads to a minimal, closed form solution, which is easily exploited within a RANSAC framework. Mirzaei et al. (2012) particularly target the initialization of the relative pose, which is typically needed for consecutive non-linear refinement.

Target-less extrinsic calibration of a camera and a depth sensing device was performed by Scaramuzza et al. (2007) with the aid of manually provided correspondences. They especially focus on simplifying correspondence selection via user guidance. Levinson and Thrun (2013) and Moghadam et al. (2013) demonstrate the applicability of line based features. While Levinson and Thrun (2013) propose to leverage a distance transform representation for efficient residual computation during the registration process, Moghadam et al. (2013) utilize a homogeneous representation of lines that allows for a direct signed distance computation between points and lines. Extrinsic calibration for non-overlapping camera setups is provided by the work of Napier et al. (2013). Their approach requires that the same rigid scene is observed by each sensor

over time and hence they rely on a scene geometry prior similar to us.

Target-less *intrinsic* calibration combined with depth distortion correction has been proposed in the works of Teichman et al. (2013) and Zhou and Koltun (2014), which both aim for the joint optimization of the absolute camera poses and depth calibration. Teichman et al. (2013) use the fact that near range depth measurements are accurate and exploit a 3D reconstruction built therefrom as geometry prior for a subsequent sensor calibration. Contrary, Zhou and Koltun (2014) rely on initially established point correspondences between the depth measurements and hence are able to simultaneously refine the camera poses and depth camera distortion. Opposed to our method, both approaches do not consider present color images within their optimization. However, we claim that constraints in the image domain are more accurate compared to depth; therefore, our approach leverages a sparse map build from image feature correspondences.

4.2 Self Calibration of RGB-D Sensors

We aim for the automatic extrinsic (wrt. a color camera) and intrinsic calibration of a depth sensor. The resulting self-calibration procedure is necessary for on-line (re-)calibration or allows to immediately capture and process data of an ad-hoc camera setup, *e.g.*, a external depth sensor added to a standard camera. As an example application we demonstrate the improved 3D modeling capability of a commodity RGB-D sensor over the provided factory calibration in Sec. 4.4.

Our work is built on the observation that SfM reconstructions are typically much more accurate than depth measurements from Kinect like sensors. On the one hand, this is because images exhibit a high resolution nowadays (typically already in the mega-pixel range), which enables fine grained feature localization and leads to well constraint 3D point triangulations. On the other hand, loop closures are detected well during image-to-image matching and provide additional constraints for the reconstruction, which reduces drifting and allows for an accurate computation of the camera odometry. While loops can also be closed from point cloud data solely (Lu and Milios, 1997), image features are superior for geometrically simple scenes as often encountered indoors. These advantages of image over depth data have also influenced current state-of-the-art camera pose tracking algorithms (*e.g.*, Engel et al., 2014; Forster

et al., 2014) to predominantly rely on images. Though, depth data can help (Kerl et al., 2013), but such approaches require a registered sensor setup. Consequently, we argue that a sparse SfM reconstruction is accurate enough to serve as calibration target for the depth sensor and no manually provided constraints are required.

For our proposed algorithm, we require some prerequisites. First, the color camera and depth sensor need to have an overlapping field of view. For the subsequent joint usage of data this requirement needs to be fulfilled in any case and thus does not pose a particular restriction on the setup. Second, we make the simplifying assumption that there exists no motion between the capture of an image and depth map for a certain view, such that the relative pose between color and depth camera is constant over all views. This either means, that the sensors are synchronized, or that the sensor setup is static during exposure for a view (but can move between exposures). Third, the *color* camera intrinsics are known upfront to allow for a more accurate sparse reconstruction from image feature correspondences. It is conceivable to also optimize for the color camera intrinsics in exchange for a slightly degraded overall calibration accuracy. Since we aim for a good calibration of the depth camera, we chose to calibrate the color camera upfront with a well established, accurate calibration method (Bouguet, 2004). For clarity we want to re-emphasize that the depth camera intrinsics are variable in our approach.

As a result, given a sparse environment reconstruction, our approach follows the two objectives:

- First, we aim for the joint alignment of all measured depth maps to the sparse map. Since only the *global* poses for the color camera are known from the SfM model, but not for the depth sensor, the alignment needs to be performed in the local coordinate frame of each view; *i.e.*, it is not possible to build one concise point cloud from the depth measurements and align it to the SfM model. Due to the projective properties of SfM, the map is only correct up to scale, requiring the consideration of a global scale factor in the alignment procedure.
- Second, we want to compensate for inaccuracies and quantization artifacts in the measured depth, which are known to increase considerably with distance from the camera (Herrera and Kannala, 2012; Teichman et al., 2013). Again, the SfM model serves as a geometry prior, but for an accurate un-distortion the global alignment needs to be solved beforehand or jointly.

4.2.1 Calibration Model

Let us first motivate the calibration model that we chose to employ for the depth sensor. For clarity we denote points in camera coordinates as $\mathbf{X}$, while points in depth sensor frame are given by $\mathbf{X}'$.

First, we account for the relative transformation between the camera and depth sensor, such that a 3D point in the depth sensor coordinate system $\mathbf{X}'$ is obtained from a point $\mathbf{X}$ in the camera coordinate frame via

$$\mathbf{X}' = \mathrm{T}_{rel} \circ s\mathbf{X} = s\mathrm{R}_{rel}\mathbf{X} + \mathbf{t}_{rel} \ . \tag{4.1}$$

The global scaling s considers the scale ambiguity of a SfM reconstruction. The transformation and scaling is independent of the motion of the whole sensor setup and thus consistent among all views.

Second, we model the projection of 3D points into pixel coordinates via radial and tangential lens distortion according to Eq. (2.8) in Sec. 2.2, followed by a standard perspective projection

$$\mathbf{u} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \end{bmatrix} \begin{pmatrix} \mathbf{x}'_d \\ 1 \end{pmatrix} \ , \tag{4.2}$$

jointly denoted as $\mathbf{u} = \pi_D(\mathbf{X}')$ in the following.

Third, for each continuous pixel location $\mathbf{u}$ and depth d in the feasible (*i.e.*, measurable depth) range, we model a depth distortion offset as $\delta(\mathbf{u}, d)$. This means, that for each 3D position in the view frustum there exists an offset that we aim to solve for. To obtain a low parametric representation we leverage the observations that the offset (i) will increase with distance from the camera and (ii) will only vary smoothly (Herrera and Kannala, 2012). Thus, we model δ as an exponential function over the depth as

$$\delta(\mathbf{u}, d) = c(\mathbf{u}) \cdot \exp(a_0 - a_1 d) \ . \tag{4.3}$$

The unknowns are a_0, a_1 and the pixel dependent scaling $c(\mathbf{u})$. The latter is based on a lattice $\boldsymbol{\beta} \in \mathbb{R}^{M \times N}$ exhibiting a significantly lower resolution than the full depth map. Intermediate values are therefore obtained via bicubic

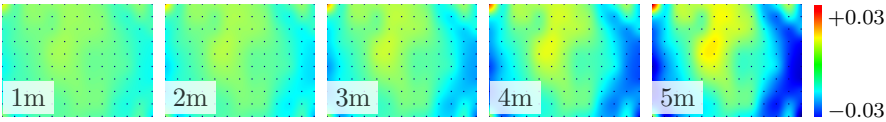


Figure 4.2: Computed depth distortion pattern, representing the offset in the inverse depth parameterization for different distances. The dots mark the lower dimensional lattice $\boldsymbol{\beta}$ that is optimized in our algorithm.

interpolation of distinct values $\beta_{m,n}$ as

$$c(\mathbf{u}) = \sum_{m,n} \gamma_{m,n}(\mathbf{u}) \cdot \beta_{m,n} = \mathrm{vec}(\boldsymbol{\gamma}(\mathbf{u}))^{\mathsf{T}} \mathrm{vec}(\boldsymbol{\beta}) \ , \tag{4.4}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{M \times N}$ denotes the appropriate interpolation coefficients. The interested reader is referred to Appendix 4.A for a derivation of the computation of bicubic interpolation coefficients. Typically $M = 13$, and $N = 9$ in our experiments (but only 16 coefficients in $\boldsymbol{\gamma}$ are non-zero due to the nature of bicubic interpolation) and a result of the offset is illustrated in Fig. 4.2.

As a result, our full calibration model has 134 degrees of freedom: 7 for the extrinsic relative motion, 8 for the lens distortion and camera intrinsics, and 117+2 unknown variables for the depth distortion offsets.

4.2.2 Problem Formulation

With the calibration parameterization at hand, we now turn to the actual problem formulation. We model the calibration task as the non-linear least squares optimization

$$\min_{\theta} \sum_{i,j} \rho_I \left(\left\| \mathbf{r}^I_{ij}(\theta_I) \right\|^2 \right) + \lambda \sum_{i,j} \rho_D \left(\left\| r^A_{ij}(\theta) \right\|^2 \right) + \lambda \sum_{j,k} \rho_D \left(\left\| r^D_{jk}(\theta) \right\|^2 \right) \tag{4.5}$$

consisting of three different data fidelity terms $\mathbf{r}^I$, r^A, and r^D, explained in the following and visualized in Fig. 4.3. The residuals are computed for the i^{th} 3D point and its observation in the j^{th} camera as obtained from the SfM reconstruction, as well as for view pairs between views j and k.

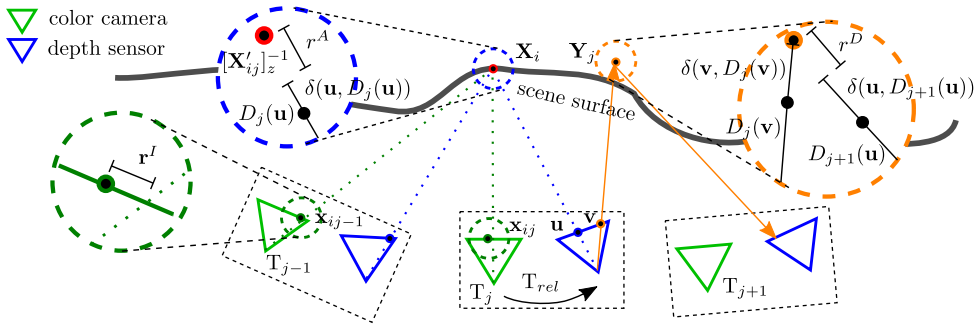


Figure 4.3: Visualization of the three residual terms for the reprojection error $\mathbf{r}^I$ (Eq. 4.6), the joint alignment error r^A between 3D points of the SfM reconstruction and measured depth maps (Eq. 4.8), and the view pair registration error r^D (Eq. 4.9).

The optimized parameters $\theta = \{\theta_D, \theta_I\}$ correspond to previously explained calibration model $\theta_D = \{s, \mathrm{T}_{rel}, \pi_D, \boldsymbol{\beta}, \mathbf{a}\}$ and the set of 3D points and camera poses $\theta_I = \{\{\mathbf{X}_i\}, \{\mathrm{T}_j\}\}$ which are initialized by means of the sparse SfM map. The weight parameter λ compensates for the different error modalities and can be set such that *e.g.*, a reprojection error of 1 pixel incurres the same penalty as a depth misalignment of 1mm at 1m distance from the camera.

Image reprojection error: The first residual models the image reprojection error between projected 3D points and their image observations $\mathbf{x}_{ij}$ as

$$\mathbf{r}^I_{ij}(\theta_I) = \mathbf{x}_{ij} - \pi_I(\mathrm{T}_j \circ \mathbf{X}_i) \ , \tag{4.6}$$

and is thus not dependent on the calibration. Due to the projective mapping of 3D points also the absolute model scale is irrelevant. The residual prevents (strong) deformations in the 3D structure and odometry and hence can be seen as a regularization for our calibration. Similar to the depth camera calibration model, it considers radial and tangential distortion of the color camera (though, the parameters are assumed to be known and thus not optimized). One could argue to keep the reconstruction fixed at all. However, we found that including it in the optimization improves the calibration. This is due to the fact that

the sparse map is not perfect itself, but contains outliers (*e.g.*, due to wrong correspondences), which will be compensated for by more accurate depth measurements of the actual geometry. To limit the influence of outliers, each residual is weighted according to a robust Huber (cf., Huber and Ronchetti, 2009) cost function ρ_I.

Depth map alignment to the sparse map: The second residual performs the joint alignment of each depth map D_j to the sparse map in the local coordinate frame of the (depth camera) view. Therefore, we transform 3D points (observable in the j^{th} view) into the coordinate frame of the depth sensor,

$$\mathbf{X}'_{ij} = \mathrm{T}_{rel} \circ s(\mathrm{T}_j \circ \mathbf{X}_i) \ , \tag{4.7}$$

and aim to minimize their deviation from the recorded depth measurements according to

$$r^A_{ij}(\theta) = [\mathbf{X}'_{ij}]^{-1}_z - D_j(\mathbf{u}) - \delta(\mathbf{u}, D_j(\mathbf{u})) \quad \text{s.t.} \quad \mathbf{u} = \pi_D(\mathbf{X}'_{ij}) \ . \tag{4.8}$$

Since the depth measurement accuracy is known to decrease with distance from the camera (*e.g.*, Teichman et al., 2013) we employ an inverse depth parameterization (Civera et al., 2008). For example, this results in an equal weighting of a 1cm error at 1m and a 25cm error at 5m distance and thus naturally accounts for the close range accuracy. As a result both, the depth measurements D_j as well as the offset δ are parametrized over inverse depth (denoted as d in this paper for simplicity). Eq. 4.8 actually resembles a point-to-plane data association, which is common in ICP and proven to work well for 3D model alignment (Rusinkiewicz and Levoy, 2001). Recently, Maier et al. (2014) also proposed a depth augmented bundle adjustment formulation; however, they directly model the alignment as 3D euclidean distance, which is not applicable with our calibration model and does not consider the varying depth accuracy.

Above residual compensates for distortions in depth by means of δ. The 3D model points $\mathbf{X}'_{ij}$ substitute the planar depth calibration pattern used in previous work (*e.g.*, Herrera and Kannala, 2012) and constrain the solution. Therefore, outliers in the sparse model will degrade the solution quality. This

motivates the use a Tukey cost function (Huber and Ronchetti, 2009) – which has constant penalty beyond a certain residual – for the loss function ρ_D. This particular choice is also required due to the presence of missing depth measurements and is further motivated in Sec. 4.3.2.

Depth map alignment between views: The third residual term only considers the depth measurements. Its goal is to enforce alignment between overlapping depth maps, *i.e.*, views that capture the same scene part. Registration between depth maps could be achieved by minimizing the euclidean distance between corresponding 3D points unprojected from the depth maps. However, neither do we know explicit correspondences, nor are the depth camera intrinsics fixed (which would be needed for the unprojection). As a result, we include the full unprojection $\pi_D^{-1}(\mathbf{u}, d)$ of a depth measurement from one view (j), transform the obtained point $\mathbf{Y}'_j$ to the second view (k) and define the error therein as the deviation between transformed and measured inverse depth. Formally this is

$$
\begin{aligned}
r_{jk}^{D}(\theta) = [\mathrm{T}_{jk} \circ \mathbf{Y}'_j]_z^{-1} - D_k(\mathbf{u}) - \delta(\mathbf{u}, D_k(\mathbf{u})) \qquad (4.9)\\
\text{s.t.} \qquad \mathbf{u} = \pi_D(\mathrm{T}_{jk} \circ \mathbf{Y}'_j)\\
\mathrm{T}_{jk} = \mathrm{T}_{rel} \circ s \circ \mathrm{T}_k \circ \mathrm{T}_j^{-1} \circ \frac{1}{s} \circ \mathrm{T}_{rel}^{-1}\\
\mathbf{Y}'_j = \pi_D^{-1}(\mathbf{v}, D_j(\mathbf{v}) + \delta(\mathbf{v}, D_j(\mathbf{v})) \ .
\end{aligned}
$$

Thereby the correction term δ is considered for both views. For determining overlapping view pairs j, k we leverage the covisibility information encoded in the point observations of the SfM model, *i.e.*, we consider all pairs of views jointly observing a 3D point. In addition we only enforce alignment for views that overlap by at least 50%. To determine the overlap ratio r_o we approximate the relative pose T_{rel} via the initially provided estimate and warp a depth map from one view to the other. Then for each view pair we sample $50 \cdot r_o$ randomly distributed 2D points in the first view and define them as the points $\mathbf{v}$ of Eq. 4.9. The same procedure is repeated for the permuted view pair (*i.e.*, the role of D_j and D_k is interchanged), such that we obtain a bi-directional registration objective. Note, that the sparsification only reduces computational cost and one can also consider a dense depth-to-depth alignment.

4.3 Optimization

Optimization of the objective function in Eq. 4.5 within a non-linear least squares framework requires the computation of Jacobians for all residuals wrt. the optimization parameters θ.

To obtain an analytic expression for the 3D transformation derivatives, we parameterize the pose T via the 6 DoF vector $\boldsymbol{\xi} = (\mathbf{w}, \mathbf{t})$, where $\mathrm{T}(\boldsymbol{\xi}) = [\mathrm{R},\, \mathbf{t}]$ with R being a member of the special orthogonal group $\mathtt{SO}(3)$, *i.e.*, $\mathrm{R} = e^{[\mathbf{w}]_\times} \in \mathtt{SO}(3)$, $[\mathbf{w}]_\times \in \mathtt{so}(3)$, and $\mathbf{w}, \mathbf{t} \in \mathbb{R}^3$. The derivative of a point transformation $\mathrm{T} \circ \mathbf{X}$ wrt. its parameters is then written as

$$\frac{\partial \mathrm{T}(\boldsymbol{\xi}) \circ \mathbf{X}}{\partial \boldsymbol{\xi}} = \left[-\mathrm{R}[\mathbf{X}]_\times,\, \mathrm{I}_{3\times 3}\right] \in \mathbb{R}^{3\times 6} \quad . \tag{4.10}$$

The Jacobian of the depth distortion δ wrt. its sub-pixel location $\mathbf{u}$, the measured inverse depth d and the calibration lattice $\boldsymbol{\beta}$ is

$$\left[\frac{\partial \delta}{\partial \mathbf{u}},\, \frac{\partial \delta}{\partial d},\, \frac{\partial \delta}{\partial \boldsymbol{\beta}}\right] = \left[\mathrm{vec}(\boldsymbol{\beta})^\intercal \frac{\partial \mathrm{vec}(\boldsymbol{\gamma})}{\partial \mathbf{u}},\, -a_1 c(\mathbf{u}),\, \boldsymbol{\gamma}(\mathbf{u})\right] e^{a_0 - a_1 d} \quad . \tag{4.11}$$

Thereby $\boldsymbol{\gamma}(\mathbf{u})$ are the interpolation coefficients and their respective derivative $\partial\boldsymbol{\gamma}/\partial\mathbf{u} \in \mathbb{R}^{|\boldsymbol{\beta}|\times 2}$ is easy to obtain from the bi-cubic interpolation equations itself (see Appendix 4.A for details). As a result the derivative of the correction term at a projected pixel location $\mathbf{u}$ is

$$\frac{\partial \delta(\mathbf{u}, D(\mathbf{u}))}{\partial \mathbf{u}} = \left(\mathrm{vec}(\boldsymbol{\beta})^\intercal \frac{\partial \mathrm{vec}(\boldsymbol{\gamma})}{\partial \mathbf{u}} - a_1 c(\mathbf{u}) \nabla_{\mathbf{u}} D\right) e^{a_0 - a_1 D(\mathbf{u})} \quad . \tag{4.12}$$

The residual term r^D contains the computation of the inverse projection function $\pi_D^{-1}(\cdot)$. It can not be solved in closed form due to the present lens distortion and we resort to an iterative procedure for the evaluation of the inverse lens distortion (*e.g.*, Civera et al., 2012, p132). Since this renders the Jacobian computation difficult, we set $\partial \mathbf{Y}'_j / \partial \pi_D$ to zero. This has the effect that updates on the camera parameters and on the depth correction δ are only due to the projection in the second view and independent from the unprojection in the first view. Since the registration is formulated bi-directionally, we achieved good convergence with this approximation.

For our implementation we make use of the Ceres solver library (Agarwal

et al., 2015) and leverage its automatic differentiation for the projection function Jacobian $\pi(\mathbf{X})/\partial\mathbf{X}$. Given all partial derivatives, the overall residual Jacobians are straight forward to compute by following the chain rule, *i.e.*, the derivative of a function $f(g(\mathbf{x}))$ is the product of its partial derivatives according to $\partial/\partial\mathbf{x}\, f(g(\mathbf{x})) = \partial f/\partial g\ \partial g/\partial\mathbf{x}$.

Our optimization then follows a three step procedure, which first sequentially solves for the extrinsic pose and depth intrinsics, and thus guides the optimization towards the correct solution. The sparse map is assumed to be optimal wrt. the reprojection residuals $\mathbf{r}^I$ already. The individual steps are:

(i) Optimization over model scale, relative pose and depth camera intrinsics, while the sparse map and the depth correction term are kept constant.

(ii) Exclusive optimization of the correction term δ. This step can exploit the sparsity introduced by the interpolation coefficients $\boldsymbol{\gamma}(\mathbf{u})$, since 3D point projections $\mathbf{u}$ do not vary and hence the subset of parameters in $\boldsymbol{\beta}$ that are affected by a particular residual are known upfront.

(iii) A final refinement over all parameters θ, including the 3D points and absolute view poses of the sparse map. Since the 3D point projections will vary during optimization, it is required to compute a Jacobian wrt. all parameters in $\boldsymbol{\beta}$ for each residual. This is computational more expensive than in the previous step; however, only a few iterations are typically required until convergence.

Note that our consecutive optimization is similar to the alternating optimization procedure of Herrera and Kannala (2012), which also solves for relative pose and depth distortion. With our approach a single optimization until convergence for the first two steps turned out to be sufficient to initialize the final refinement.

4.3.1 Initialization

For reconstruction of the sparse map we leverage the publicly available Sparse Bundle Adjustment (SSBA, Zach, 2014) and use SIFT features (Lowe, 2004) as image observations. The color camera calibration is computed upfront with the help of a checkerboard pattern to obtain an accurate model.

The scale of the computed SfM reconstruction will be arbitrarily off from the correct, real world scale. However, an initial scale that is close to the

real scale is important for our procedure to converge correctly. Making the approximating assumption that captured images and depth maps are registered already, *i.e.*, that $\mathrm{T}_{rel} = [\mathrm{I}, \mathbf{0}]$, allows to evaluate the scale difference between the depth $[\mathrm{T}_j \circ \mathbf{X}_i]_z$ of computed 3D points in the individual views and the actually measured depth at the projected pixel locations. By taking the median over these computed scales we can get a fairly good estimate that is within a few percent of the correct scale.

The initial value for the relative pose is typically set to identity, which works well forsmall baseline sensors such as Kinect. In case the relative motion is large, one can resort to approaches that utilize mutual information (Pandey et al., 2012) or line features (Moghadam et al., 2013) between image an depth data and compute an initial transformation in an automatic manner.

4.3.2 Handling of Missing Depth Measurements

Reflective surfaces, bright light, or geometry that is outside the measurable depth range, lead to missing values within the depth measurements. This raises the question how to correctly handle the case when a 3D point projects into such an unobserved region, or equivalently outside the depth map domain. Simply setting the residual to zero is misleading, since then the optimal solution would be an invalid configuration where the depth camera does not see the reconstructed geometry in any view. As an alternative we follow the approach of Li et al. (2008) and set the (inverse) depth in the unobserved regions in a preprocessing step to twice the maximal measurable value. A local smoothing of the boundary region prevents discontinuities and guarantees smooth derivatives. The Tukey loss function ρ_D introduced in Sec. 4.2.2 is particularly suited for this scenario. Its constant penalty for large errors results in a large residual, but a vanishing Jacobian. Hence, those regions have no influence in the optimization and invalid configurations are not favored as solutions.

We also experimented with a lifted kernel loss function as used in Li et al. (2008); Zach (2014), but were not able to observe any improvements. As the computational complexity increases due to the introduction of a variable weight for each residual, we propose to utilize the more efficient Tukey loss.

Dataset	#points	#views	#obs	#view pairs	#projections
STATUE	22,970	196	157,966	1,555	62,640
WALL	30,376	187	212,443	2,067	89,736
RELIEF	25,998	104	208,500	2,205	96,654

Table 4.1: Datasets for evaluation and their characteristics. From left to right: size of the sparse map, number of contained views, total number of 3D point observations in all views, utilized view pairs with sufficient overlap, and the established links between those view pairs.

4.4 Experiments and Results

To demonstrate the effectiveness of our proposed algorithm, we have recorded three datasets with an Asus Xtion Pro Live RGB-D sensor and conducted different experiments. The characteristics of the datasets are listed in Tab. 4.1. We chose this sensor over a self assembled setup, because it is widely available and thus fosters reproducibility. In addition, it comes with a decent factory calibration which allows to evaluate our improvement over it. The range of valid depth measures is set to the interval from 0.5 to 5 meters for all experiments. The run-time of our algorithm for the different datasets is in the order of 30 seconds to 2-3 minutes.

4.4.1 Relative Pose and Intrinsic Calibration

In the first experiment we evaluate the accuracy of the computed relative pose. Due to the absence of ground truth correspondences between image and depth data (which would allow a quantitative analysis), a visual comparison is provided. In order to do so, we transform the depth map into a mesh (also considering the optimzed depth camera intrinsics and depth distortion) and render it into the viewpoint of the color camera. For an accurate relative motion, geometry boundaries that are well visible in both modalities need to aligned. In Fig. 4.4 and also in Fig. 4.1 the result of our method are compared to the factory calibration and a calibration that is obtained for the sensor's infrared (IR) camera (which is the camera the depth measurements are provided for). The latter is achieved with the help of a checkerboard pattern and the utilization of the stereo setup between the color and IR camera.

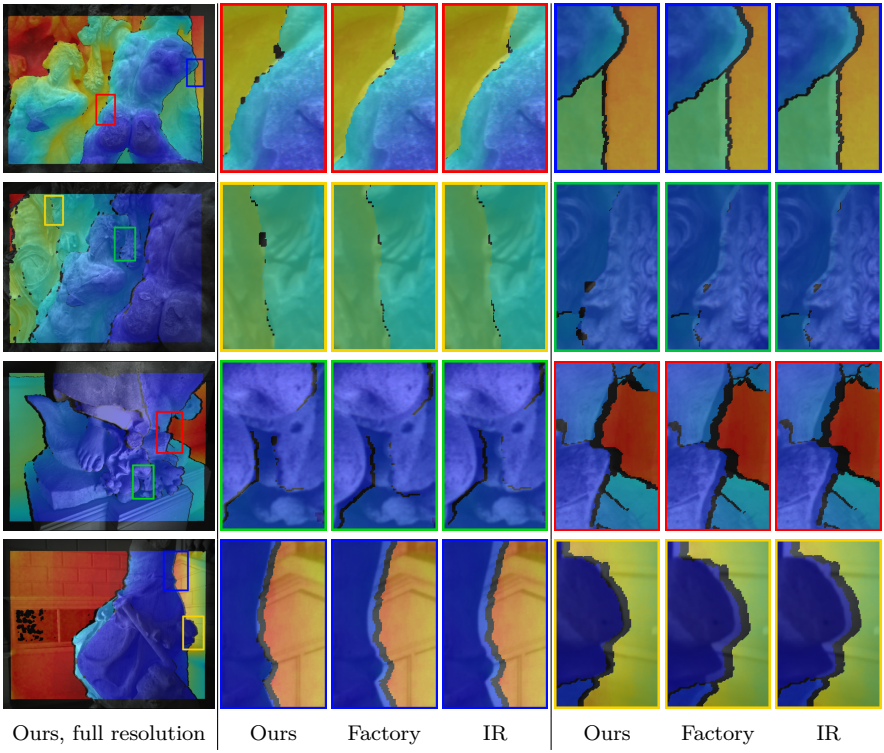
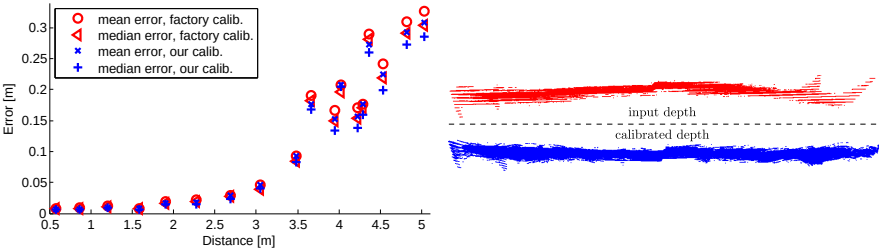


Figure 4.4: Evaluation of the computed relative pose between color and depth camera. The depth map is rendered into the viewpoint of the color camera, such that image and geometry boundaries will align well under a correct calibration. The full resolution images and the different cutouts show the image, overlaid with the warped, color coded depth map. It is clearly visible that both, the factory calibration as well as a calibration only considering the IR images exhibit a misalignment in the order of a few pixels. In contrast, our method shows an accurate overlay with denotes a correct calibration.



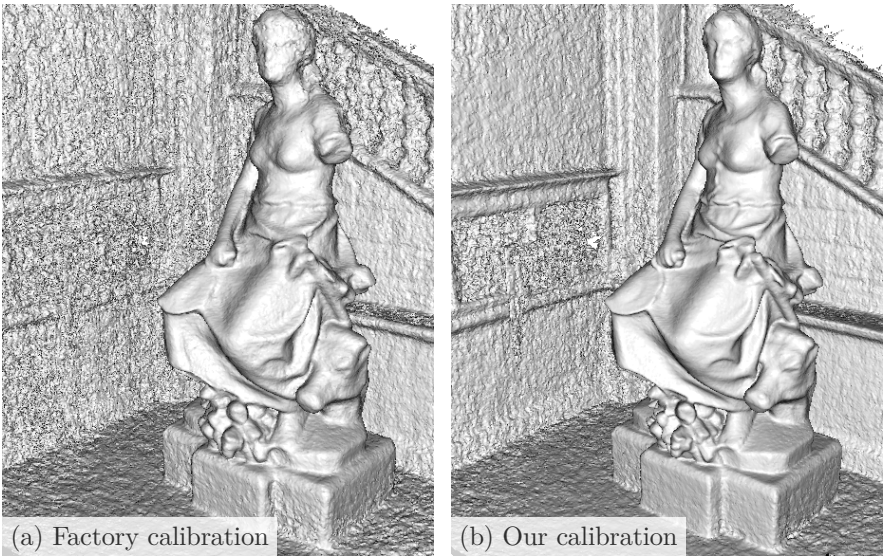
(a) Deviation of points to the fitted plane. (b) Effect of un-distortion with our calibration of a flat scene.

Figure 4.5: Given recordings of a planar surface at various distances, (a) illustrates the mean and median deviation of measured and undistorted 3D points wrt. a fitted plane (a large error remains due to quantization effects). (b) visualized the effect of un-distortion with our calibration for a recorded plane at roughly 4 meters distance.

The overlays of warped depth maps and images demonstrate that both, the factory and IR based extrinsics exhibit a small, but noticeable error, while the alignment based on our calibration is the most accurate.

4.4.2 Depth Correction Term

A result for the obtained depth distortion correction is illustrated in Fig. 4.2 for different distances from the camera. The circular error pattern which had already been noticed by Herrera and Kannala (2012); Teichman et al. (2013) for this kind of sensor is well visible. To obtain a quantitative measure of the depth (un-)distortion effect we recorded a planar scene at various distances. Then a plane is fitted to the corresponding point cloud and the average and median distance of points to the plane is evaluated. The resulting error should be the smaller – and the unprojected depth map more planar (cf. Fig. 4.5(b)) – the better the calibration models the true depth camera characteristics. Fig. 4.5(a) shows the results we obtain compared to the factory calibration. It is clearly visible that (i) the distance is reduced especially for the far range and (ii) the reduction is consistent over the whole depth range, *i.e.*, already accurate near range values are not wrongly distorted. The large errors further



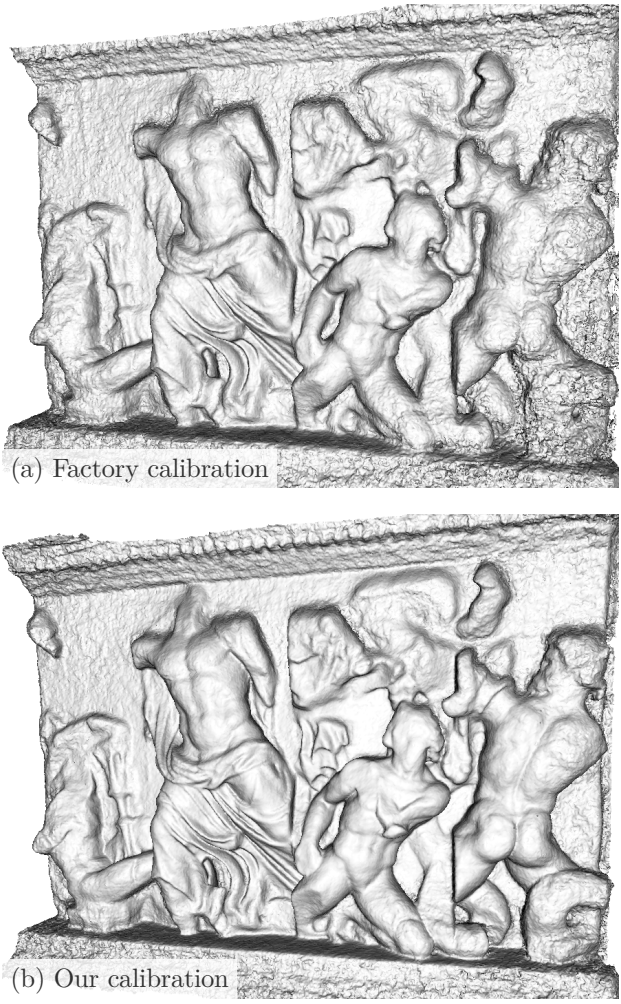
(a) Factory calibration (b) Our calibration

Figure 4.6: Volumetric fusion result for the STATUE dataset. Note the more accurate geometry of *e.g.*, the face , around the hip, or in the background.

from the camera are due to the present quantization in depth which can not be eliminated.

4.4.3 Reconstructed Models

Finally, we quantitatively compare the reconstruction quality that is achievable by utilizing our calibration. To this end, we first undistorted each depth map and then perform volumetric fusion in a 512^3 sized voxel grid. The influence of erroneous depth measurements is limited via the usage of a truncated signed distance function (*e.g.*, Curless and Levoy, 1996; Zach et al., 2007; Newcombe et al., 2011b) for implicit surface representation. In addition we widen the zero crossing by the expected quantization in depth (similar to what has been observed in Smisek et al. (2011)). This has the desired effect that close range measurements define the surface location more precisely than far range measurements. Figures 4.1, 4.6, and 4.7 illustrate the models that are obtained



(a) Factory calibration

(b) Our calibration

Figure 4.7: 3D model for the RELIEF dataset. The difference in reconstruction quality is solely due to the depth sensor calibration and particularly noticeable for the far range of the sensor (*e.g.*, right side of model).

after a surface mesh extraction via Marching Cubes (Lorensen and Cline, 1987). We can conclude that our calibration improves the reconstruction quality and enables to obtain less noisy, more detailed 3D models. We do not perform any regularization for the final model; thus, the improvement solely stems from the more accurate calibration.

4.5 Discussion

In this chapter we have presented an auto-calibration approach for RGB-D sensors that enables to automatically determine both, the extrinsic and intrinsic parameters of the sensor setup. Our method is based on the observation, that sparse maps computed via SfM or SLAM typically are quite accurate and thus can be leveraged as a geometry prior for the calibration. This is an important observation, since it makes artificial calibration targets dispensable and self-calibration possible.

There are certainly several approaches conceivable to incorporate a sparse 3D model in the actual calibration method. The optimization strategy we introduced, jointly minimizes the alignment error between the sparse map and all recorded scene depths as well as between the depth maps themselves. This has been shown to provide a calibration that leads to an improved alignment between image and depth measurements and more accurate 3D models.

4.A Bicubic Interpolation

Bi-cubic interpolation provides a method for smooth interpolation of data points on a two dimensional regular grid. The derivation of the two-dimensional interpolation coefficients is directly obtained from the underlying interpolation via spline functions in one dimension. The so called Cubic Hermite spline function consists of 4 polynomials of degree three and is dependent on the particular data point values and their first order derivatives. In case derivatives are not available, they can be numerically estimated from the data points itself. For a uniform parameter spacing this leads to a Catmull-Rom spline (Catmull and Rom, 1974), which will be considered in the following.

For a desired *one-dimensional* interpolation point x' we can obtain a corresponding point x (by subtraction of the lower integer value) that lies in the interval [0 1], *i.e.*, between two data points (or pixels in our application). Then, the vector $\mathbf{p} = [p_{-1}, p_0, p_1, p_2]^\mathsf{T}$ contains the four data points surrounding x on both sides. Border cases are handled via mirroring of data values in our case. The interpolated value $p(x) = \boldsymbol{\kappa}(x)^\mathsf{T}\mathbf{p}$ is solely dependent on the interpolation coefficients $\boldsymbol{\kappa}(x)$ as defined in the following. Their derivatives wrt. x are also trivial to compute:

$$\boldsymbol{\kappa}(x) = \frac{1}{2}\begin{bmatrix} -x^3 + 2x^2 - x \\ 3x^3 - 5x^2 + 2 \\ -3x^3 + 4x^2 + x \\ x^3 - x^2 \end{bmatrix} \in \mathbb{R}^4 \quad \text{and} \quad \frac{\partial \boldsymbol{\kappa}(x)}{\partial x} = \frac{1}{2}\begin{bmatrix} -3x^2 + 4x - 1 \\ 9x^2 - 10x \\ -9x^2 + 8x + 1 \\ 3x^2 - 2x \end{bmatrix} . \tag{4.13}$$

Consequently the derivatives of the interpolated value wrt. the interpolation point x is $\partial p(x)/\partial x = \partial \boldsymbol{\kappa}(x)^\mathsf{T}/\partial x\, \mathbf{p}$.

In the *two-dimensional* case we need to consider a grid of data points

$$\mathrm{Q} = \begin{bmatrix} q_{-1,-1} & q_{0,-1} & q_{1,-1} & q_{2,-1} \\ q_{-1,0} & q_{0,0} & q_{1,0} & q_{2,0} \\ q_{-1,1} & q_{0,1} & q_{1,1} & q_{2,1} \\ q_{-1,2} & q_{0,2} & q_{1,2} & q_{2,2} \end{bmatrix} \in \mathbb{R}^{4\times 4} , \tag{4.14}$$

which are located around the desired interpolation point $\mathbf{u} = (u, v)^\mathsf{T}$. Again we assume that the interpolation location is transformed such tat it lies within

the unit interval, *i.e.*, $\mathbf{u} \in [0\ 1] \times [0\ 1]$. Similarly to the one-dimensional case, the interpolated value is obtained via element-wise multiplication of Q with the interpolation coefficients $\boldsymbol{\gamma}$, *i.e.*:

$$q(\mathbf{u}) = \sum_{m,n} \gamma_{m,n}(\mathbf{u}) \cdot q_{m,n} = \operatorname{vec}(\boldsymbol{\gamma}(\mathbf{u}))^{\mathsf{T}} \operatorname{vec}(\mathrm{Q}) \ . \tag{4.15}$$

The interpolation coefficients itself are directly dependent on the Catmull-Rom spline in horizontal and vertical direction, such that

$$\boldsymbol{\gamma}(\mathbf{u}) = \boldsymbol{\kappa}(v)\boldsymbol{\kappa}(u)^{\mathsf{T}} \in \mathbb{R}^{4\times 4} \ . \tag{4.16}$$

With this relation at hand, it is visible that the derivative of the interpolation coefficients also factors into a product containing the one-dimensional derivatives and is computed to

$$\frac{\partial \operatorname{vec}(\boldsymbol{\gamma}(\mathbf{u}))}{\partial \mathbf{u}} = \left[\operatorname{vec}\left(\boldsymbol{\kappa}(v) \frac{\partial \boldsymbol{\kappa}(u)^{\mathsf{T}}}{\partial u} \right), \operatorname{vec}\left(\frac{\partial \boldsymbol{\kappa}(v)}{\partial v} \boldsymbol{\kappa}(u)^{\mathsf{T}} \right) \right] \in \mathbb{R}^{16\times 2} \ . \tag{4.17}$$

Finally, this reveals that the derivative of the cubic interpolated value on a two-dimensional grid at location $\mathbf{u}$ is defined according to

$$\frac{\partial q(\mathbf{u})}{\partial \mathbf{u}} = \operatorname{vec}(\mathrm{Q})^{\mathsf{T}} \frac{\partial \operatorname{vec}(\boldsymbol{\gamma}(\mathbf{u}))}{\partial \mathbf{u}} \in \mathbb{R}^2 \ . \tag{4.18}$$

5 Viewpoint Invariant 3D Model Registration

Once partial models have been built we are interested in their fusion for accurate global environment modeling. This chapter examines two different approaches for relative pose estimation between 2.5D scans under strong viewpoint variations. Both exploit the underlying scene geometry for correspondence search and by this retain the discriminative power of image features that is normally lost due to perspective distortions.

Utilizing image based features to compactly describe image content or to identify corresponding points in images has become a de facto standard in computer vision over recent years. However, a major problem when trying to find correspondences between widely separated views is that the appearance of objects can change drastically with viewpoint. To remedy this problem techniques have been developed which normalize images or image regions such that they become (at least approximately) invariant to viewpoint changes. In case one matches two images against each other the most popular method is to use local image features (cf. Sec. 2.3.1) that compensate for the first order effects of viewpoint change by normalization, *i.e.*, affine transformations. Since scale, orientation and (anisotropic) stretch are all effects that could have been caused by a viewpoint change, they need to be factorized out and thus it is not possible to distinguish, *e.g.*, real-world circles from ellipses or a small round dot from a huge sphere any more. This is a general dilemma of discriminative power vs. invariance.

When geometry information is available (*e.g.*, from a depth sensor, laser scanner or from vanishing points in a Manhattan scenario) one can normalize wrt. the given 3D structure by moving a virtual camera to a frontal view and then rendering a canonical representation of a local image feature. However, this rectification process imposes the strong limitation that it either requires an affine detector and thus the number of features obtained is limited (Köser

Figure 5.1: Examples of developable surfaces present in our environment. Note, that in the image on the left only cones are highlighted, although planes and cylinders exist as well.

and Koch, 2007), or relies on the existence of dominant scene planes (Wu et al., 2008; Robertson and Cipolla, 2004; Cao and McDonald, 2012).

In contrast to this assumption we observe that many structures in our environment are also curved, *e.g.*, like cylinders, cones or consist of free-form shapes. First, many man-made objects are made by bending sheets or plates and thus - by construction - form *developable surfaces* that can virtually be "unrolled" when their geometric structure is known (see Fig. 5.1). We follow the idea to

> *develop such observed scene surfaces and to extract image features in the flat 2D wall-paper version of that very same surface.*

Second, for free-form surface we can not pose any requirements on the presence of particular geometric shapes. Thus, we propose to

> *become independent of the original sensor viewpoint by exploiting characteristic* salient directions *of the scene, which are repeatable among different scans.*

Examples include peaks in the distribution of the surface normals, vanishing points, symmetry, gravity or other directions that can be reliably obtained from the sensor or the scene. Each salient direction is then exploited to render an orthographic view, and by this way removing the perspective effects that had been introduced by the particular sensor position. In summary, we see our contributions as follows:

– We depict the detection and extraction of developable surfaces as well as

salient directions from RGB-D scans, and illustrate their usage for the generation of normalized image representations.

– It is demonstrated that the generated images are identical (for jointly seen Lambertian scene parts) up to a 2D similarity transformation, which renders them suitable for the use of established scale and rotation invariant local detectors and descriptors (such as SIFT, (Lowe, 2004)).

– In case scale is known (*e.g.*, from a Kinect camera or laser scanner) the perspective invariance requirement of the original problem is limited to a rotation in the image plane and is reduced even further for surfaces such as cones or cylinders or if the gravity direction is known.

– By this, we can relate more features to each other and achieve an improved registration performance.

The remainder of this chapter will give an overview of feature normalization approaches employed in the literature in Sec 5.1. We then illustrate the concept of developable surfaces and their utilization to obtain viewpoint invariance in Sections 5.2 and 5.3, together with results in Sec. 5.4. Obtaining viewpoint invariance by means of salient directions is discussed in Sections 5.5 and 5.6. Sections 5.7 and 5.8 cover our proposed pose estimation strategy for scan alignment and the experimental evaluation, respectively. The chapter concludes with a discussion in Sec. 5.9.

5.1 Review

In Sec. 2.8 we have already presented a broader overview of 3D model registration algorithms. Here we want to concentrate on the setting where we have given explicit correspondences between models and will argue that image based registration is superior over purely geometry based alignment approaches. For the joint usage of texture and structure information we assume that the calibration of the capture system is given, so that a RGB-D scan can be created where range and image data share the same center of projection. A possible solution to automatically obtain the requited calibration was presented in the previous Sec. 4. For determining the system's pose at two largely different positions with different orientations, related work can be classified into three categories.

Appearance based viewpoint invariance: Appearance can change drastically wrt. changes in viewpoint and also illumination as shown by Kaneva et al. (2011). Therefore, purely image based approaches build upon features which are approximately invariant against perspective distortion, such as affine features (Mikolajczyk et al., 2005; Mikolajczyk and Schmid, 2005), or – to a lesser degree – SIFT Lowe (2004) and variants thereof. For an in depth discussion of invariant feature constructions we refer the reader to work of Van Gool et al. (1995). The utilization of an affine normalization leads to a loss in discriminative power, *e.g.*, one can no longer distinguish the appearance of real world circles and ellipses (Köser and Koch, 2007). Moreover, the need for affine invariance implies strong requirements on the local region, which results in considerable less reliably detected features compared to simpler detectors (such as Harris corners). In addition the affine detector is taking substantially more time.

In comparison, the repeatability of features is significantly improved, if the image is first normalized (*e.g.*, via geometry information; see below) and then keypoints are detected (Wu et al., 2008).

Geometry based viewpoint invariance: Approaches using geometry descriptors have been shown to work on 3D scenes (*e.g.*, Johnson and Hebert, 1999; Yamany and Farag, 2002; Lo and Siebert, 2009; Rusu et al., 2009a). A key difficulty is the estimation of the position, scale and orientation in 3D space where to compute the descriptor, *i.e.*, the detection of good 3D features. Several detectors have been proposed (see for example Vanden Wyngaerd and Van Gool, 2002; King et al., 2005; Holzer et al., 2012), but a major dilemma in 2.5D, as opposed to real 3D, is as follows: A useful point for matching requires a repeatable detection; thus, surface parts need to be seen also from another, widely different viewpoint. However, this repeatability is likely to decrease with increasing surface complexity because of self-occlusions. On the other hand, for less complex surfaces, like nearly flat regions, the exact localization is sensitive to noise and the local geometry is not discriminative for matching. An alternative to 3D feature detectors is to densely sample the surface, leading to a very high number of descriptors (Johnson and Hebert, 1999) that need to be handled in matching and verification.

Methods obtaining dense point correspondences via a neighborhood search, such as ICP (see Sec. 2.8.1), are likely to get stuck in local minima. In the Great

Buddha project (Ikeuchi et al., 2007) the point-to-point distance is augmented by reflectance values of laser scans in order to be more discriminative. Besides sparse and dense point correspondences, other geometric features such as lines and planes (Stamos and Leordeanu, 2003) are used to obtain potential matches between individual 3D models.

Joint appearance and geometry utilization If both, image data and the underlying geometry is available, recent results show that it pays of to leverage both modalities jointly. To address the mentioned problems of image feature (loss of discriminative power, detection repeatability), images or local image regions are rectified with respect to the present geometry before image feature matching. For planar scenes like facades with clearly visible straight lines, vanishing points can be used, even if no depth information is available (Robertson and Cipolla, 2004; Baatz et al., 2011; Cao and McDonald, 2012). For more general scenes, it was shown that the sole usage of affine features can be improved, if they are normalized with respect to the local surface normal rather than to the affine shape (Köser and Koch, 2007). Still this approach relies on the affine detector and shares its drawbacks. For large planar scene parts, viewpoint invariant patches on the dominant planar structures can be extracted and rotated to a frontal view (Wu et al., 2008; Cao et al., 2011). This allows to obtain perspective invariance up to in-plane rotations in the image – or up to similarity transformations if absolute scale is unknown.

Our first contribution extends this approach to general developable surfaces, utilized for rectification and matching. For complex scenes the detection of the underlying parametric objects becomes the bottleneck. Further, the main problem still remains and matches can only be obtained on isolated objects and interesting texture has to lie on the detected geometry. Thus, our second contribution leverages salient directions to obtain a rectification for free form surfaces in urban environments.

5.2 Developable Surfaces

As our first approach builds on the notion of developable surfaces, we start by briefly introducing the underlying concept. In general a surface with *zero* Gaussian curvature at every surface point is developable (Kühnel, 2006) and can be flattened onto a plane without distortion (such as stretching or short-

ening).

To determine the Gaussian curvature of a surface, suppose we are given a smooth function s that maps 2D parameters u, v to points in 3D space, *i.e.*, $s : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ such that $s(u, v) = (x, y, z)^T$. The graph S of this function is a two-dimensional manifold and our surface of interest in 3D space. The derivatives $s_u = \partial s/\partial u$ and $s_v = \partial s/\partial v$ of s with respect to the parameters u and v define tangent vectors to the surface at each point. Their cross product yields the normal vector $n = s_u \times s_v$ to the surface. The second partial derivatives of s with respect to u, v are now used for constructing the shape operator

$$\mathrm{II} = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} n^\intercal s_{uu} & n^\intercal s_{uv} \\ n^\intercal s_{uv} & n^\intercal s_{vv} \end{bmatrix} \quad , \tag{5.1}$$

which is also called the second fundamental form of s. The principal curvatures κ_1, κ_2 of the surface at a given position are defined as the eigenvalues of II. They measure how the surface bends by different amounts in different directions at a particular point. Finally, the determinant $\det(\mathrm{II}) = \kappa_1\kappa_2$ denotes the Gaussian curvature; in case it vanishes everywhere on the surface (at least one of the eigenvalues is zero) the surface is developable. The intuition is that in direction of zero curvature the surface can be described as a line. Hence, the surface development is just an unrolling of all corresponding lines into one plane. We refer the interested reader to Kühnel (2006) for more details.

For example a cylinder is developable (see Fig. 5.2b for an illustration), meaning that at every point the curvature in one direction vanishes. Its mean curvature is not zero, though; hence it is different from a plane. Contrary, a sphere is not developable, since its surface has constant positive Gaussian curvature at every point. Other basic developable shapes are planes, cylinders, cones, or oloids and sphericons[1], and variants thereof such as cylindroids or oblique cones. Intuitively, they are flattened by rolling the object on a flat surface, where it will develop its entire surface. In fact, all surfaces which are composed of the aforementioned objects are developable as well. In practice, many objects in our environment are made by bending sheets or plates and thus form developable surfaces. Fig. 5.1 illustrates several real-world developable

[1] An oloid is defined as the convex hull of two congruent disks lying in perpendicular planes, so that the distance between their centers is equal to their radius. A sphericon is similar to an oloid, but is defined by the convex hull of two perpendicular semicircles.

surfaces; note that even such complex structures as the church roof top (Fig. 5.1 very right) are (piece-wise) developable.

5.3 Viewpoint Invariance via Developable Surfaces

In the following we present our approach of matching two views of a rigid scene, separated by a wide-baseline, by means of developable surfaces. However, we point out that the same techniques are applicable for identifying and recognizing a single object in a database, for loop detection or for automatically registering multiple overlapping textured 3D models. As input to our algorithm we assume two RGB-D images with sufficient overlap. Given pixel-wise depth measurements $d_{u,v}$ and camera intrinsics K a 2.5 dimensional point cloud is obtained per view via

$$(x, y, z)^{\mathsf{T}}_{u,v} = \mathrm{K}^{-1}(u, v, 1)^{\mathsf{T}} d_{u,v} \quad \forall u, v \in \Omega, \tag{5.2}$$

with image coordinates u, v in the image domain Ω. Then our method progresses in four steps, which are:

(i) Detection and parameter estimation of certain developable surfaces in the depth data.

(ii) Generation of flat object textures by means of developing the detected surfaces.

(iii) Detection/Description of features in the unrolled images (*i.e.*, in the surface) and matching against the other views.

(iv) Geometric verification of found correspondences.

We will explain them in more detail in following Sections 5.3.1 to 5.3.3.

5.3.1 Multi-Model Estimation

As described in the previous Sec. 5.2, many different developable surfaces exist. We focus on three basic shapes, the plane, the cylinder and the cone, because these shapes possess a low parametric representation and thus are detected reliably in depth data. Identifying these surfaces falls into the category of multi-model estimation and several techniques have been suggested to cope with it, including randomized hough transform (Ballard, 1981; Xu et al., 1990;

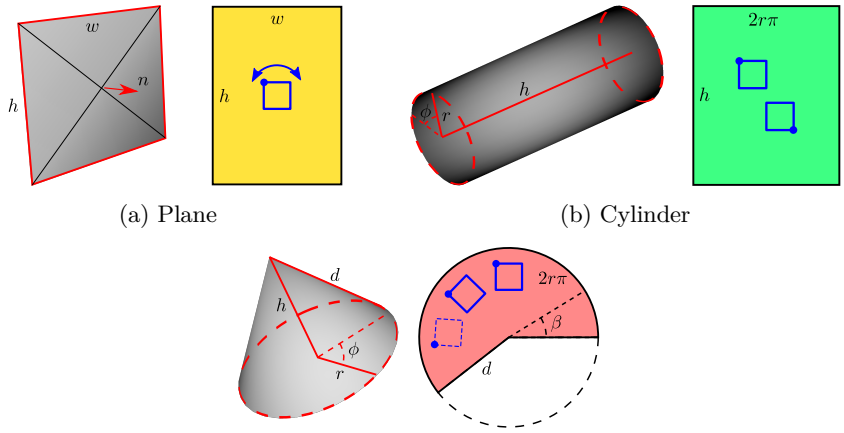
Kälviäinen et al., 1995), sequential RANSAC or more recently J-Linkage (Toldo and Fusiello, 2008), multi-structure segmentation (Wang et al., 2012), energy based multi model fitting (Isack and Boykov, 2012) or more problem-specific machine-learning inspired geometric classification approaches in the spirit of Rusu et al. (2009b).

We employ RANSAC to obtain model parameters in the captured depth data. Since surfaces are mostly local and continuous we utilize a local sampling strategy, where consecutive samples are drawn within a 0.5m radius. Surface models are searched for in order of increasing complexity, *i.e.*, initially planes are detected, followed by cylinders and cones. We limit the size of models to physically plausible extents for the expected outdoor or indoor environments. In addition, found models need to guarantee that they show sufficient support over their surface to avoid algorithmic plausible, but incorrect estimations. Consequently, we reject models whose support is only defined at isolated points or clusters. Once detected, we robustly determine the model size in the image and estimate the spatial extent (*e.g.*, height of cylinder). Subsequently, initial model parameters are updated via a non-linear optimization on the evaluated inlier set. After each iteration 3D points supporting the estimated model are removed from the search space, which prohibits assignment to multiple models. This iterative procedure terminates as soon as no model with sufficiently large support is found any more. While the presented sequential approach is rather simple, it also is quite effective and can run on down-sampled input data to obtain processing at interactive frame rates without degrading accuracy.

5.3.2 Developing Surfaces

After the initial model estimation and parameterization, a flat texture is generated per detected model. We describe obtained mappings for principal geometric shapes in the following. An illustration of the considered shapes and their developed surfaces is given in Fig. 5.2.

Planes A plane is parameterized as the tuple $\pi_S = (\mathbf{n}, d)$ with normal vector $\mathbf{n}$ and distance to the origin d. In addition, the model estimation provides a bounding box (or mask) for the region of interest on the 3D plane. Two orthogonal vectors in the plane are chosen to form a basis $\mathcal{B}_S$ and we sample the plane in equidistant steps, *i.e.*, we define a grid in the plane. The original image and surface plane π_S together with the origin $\mathbf{O}$ of $\mathcal{B}_S$ define a unique



(a) Plane (b) Cylinder

(c) Cone

Figure 5.2: Illustration of parametric 3d models and their developed 2D surfaces The (blue) oriented square denotes that there remains a rotation variance for the plane texture, which is reduced to an 180° degree ambiguity for the developed cylinder surface and completely resolved for the cone.

linear mapping. It is used to project each of the grid vertices into the original image to obtain the appropriate color. The resolution is chosen such that we do not lose any image details; *i.e.*, we project the four bounding box corners into the original image and evaluate the Jacobian matrix of the texture warp for some arbitrary grid resolution. Afterwards we alter the scale such that the smallest minification between the developed surface and the original image is 1 (for details on texture mapping see Heckbert (1989)).

The result is equivalent to (Wu et al., 2008), where the virtual frontal rendering coincides with the developed plane and a homography describes the mapping. In general a plane-induced homography is given by the relation

(Hartley and Zisserman, 2004, *p*326f)

$$\begin{aligned}\tilde{\mathbf{x}}' &= \mathrm{K}'\mathrm{P}' \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tilde{\mathbf{X}} \quad \text{s.t.} \quad \tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{r} \\ -\mathbf{n}^\mathsf{T}\mathbf{r}/d \end{pmatrix} \\ &= \mathrm{K}'\mathrm{P}' \begin{bmatrix} \mathrm{R} - \mathbf{t}\mathbf{n}^\mathsf{T}/d \\ -\mathbf{n}^\mathsf{T}/d \end{bmatrix} \mathrm{K}^{-1}\tilde{\mathbf{x}} \ . \end{aligned} \tag{5.3}$$

This is because for a projective (source) camera any point on the ray $\mathbf{r} = \mathrm{K}^{-1}\tilde{\mathbf{x}}$ will project to $\mathbf{x}$, *i.e.*, $\tilde{\mathbf{X}} = (\mathbf{r}, \lambda)^\mathsf{T}$ where λ parameterizes the point on the ray. Since, the 3D point lies on the plane it satisfies $(\mathbf{n}^\mathsf{T}, d)(\mathbf{r}, \lambda)^\mathsf{T} = 0$, which determines the point depth $\lambda = -\mathbf{n}^\mathsf{T}\mathbf{r}/d$. To obtain a frontal virtual camera, R is given by the rotation that aligns the optical axis and plane normal, *i.e.*,

$$\mathrm{R}(\theta, \mathbf{a}): \quad \theta = \arccos(-\mathbf{n}^\mathsf{T}\mathbf{e}_z) \quad \text{and} \quad \mathbf{a} = \mathbf{e}_x \times \mathbf{n} \ . \tag{5.4}$$

The new camera center $\mathbf{C} = -\mathrm{R}\mathbf{t}$ is positioned such the new optical axis intersects with $\mathbf{O}$. Previous surface sampling is equivalent to using a scaled affine destination camera (cf. Eq. (2.5) in Sec 2.2); therefore, with $\mathrm{P}' = \mathrm{P}_\infty$ and $\mathrm{K}' = \mathrm{diag}(s, s, 1)$ we obtain

$$\tilde{\mathbf{x}}' = \underbrace{\left[\mathrm{diag}(1,1,0)\,\mathrm{R} - \begin{pmatrix} t_1 \\ t_2 \\ 1/s \end{pmatrix} \mathbf{n}^\mathsf{T}/d \right] \mathrm{K}^{-1}}_{\mathrm{H}} \ \tilde{\mathbf{x}} \ . \tag{5.5}$$

From this relation we can observe that $\mathbf{O} \propto (t_1, t_2)^\mathsf{T}$ and that the scale s models the distance of the new virtual camera from the surface plane.

Cylinders A cylinder is given by the tripled $(\mathbf{c}, \mathbf{a}, r)$ with cylinder base center $\mathbf{c}$, axis vector $\mathbf{a}$ of length h and radius r. In order to unroll it, 3D points on the surface are expressed in their cylindric coordinates (r, ϕ, z) (with $\mathbf{c}$ as the origin). This results in the 2D surface parameterization (ϕ, z) and a unique mapping into image plane coordinates. The angular resolution in ϕ is determined to match the resolution along the cylinder axis and to obtain an image of aspect ratio $h \times 2\pi r$ for a full 3D development of the cylinder (see Fig. 5.2b). In case scale is known and when it is desirable not to normalize over scale (*e.g.*, because of similar features at different scales) we choose a metric surface reso-

lution. Otherwise, we evaluate the local magnification/minification between the original image and the surface texture and ensure that no resolution is lost during unrolling.

Cones A cone is parameterized and developed very similarly to the cylinder, taking into account that the surface tappers smoothly towards the apex. To obtain a flat surface texture, it is positioned with a line from the apex to the base circle of length $d = \sqrt{r^2 + h^2}$ (see Fig. 5.2c) in the plane for development. Afterwards the apex is fixed and the cone rolled around it, resulting in a circle segment. Thus 2D texture coordinates (β_i, d_j) are directly related to points on the cone surface. Similar to the cylinder, we backward map texture coordinates across the 3D surface into the original image to obtain the colors for the surface texture, maximizing its resolution.

All mappings are very efficiently implemented on the GPU or by using standard backward mapping on the CPU.

5.3.3 Feature Detection and Correspondence Verification

Feature detection is performed directly in the unrolled textures. This is conceptually different to Mikolajczyk et al. (2005); Köser and Koch (2007) which first detect features and then try to normalize these wrt. to viewpoint variations. It is related to Wu et al. (2008); Robertson and Cipolla (2004); Cao and McDonald (2012), however these approaches only consider planes for normalization. The unrolled textures allow to reach perspective invariance with only normalizing in-plane rotation in the image (or similarity normalization in case absolute scale is unknown). Even better, since cylinder[2] and cone define an inherent reference direction with their axes, all features can be expressed with respect to this orientation. Consequently, it allows the fast extraction of basic features and to distinguish local regions that differ only by scale, orientation or linear shape. All detected features on the different developed textures are combined to form the set of features for a RGB-D image and are subsequently used for wide-baseline matching.

Naturally the set of estimated matches contains numerous outliers, which do not satisfy the underlying camera pose change. Therefore, correspondences are

[2]For the cylinder there still exists a 180° ambiguity for the direction of the axis.

Scene type	Synthetic setup	Floor (Kinect)	Table (Kinect)	Trees (stereo)	Pylon (stereo)
RGB-D images	37	9	27	3	56
Developed surfaces	255	79	195	22	227
Enhancement ratio	6.89	8.78	7.22	7.33	4.05

Table 5.1: Quantitative comparison of SIFT descriptor matches between original RGB-D images and developed surfaces for different scenes.

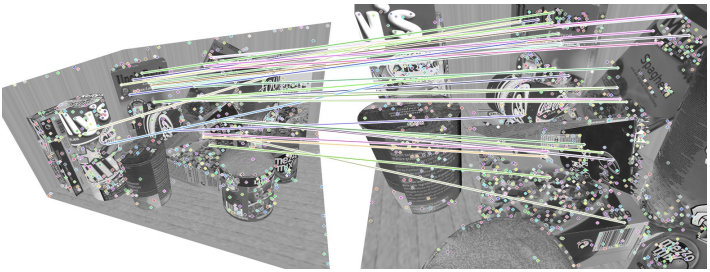
checked using geometric verification (*e.g.*, using RANSAC). Each feature does not only include information about the 3D position, but also the local surface normal. Additionally, when orientation is known from the cylinder or cone geometry or estimated locally from gradients (Lowe, 2004), three characteristic directions are known at each 3D feature point. This allows for a stratified verification as proposed by Wu et al. (2008) or for a minimal solution in RANSAC that requires only a single correspondence.

5.4 Results for Active and Passive Stereo Devices

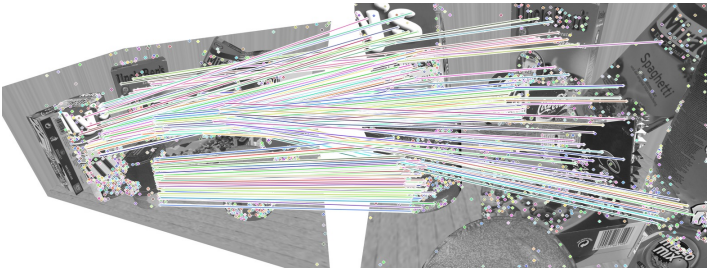
In this section we demonstrate our technique for different scenes and cameras. As mentioned previously, image feature estimation in the developed surfaces can be accomplished with a basic detector such as a Harris corner detector. However, since we aim to compare obtained matches from developed surfaces with matches in the original RGB-D data, we chose standard SIFT as our detector and descriptor to guarantee comparability. Note, that employing upright-SIFT would also treat our approach with favor due to its greater discriminative power. Detected image features are matched against each other in their descriptor space. To eliminate ambiguous matches (*e.g.*, between repetitive structures) all best matches are kept for which the distance ratio to the second best match falls below 0.6 (known as the ratio test in Lowe (2004)). Then, each feature is additionally augmented by its position in 3D space, which is determined by the corresponding 3D surface model. For features in the original RGB-D images we consider present pixel-wise depth measurements and neglect them in case no depth is available, *e.g.*, due to occlusions.

Fig. 5.3, Fig. 5.4 and Fig. 5.5 illustrate obtained results for a synthetic setup,

(a) Detected models in the points cloud and their developed surface textures.

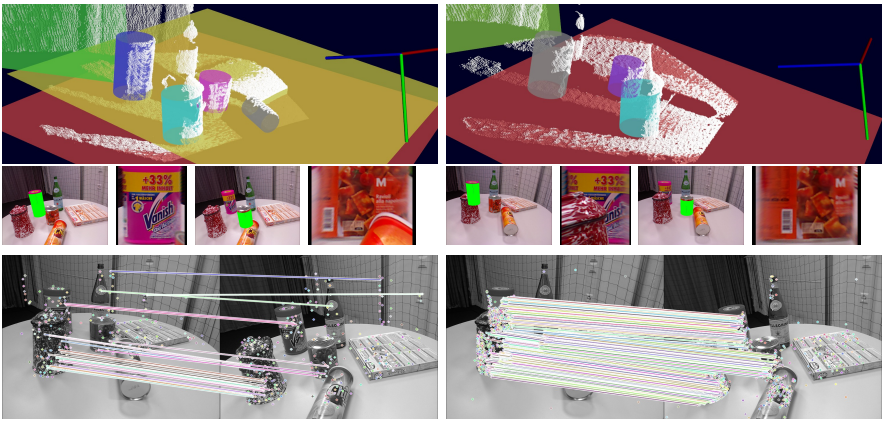


(b) Matching between initial RGB-D images

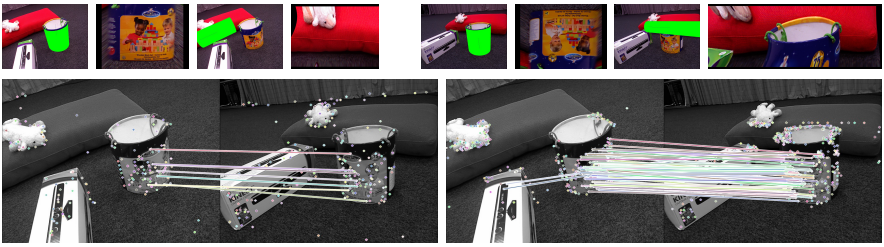


(c) Matching between developed surfaces, *i.e.*, the textures from (a).

Figure 5.3: Wide baseline matching for a synthetic setup. Illustrated matches are consistent wrt. to the underlying 6DoF transformation.

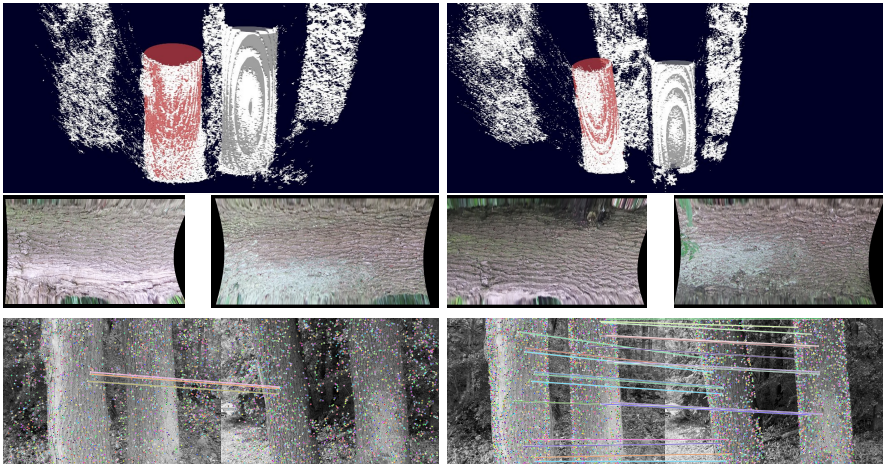


(a) Table

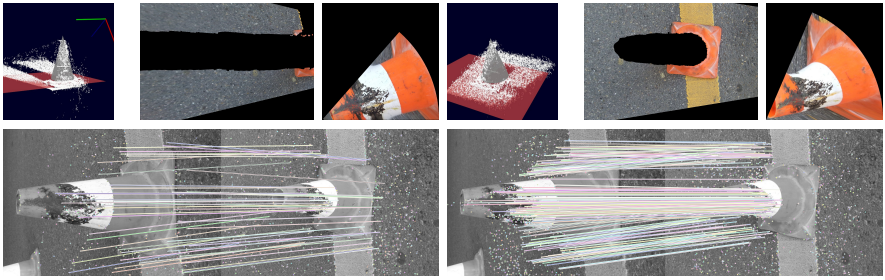


(b) Floor

Figure 5.4: Wide baseline matching for RGB-D data captured with a Kinect sensor. (top rows) Detected objects (green) and their respective developed surface for the two views. (bottom rows) Consistent SIFT matches between original images (left) and developed surfaces (right), respectively.



(a) Trees



(b) Pylon

Figure 5.5: Wide baseline matching between images taken by a Fuji3D consumer stereo camera . (top rows) Detected models in the 2.5 point cloud and their respective developed surfaces. (bottom rows) SIFT feature matches between original scenes (left) and developed surfaces (right), respectively. Note, that for (b) depth estimates are noisy and contain a considerable amount of errors, leading to degraded parameter estimation for the detected cone.

indoors scenes captured with a Kinect camera, and outdoor scenes taken with a Fuji3D stereo camera, respectively. Comparing feature detection and matching in the original images and in the images of developed surfaces (see Tab. 5.1) we can record the following: While approximately the same number of features are detected and an equal amount of potential matches is obtained, evaluation shows that for the latter the amount of finally remaining *correct* matches is significantly larger. Between the original RGB-D images many potential matches are wrong due to viewpoint distortions in the descriptor space and thus need to be rejected. This validates that our approach of viewpoint invariant description of developable surfaces is able to extract features, which are stable over a variety of largely different viewpoints and improves wide-baseline matching considerable. In addition, rather than interpreting our approach as a competitor to standard feature matching, one should see it as an additional cue for obtaining more stable features.

5.5 Viewpoint Invariance via Salient Directions

Normalization by means of developable surfaces has the limitation that particular geometric shapes need to present and detected in the scene. Consequently, it is desirable to drop this requirement and achieve viewpoint invariance also for free-form surfaces. We thus propose to utilize the principle of salient directions present in the geometry and suggest to extract (several) directions from the distribution of surface normals or other cues such as observable symmetries. Rendering the whole scene from these repeatable directions using an orthographic camera generates textures which are identical up to 2D similarity transformations.

First, let us define what we mean by a salient direction. The pose of a RGB-D camera in the world coordinate system is specified by the mapping of a point $\mathbf{X}$ from world to camera coordinates via

$$\mathbf{X}_i = s_i \mathrm{R}_i \mathbf{X} + \mathbf{t}_i = s_i \mathrm{R}_i \mathbf{X} - s_i \mathrm{R}_i \mathbf{C}_i \ . \tag{5.6}$$

Here, $\mathbf{C}_i$ represents the origin of the scanning device in world coordinates, while R_i represents its orientation and s_i is the scaling. In the following we will use the index i to refer to any single scan and indices i, j to distinguish between any two scans.

(a) Scanner images

(b) SDR images of roof with rotation ambiguity

(c) SDR images of altar with fixed rotation

Figure 5.6: (a) Images taken from two different positions, which naturally exhibit a wide baseline. The red altar visualizes correspondence. Feature matching and thus registration from these images fails in most cases. (b-c) Generated salient direction rectified (SDR) renderings along corresponding salient directions. Images of the roof are equivalent up to a 2D euclidean transformation (cf. Claim 2), while images in (c) correspond up to a translation (cf. Claim 3).

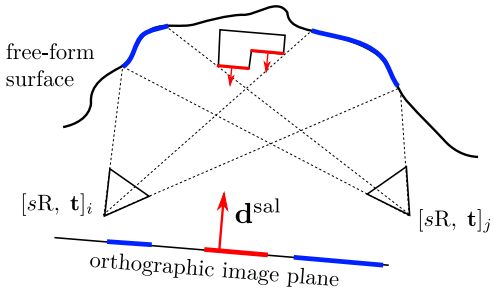


Figure 5.7: Orthographic renderings along a salient direction. The scene overlap of planar (red) and free form (blue) surface will be rendered identically along $\mathbf{d}^{\mathrm{sal}}$ for each scanner.

Definition 1. *A salient direction is a real-world direction in global coordinates* $\mathbf{d}^{\mathrm{sal}}$ *that can be observed locally as* $\mathbf{d}_i^{\mathrm{sal}}, \mathbf{d}_j^{\mathrm{sal}}$ *in independent scans* i *and* j*:*

$$\mathbf{d}^{\mathrm{sal}} = \mathrm{R}_i^{\mathrm{T}} \mathbf{d}_i^{\mathrm{sal}} = \mathrm{R}_j^{\mathrm{T}} \mathbf{d}_j^{\mathrm{sal}}. \tag{5.7}$$

Intuitively, imagine $\mathbf{d}^{\mathrm{sal}}$ is the north direction, that is represented in scans i and j as $\mathbf{d}_i^{\mathrm{sal}}$ and $\mathbf{d}_j^{\mathrm{sal}}$ respectively.

As input to our algorithm we consider 2.5D depth and image data, either from a laser scanner or from a consumer depth device or stereo system. In case of panoramic data we assume that both image and depth data are given as faces of a cube-map. Then, for the depth data, local normals are estimated and we will call the set of range data, color data and normals taken from one position a *scan*. The goal is now to render a view which is suitable for matching it against other scans. Ideally we want to produce a normalized image that looks the same as a normalized image from another location (see Fig. 5.6 for examples and Fig. 5.7 for an illustration).

Definition 2. *A salient direction rectified (SDR) image, is an image which is obtained by rendering the scene along a salient direction* $\mathbf{d}_i^{\mathrm{sal}}$ *with orthographic*

projection matrix

$$\mathrm{P}_i = \begin{bmatrix} \tilde{\mathbf{r}}_{i,1}^T \\ \tilde{\mathbf{r}}_{i,2}^T \end{bmatrix} \quad \text{with} \quad \mathrm{P}_i\, \mathbf{d}_i^{\mathrm{sal}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ , \tag{5.8}$$

where $\left\{\tilde{\mathbf{r}}_{i,1}, \tilde{\mathbf{r}}_{i,2}, \mathbf{d}_i^{\mathrm{sal}}\right\}$ *forms an orthonormal basis of* $\mathbb{R}^3$ *and relates to the orthographic camera coordinate system.*

Claim 1. *Given a salient direction* $\mathbf{d}^{\mathrm{sal}}$ *with corresponding local directions* $\mathbf{d}_i^{\mathrm{sal}}, \mathbf{d}_j^{\mathrm{sal}}$ *in scans* i *and* j*, then corresponding points in the two SDR-images relate to each other via a 2D similarity transformation.*

As simple proof we want to show that with the given projection matrices $\mathrm{P}_i^{\mathrm{sal}}, \mathrm{P}_j^{\mathrm{sal}}$ image points $\mathbf{x}_i, \mathbf{x}_j$ relate to each other via

$$\mathbf{x}_j = s'\mathrm{R}'\mathbf{x}_i + \mathbf{t}' \ , \tag{5.9}$$

where s', R' and $\mathbf{t}'$ denote 2D scaling, rotation and translation, respectively. Without loss of generality we set the i^{th} scanner pose to $[\mathrm{I} \quad \mathbf{0}]$ and denote $[s_j\mathrm{R}_j \quad \mathbf{t}_j] = [s\mathrm{R} \quad \mathbf{t}]$. Then according to Eq. (5.8) for a 3D point $\mathbf{X}$ its projections in the two SDR-images are $\mathbf{x}_i = \mathrm{P}_i\mathbf{X}$ and $\mathbf{x}_j = \mathrm{P}_j(s\mathrm{R}\mathbf{X} + \mathbf{t})$. Also according to Eq. (5.8) P_i and $\mathrm{P}_j R$ span the same basis. Thus, comparison with Eq. (5.9) reveals

$$\mathbf{t}' = \mathrm{P}_j\mathbf{t} \quad \text{and} \quad \mathrm{R}' = \mathrm{P}_j\mathrm{R}\mathrm{P}_i^{\mathrm{T}} \quad \text{and} \quad s' = s. \tag{5.10}$$

Since Eq. (5.10) holds for every point $\mathbf{X}$ the solution is unique. Further it is easily verified that $\mathrm{R}'^{\mathrm{T}}\mathrm{R}' = \mathrm{I}$ and thus R' is orthogonal. As a result images must be related by a similarity transform, which proves the claim. SIFT features are well suited for handling this remaining ambiguity.

Claim 2. *If absolute scale is known – as for depth sensors and laser scans – the freedom reduces to a 2D euclidean transformation.*

The proof is trivial, since for constant scale across scenes $s = 1$. As a result this allows for scale variant feature description and matching. Observe that there is still one degree of freedom in choosing P_i, *i.e.*, there is an undetermined in-plane rotation.

Claim 3. *Given that a global direction* $\mathbf{g}$ *is known commonly among scans in local coordinates as* $\mathbf{g}_i$ *and that* $\tilde{\mathbf{r}}_{i,1}$ *is chosen as* $\tilde{\mathbf{r}}_{i,1} = (\mathbf{g}_i \times \mathbf{d}_i^{\text{sal}})/|\mathbf{g}_i \times \mathbf{d}_i^{\text{sal}}|$*, then generated images differ only in translation.*

Defining $\tilde{\mathbf{r}}_{i,1}$ as above and setting $\tilde{\mathbf{r}}_{i,2}$ orthogonal to it via $\tilde{\mathbf{r}}_{i,2} = (\mathbf{d}_i^{\text{sal}} \times \tilde{\mathbf{r}}_{i,1})/|\mathbf{d}_i^{\text{sal}} \times \tilde{\mathbf{r}}_{i,1}|$ ensures that $\mathbf{g}$ appears upright in the SDR-images. In this case $\mathrm{R}' = \mathrm{I}$ which leaves only $\mathbf{t}'$ and proves the claim. Only in case $\mathbf{g}$ coincides with $\mathbf{d}^{\text{sal}}$, $\tilde{\mathbf{r}}_{i,1}$ is undefined (a case which is easily spotted) and in-plane rotation is still ambiguous. In all other cases simple upright feature descriptors can be employed, which have been shown to be more discriminative than features with locally-adaptive orientation (*e.g.*, Baatz et al., 2011).

Our approach is separated into four stages, which we will explain in more detail in the next Sec. 5.6 and Sec. 5.7:

(i) Detection of salient directions (per scan).

(ii) Normalization of image data with respect to salient directions (per direction per scan).

(iii) Extraction of features (per SDR-image) and establishment of tentative correspondences.

(iv) Geometric verification and concurrent pose estimation (for a scan pair).

5.6 Salient Direction Detection and Image Normalization

Given a salient world direction that can be identified in two different scans, we have shown that we can transform the image content in a way that it becomes virtually invariant with respect to the unknown pose. Depending on the scene type several possibilities exist how to identify salient directions, including vanishing points (Baatz et al., 2011) in modern architecture, directions of repetitions or symmetries (Köser et al., 2011) in historical buildings or north direction from the sky or the time and the sun (Lalonde et al., 2010) in outdoor scenes. However, in this contribution we demonstrate the idea using salient directions derived from characteristics of geometric structures, that is peaks in the distribution of surface normals (cf. Fig. 5.8 and 5.9), as for example also utilized in (Novak and Schindler, 2013). For successful registration only a single peak needs to be consistent, while remaining modes can be different.

Dominant normal directions Potentially disjoint, locally planar surfaces give rise to dominant surface normals. Detection of those is rephrased as finding peaks within the sampled point-normal distribution in each scan. Mean shift (Comaniciu and Meer, 2002) is a suited approach to achieve this goal. It allows to model the density without explicitly parameterizing it, by evaluating a kernel K for normal n via

$$\hat{f}(\mathbf{n}) = \frac{1}{|\mathcal{N}(\mathbf{n})|} \sum_{\mathbf{n}_i \in \mathcal{N}(\mathbf{n})} K(\mathbf{n}, \mathbf{n}_i) \ , \tag{5.11}$$

with $\mathcal{N}(\mathbf{n})$ being the set of neighbors of $\mathbf{n}$. We initialize mean shift with 50 samples obtained as cluster centers from K-means. The algorithm now performs gradient descent on the density estimate $\hat{f}(\mathbf{n}_k)$ and sample trajectories reach stable points at peaks of the density function.

As a distance measure between normals we use their orientation agreement. In particular we utilize the cosine distance $1 - \mathbf{n}^\mathrm{T}\mathbf{n}_i$ which in general relates to density estimation on a hypersphere. Furthermore, we employ a symmetric kernel with a smooth Parzen estimate (*i.e.*, decaying weight on normals at larger distance) with an additional cut off at a maximum of $\varphi = 10°$. Thus

$$K(\mathbf{n}, \mathbf{n}_i) = \begin{cases} c_h \cdot \exp\left(-\frac{1}{h}(1 - \mathbf{n}^\mathrm{T}\mathbf{n}_i)\right), & \mathbf{n}^\mathrm{T}\mathbf{n}_i > \cos(\varphi) \\ 0, & \text{otherwise} \end{cases} \ , \tag{5.12}$$

where h is specifying the strength of the exponential weighting and c_h is a normalization constant [3].

The sampling density of points on a surface highly depends on the distance of the surface from the scanner, as well as the slant of the surface wrt. the scanning direction. Thus, if we used raw 3D points $\mathbf{X}$ (and their normals $\mathbf{n}$) as generated from the scanner much higher emphasis would be given to surfaces close to the scanner and frontal to the scanning direction. In particular SDR-images would be rendered from salient directions highly supported by structures near the scanner, and repeatability of salient directions between scans would be degraded. Thus, before mode finding we re-sample the point

[3] For two points on the unit sphere squared euclidean and cosine distance are equivalent: $\frac{1}{2}\|\mathbf{a} - \mathbf{b}\|^2 = \frac{1}{2}\left(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2a^\mathrm{T}\mathbf{b}\right) = 1 - \mathbf{a}^\mathrm{T}\mathbf{b}$. Thus a also mean-shift with a symmetric Gaussian kernel with variance $\sigma^2\mathrm{I} = h\mathrm{I}$ fulfills our conditions exactly.

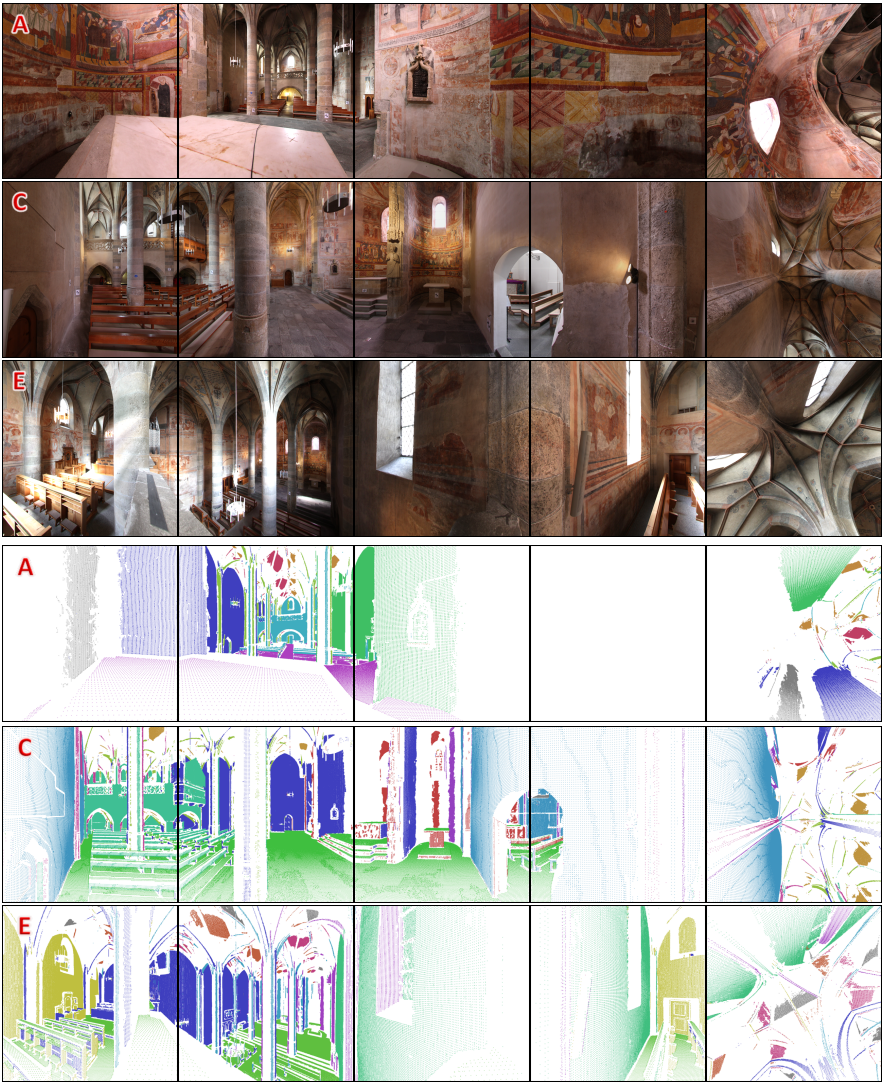


Figure 5.8: Input images (cube faces) for scan locations A,C,E for the CHURCH dataset and their computed support regions for salient directions (color coded with random colors). Points are sampled according to Eq. 5.13 and thus appear denser at further distance.

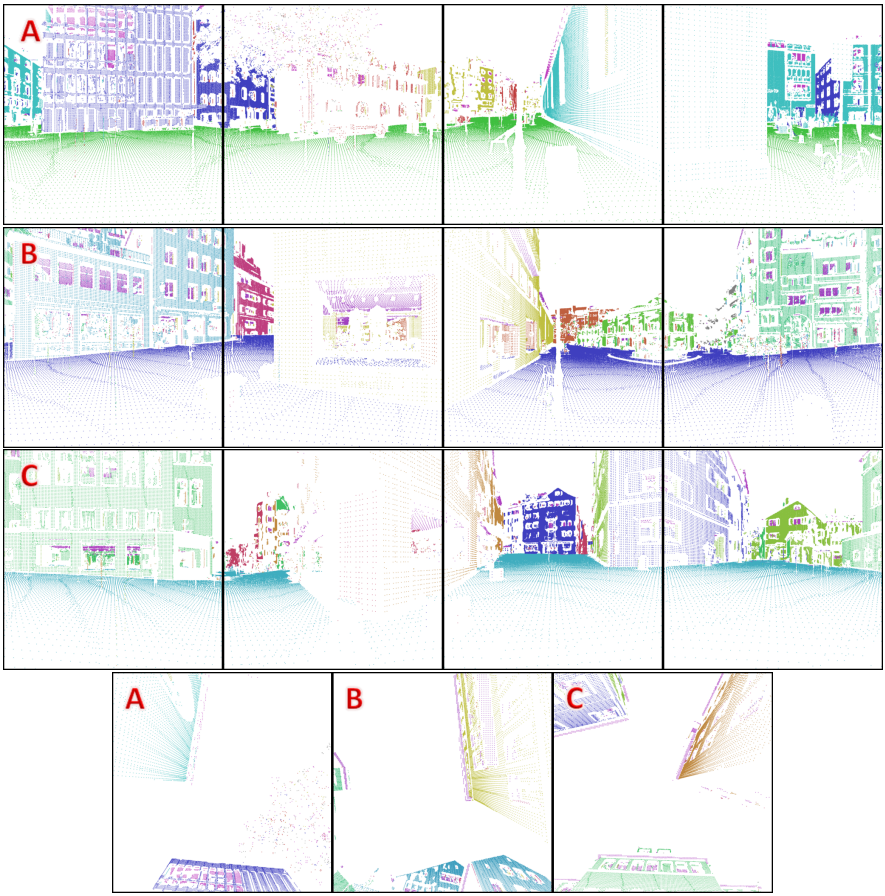
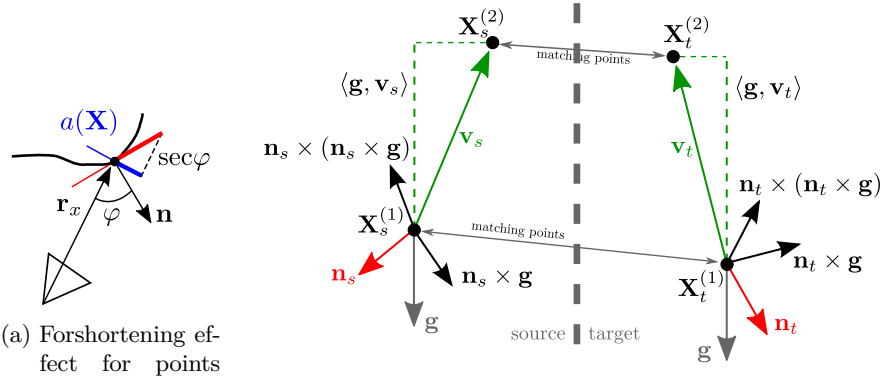


Figure 5.9: Support regions for salient direction in the CITY dataset. The bottom row shows the top faces. (Color codings between different scan locations do not indicate corresponding directions). See Fig 5.14 for the input images and a final registration.



(a) Forshortening effect for points sampling

(b) Relation for matching points in our 2pt RANSAC.

Figure 5.10: (a) To obtain a uniform sampling of 3D points each point is selected according to its likelihood $p(\mathbf{X})$ (see Eq. (5.13)) that depends on the angle between φ the point ray $\mathbf{r}_x$ and its normal $\mathbf{n}$; *i.e.*, the sampling likelihood increases with the slant of surfaces. (b) Each point defines a local coordinate system via its normal $\mathbf{n}$ and the gravity direction $\mathbf{g}$, which is sufficient for pose estimation. To be robust against we noise, we consider 2 point pairs and use their relative distances $\mathbf{v}_s$ and $\mathbf{v}_t$ for an early rejection criterion. See text for more details.

data. Conceptually the sampling likelihood $p(\mathbf{X})$ of a point $\mathbf{X}$ is proportional to the area it describes in the 3D scene, *i.e.*,

$$p(\mathbf{X}) \propto a(\mathbf{X}) \cdot \sec(\arccos\langle -\mathbf{r}_x, \mathbf{n} \rangle) \ . \tag{5.13}$$

See Fig 5.10a for a visualization. Here $a(\mathbf{X})$ denotes the surface area orthogonal to the scanning direction $\mathbf{r}_x$. For a depth map it is the projected pixel footprint at point depth X_z, while for a laser-scan it relates to the projected 2D scan interval (given by the angular scan resolution) at distance $\|\mathbf{X}\|_2$. As a result we generate a spatially evenly sampled point cloud and are able to determine salient directions bias-free.

View Synthesis When rendering 2.5D data from a different viewpoint, missing 3D information introduces holes in the generated images. Keypoints are

not detected in these visual artifacts, but descriptors might reach into or gap them. Since a descriptor captures gradient information, our desire is to avoid strong edges due to artifacts (which would perturb it) and we perform in-filing via a diffusion process. This keeps gradients small such that descriptors focus on the present texture information.

Since we don't require a fully consistent 3D mesh, we fill small holes directly in each SDR-image. Two different kinds of holes must be distinguished and in-painting is handled differently:

(a) Holes which are caused by occluders in the original scanner viewpoint placed in front of the surface to render, *e.g.*, a statue in front of a facade. In this case we not only fail to capture depth data (for parts of the facade in the given example) but also the corresponding texture (the texture of the statue will be captured, not the facade). As a matter of fact, in-painting is performed on the rendered SDR-images itself.

(b) Holes that are caused by missing data in the scanning process (*e.g.*, at reflective structures). Compared to the former, here texture information is available and thus we aim for a smooth inpainting on the orthographic depth-map. Then detected areas are re-rendered with the updated depth information to obtain the original texture.

Both cases can be easily distinguished by back-projecting hypothesized surface points into the original views and evaluating whether or not an occluder is present. Holes themselves are detected by searching for connected components in the initially rendered images, while for inpainting we utilize the fast marching method (Telea, 2004) as it is simple and fast.

5.7 Efficient Pose Estimation

Given several SDR-images of the scene, in each local image features are extracted. Since 3D models are given with absolute scale we can make use of Claim 2 and for each feature its absolute size is known. Thus, we could even apply features of fixed size, e.g. Harris corners (Harris and Stephens, 1988). However, to detect (different) structures at various levels of detail we perform feature detection in scale space using DoG (Lowe, 2004). During matching we still restrict the search for correspondences to those with same spatial extent. Similarly, in case the in-plane rotation of the orthographic camera is fixed (cf.

Claim 3) we employ upright descriptors. For each feature we find tentative correspondences via fast approximate nearest neighbor search.

For the relative registration of two scans we augment each feature by its 3D position and normal in the local coordinate system and denote points as $\mathbf{X}_s$ and $\mathbf{X}_t$ (in the following indices s and t indicate source and target scan, respectively). Then we seek the parameters of the relative transformation [R, $\mathbf{t}$] from source to target. For a laser scanner the gravity direction is usually known (assumed to be aligned with the z-axis in the following), so we need to estimate only 4 parameters; however, for a hand-held RGB-D sensor 6 DoF need to be estimated. In either way, pose estimation is performed within a random sample consensus scheme, *i.e.*, in each round the support for a generated transformation hypothesis [R, $\mathbf{t}$] is evaluated. Approaches presented in the following differ in the way they generate a transformation hypothesis in each iteration.

Relative Bearing and 3D Offset (4 DoF) As pointed out by Wu et al. (2008) each point defines a local coordinate system via its normal and feature orientation. For upright features the latter is fixed by the gravity direction and the local coordinate system is defined as $[\mathbf{n}, \mathbf{n} \times \mathbf{g}, (\mathbf{n} \times \mathbf{g}) \times \mathbf{n}]$). Thus a single feature correspondence suffices to estimate a transformation hypothesis. The rotation angle θ around the gravity direction $\mathbf{e}_z$ is computed between normals $\mathbf{n}_s, \mathbf{n}_t$ projected in the x-y plane

$$\begin{aligned} \bar{\mathbf{n}}_s &= \mathbf{n}_s - \langle \mathbf{n}_s, \mathbf{g} \rangle \mathbf{g} \quad \text{and} \quad \bar{\mathbf{n}}_t = \mathbf{n}_t - \langle \mathbf{n}_t, \mathbf{g} \rangle \mathbf{g} \\ \theta &= \arccos \langle \bar{\mathbf{n}}_s, \bar{\mathbf{n}}_t \rangle \cdot \operatorname{sign} \langle \mathbf{g}, (\bar{\mathbf{n}}_s \times \bar{\mathbf{n}}_t) \rangle \ , \end{aligned} \tag{5.14}$$

while the translation is then given by $\mathbf{t} = \mathbf{X}_t - \mathrm{R}_z(\theta)\, \mathbf{X}_s$. As an alternative to RANSAC a 1D voting scheme via kernel density estimation can be employed efficiently Wu et al. (2008).

We have found that normal vectors of extracted features tend to be noisy and are thus of limited value in their use for pose estimation. This is in particular the case for consumer depth cameras or stereo systems[4] and has two reasons. First, they are computed only in a local neighborhood and second, detected image features often correspond with structure boundaries introducing errors

[4] Normals in Wu et al. (2008) are taken from the estimated plane model, *i.e.*, the approach fits planes rather than individual feature points.

Algorithm 2 2-point geometric pose verification

Require: set $\mathbf{m} = [m_1, \dots m_n]$, $m_i = \{\mathbf{X}_s^{(i)}, \mathbf{X}_t^{(i)}\}$ of M potential matches between source and target scene
Require: number of iterations K and inlier threshold ε
for $k = 1, \dots K$ **do**
uniformly sample 2 matches m_i, m_j from $\mathbf{m}$
$\mathbf{v}_s \leftarrow \mathbf{X}_s^{(i)} - \mathbf{X}_s^{(j)}, \quad \mathbf{v}_t \leftarrow \mathbf{X}_t^{(i)} - \mathbf{X}_t^{(j)}$
if $|\|\mathbf{v}_s\| - \|\mathbf{v}_t\|| > \varepsilon$ **or** $|\langle \mathbf{v}_s, \mathbf{g} \rangle - \langle \mathbf{v}_t, \mathbf{g} \rangle| > \varepsilon$ **then**
reject sample pair and **continue**
$\bar{\mathbf{v}}_s \leftarrow \mathbf{v}_s - \langle \mathbf{v}_s, \mathbf{g} \rangle \mathbf{g}$ and $\bar{\mathbf{v}}_t \leftarrow \mathbf{v}_t - \langle \mathbf{v}_t, \mathbf{g} \rangle \mathbf{g}$
$\theta \leftarrow \arccos\langle \bar{\mathbf{v}}_s, \bar{\mathbf{v}}_t \rangle \cdot \operatorname{sign}\langle \mathbf{g}, (\bar{\mathbf{v}}_s \times \bar{\mathbf{v}}_t) \rangle$
$\mathbf{t} \leftarrow \frac{1}{2} \left(\mathbf{X}_t^{(i)} - \mathrm{R}_z(\theta)\, \mathbf{X}_s^{(i)} + \mathbf{X}_t^{(j)} - \mathrm{R}_z(\theta)\, \mathbf{X}_s^{(j)} \right)$
for all $l \in [1, M]$ **do**
if $\|\mathbf{X}_t^{(l)} - \mathrm{R}_z(\theta)\, \mathbf{X}_s^{(l)} + \mathbf{t}\| < \varepsilon$ **then**
insert m_l in $\mathbf{s}$
if $|\mathbf{s}| > |\mathbf{s}^*|$ **then**
$\mathbf{s}^* \leftarrow \mathbf{s}, \quad [\mathrm{R}_z^* \;\; \mathbf{t}^*] \leftarrow [\mathrm{R}_z(\theta) \;\; \mathbf{t}]$
return final transformation $[\mathrm{R}_z^* \;\; \mathbf{t}^*]$ and best inlier set $\mathbf{s}^*$

in the normal computation. As an alternative to using normals for registration, we will exploit the fact that corresponding local coordinate system axes can also be computed from pairs of correspondences. The orientation of these vectors is more precisely compared to normals due to their much larger spatial extent.

This gives rise to our robust 2-point geometric relative pose verification, which is presented in Alg. 2 (typical values are $K = 1000$, $\varepsilon = 3cm$) and Fig 5.10b visualized the relations between used variables. It incorporates an early rejection of generated hypotheses, such that only a fraction (on average 25% in our experiments) of generated transformation hypotheses need to be evaluated wrt. all data. Related to our new algorithm is the idea of filtering wrong correspondences in Johnson and Hebert (1999); however there authors use a heuristic rather than constructing an efficient RANSAC framework.

A transformation hypothesis is formed from 2 potential matches i, j drawn

at random from the correspondence set. 3D points $\mathbf{X}_s^{(i)}$, $\mathbf{X}_s^{(j)}$ in the source and $\mathbf{X}_t^{(i)}$, $\mathbf{X}_t^{(j)}$ in the target scene form vectors $\mathbf{v}_s$ and $\mathbf{v}_t$ respectively, connecting the 2 points in the local scans. If the chosen samples are correct matches, then the length of these two vectors must be equal. In addition, because we are searching for a rotation around the z-axis, their height difference has to be equal as well. This leads to an *early rejection criterion* allowing to avoid computing and testing the underlying transformation hypothesis. Given the previous two conditions hold, we first compute a relative rotation $\mathrm{R}_z(\theta)$ from the two vectors similar to Eq. (5.14). Second we evaluate the translation $\mathbf{t}$ between target and rotated source points.

Full 6 DoF transformation To estimate all 6 DoF of a 3D rigid body transformation, at minimum 3 corresponding points are required (if normals and feature orientations should be avoided). Procrustes analysis (Eggert et al., 1997) returns the optimal rotation and translation by decomposing the 3×3 correlation matrix between points (cf. Sec 2.5.1). An early rejection of samples based on the vector length between point pairs can be employed in a similar way to our previously mentioned 2-point pose verification.

5.8 Experimental Evaluation

For evaluation we recorded 3 different datasets with different scene characteristics which are typical for laser scanning scenarios. CHURCH is an indoor dataset of an old church consisting of 5 scans and exhibiting many vaults. Besides peaks in the normal distribution, in this scenario we also extract symmetry planes. Note that there exists a sign ambiguity for the symmetry plane normal, thus we use both possible normal directions as salient direction. For CITY we captured 3 scans in an urban area showing a high number of structured facades (*e.g.*, balconies). Finally CASTLE combines a construction site and a historic building [5].

For all experiments the input data format and parameters of our algorithm were kept constant. Panoramic images and range data are represented as 6 faces of a cube-map, each of size $2k \times 2k$ and $1k \times 1k$ pixels, respectively. For salient direction estimation we subsample the depth data as explained, while

[5] The datasets and additional results are available at `http://www.cvg.ethz.ch/research/saldir-rgbd-registration`

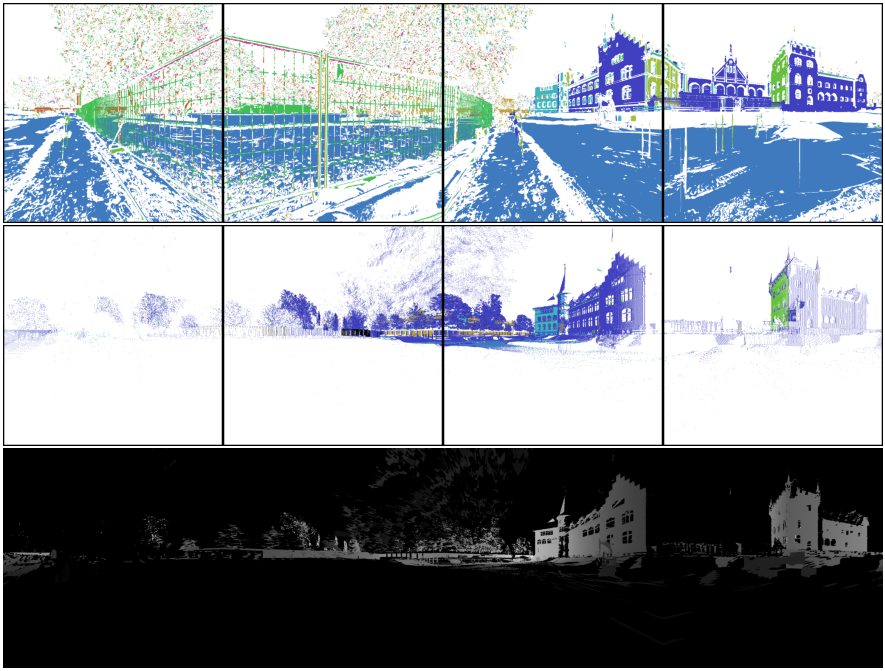


Figure 5.11: Color coded are the support regions of salient directions. (Top) Point cloud from scanner location 1; (middle) point cloud from position 2 projected into viewpoint of 1. (Bottom) Depth-map for the projection.

the kernel bandwidth and standard deviation (Eq. 5.12) are set to 10° and 5°, respectively.

Repeatability of Salient Directions It is essential for successful registration that we extract at least one salient direction (up to small variation) in both viewpoints. This task becomes more difficult with less overlap between regions. For evaluation we have taken scans with known relative pose and rendered the source scene into the viewpoint of the target scene (see Fig 5.11). There we compare the original depth values to those of the rendering. Areas with small difference in depth are considered as visible in both scenes, *i.e.*, they define

(a) SDR-image (b) Planar rectification

Figure 5.12: Comparison between viewpoint normalization via SDR-images and planar rectification (Wu et al., 2008; Cao et al., 2011). Clearly our approach can handle arbitrary surface shape and extract features on those.

the area of overlap between scans. Thus, in these regions corresponding salient directions (defined as directions differing by 10° at maximum) can and should get support. We now determine repeatability scores by comparing the number of corresponding salient directions to the total number of detected salient directions. The lower left parts in Table 5.2 list our evaluation of repeatability scores. One can observe that re-detection rates are high.

Registration performance To demonstrate the registration performance of our approach we compare it against state-of-the-art planar RGB-D rectification (Wu et al., 2008; Cao et al., 2011). We also tried to match SIFT features extracted from the original images (*i.e.*, cube face images), but registration fails in more than half of the cases. Tab. 5.2 lists the number of correct matches vs. tentative correspondences for both our approach and the baseline. A match is seen as correct if the corresponding points are within a threshold of 5cm for the outdoor datasets and 3cm for Church (since it has smaller scale). As can be seen, we generate more tentative and correct matches, which enables us to register scan-pairs in cases where the other approach fails. As expected this is the case for scenes with numerous non-planar surfaces, such as the roof and apse dome in Fig. 5.6. Here our approach is crucial for successful registration, as planar rectification requires textured planes, which are small or non-existent

	A	B	C	D	E
A		**418 / 541** 222 / 261	**301 / 439** 127 / 160	**21 / 68** —	**15 / 39** —
B	5 / 5		**242 / 322** 131 / 161	**19 / 54** —	**53 / 95** 58 / 75
C	4 / 5	4 / 5		**159 / 225** 89 / 103	**154 / 190** 29 / 32
D	1 / 1	2 / 3	5 / 7		—
E	1 / 2	1 / 2	2 / 2	0 / 0	—

(a) Castle

	A	B	C	D	E
A		**335 / 419** 166 / 206	**75 / 146** —	**82 / 144** 65 / 75	**24 / 63** 16 / 23
B	6 / 7		**405 / 480** —	**349 / 435** 114 / 142	**69 / 148** 44 / 60
C	6 / 7	7 / 9		**121 / 168** —	**63 / 118** —
D	7 / 7	8 / 8	6 / 8		**123 / 166** 77 / 79
E	5 / 6	5 / 8	5 / 7	6 / 8	

(b) Church

Table 5.2: Registration evaluation for Castle and Church. (Upper right parts) Relation between correct and tentative matches, for our approach (in bold) and planar rectification (Wu et al., 2008; Cao et al., 2011). The results indicate our superior performance. (Lower left parts) Repeatability scores for salient directions, i.e. the ration of found and present salient directions in the scan overlap.

(cf. Fig. 5.12). Note that besides exploiting features on free-form surfaces, we completely separate stable geometries and textures; *e.g.*, salient directions can be established from an untextured white wall, while the features for matching originate from some other textured free-form surface.

Global Registration In addition, Fig 5.13 and Fig 5.14 illustrate the global registration results for CHURCH and CITY, respectively. Previously pair-wise estimated relative poses form a graph connecting the scans with successful registration. An initial solution for the absolute pose of each scans is obtained by construction of a minimum spanning tree (MST) in the graph and concatenating the relative transformations accordingly. To improve this initial set of absolute poses one can examine pose-graph optimization or bundle-adjustment. We execute the former where the goal is to minimize the error e_{ij} over camera poses P_i between a measure for the previously estimated (z_{ij}) and expected ($\hat{z}_{ij}$) relative transformation via

$$\arg\min_{P_i} \sum_{(i,j),i\neq j} e_{ij}^{\mathsf{T}} \Omega_{ij} e_{ij} \quad \text{s.t.} \quad e_{ij} = z_{ij} - \hat{z}_{ij}(P_i, P_j) \ . \tag{5.15}$$

The information matrix Ω is obtained from a non-linear refinement of the pair-wise estimation and edges are weighted by their number of inliers. Since the focus of this work is on the initial pair-wise registration, we refer to Grisetti et al. (2010); Kummerle et al. (2011) for further details. However, we want to point out that our estimated relative poses are very precise, as we observe that the solution obtained via the MST approximation is very close to the solution obtained after global optimization.

5.9 Discussion

Our second approach is more general, since we do not rely on features on particular fitted models, but match the whole visible scene, this way significantly increasing the surface area where features can be extracted. This is an important aspect, if the visible overlap between RGB-D scans is small. In addition it generates images that consistently capture objects and features across different levels of depth. Such features at geometry boundaries and folds are among the most discriminative, as known *e.g.*, from stereo. Contrary,

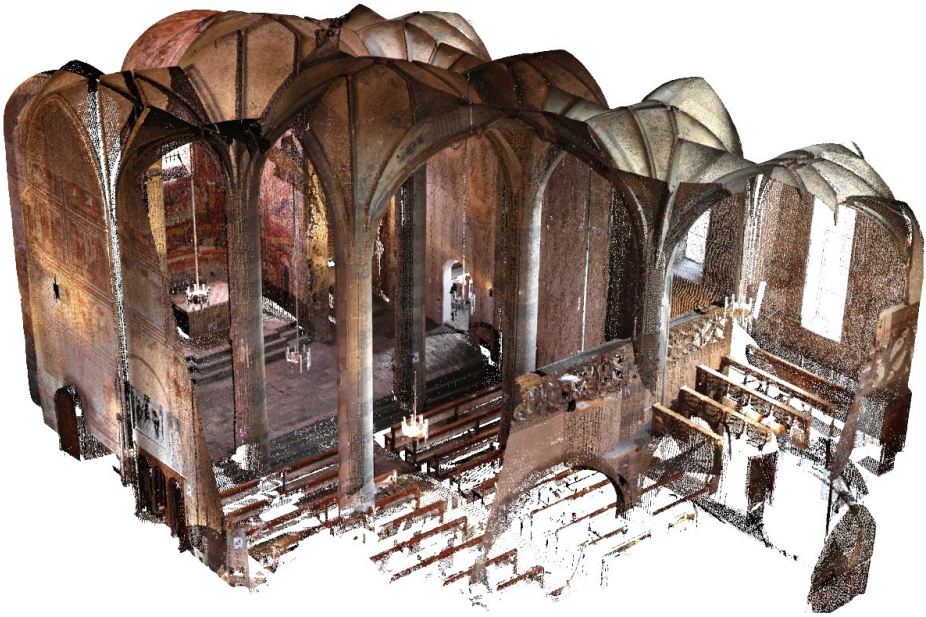


Figure 5.13: Cut through a 3D models obtained by our algorithm from 5 individual scans (CHURCH dataset). We achieve entirely automatic registration of arbitrary geometry from largely different viewpoints by exploiting depth and image data jointly.

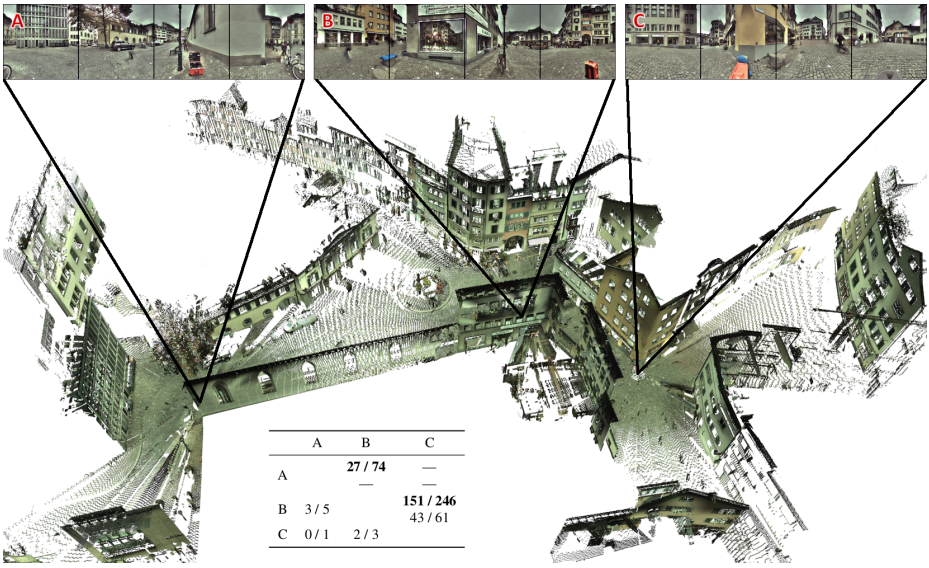


Figure 5.14: Result of our automatic scan registration for City. The scene was created from 3 viewpoints, visualized via their horizontal cube faces. The numbers in the table are organized in the same way as in Table 5.2.

Figure 5.15: Result of our automatic scan registration for CASTLE. The scene was created from 5 viewpoints, color coded in the bottom figure.

they are deliberately rejected in previous work and also in a developed surface representation, where depth discontinuities can not be handled. However, a salient direction based rectification requires fairly accurate depth information in order not to introduce artifacts in the viewpoint normalized images.

In summary, we can clearly see that the joint exploitation of image content and scene depth is superior over using either one of them separately. Our results demonstrate that accurate model registration is possible in cases where viewpoint variant features reach their limits. Inspired by these good results we were wondering if similar viewpoint invariance can be achieved for a single image without any measured depth information. A successful feature normalization would lead to stronger connections between images and thus be of great help in SfM – especially in indoor scenarios where relatively large viewpoint changes between image captures are enforced by the building architecture.

5.9.1 Rectification from a Single Image

So far rectification of features from a single image without any depth information had to rely on affine invariance derived from the local texture (Mikolajczyk et al., 2005). Recently, single-view reconstruction methods, estimating scene geometry directly by learning from data, have gained quite some popularity. In particular, coarse information about the 3D layout of a scene has shown to help boost the performance of applications such as object detection (Hoiem et al., 2008b), semantic reasoning (Ladický et al., 2014) or general scene understanding (Hoiem et al., 2008a).

Consequently, there arises the question if such estimates are also useful to gain viewpoint invariance for features. Admittedly, the resulting 3D reconstructions of such methods are of insufficient quality. The principal underlying idea behind these methods (Saxena et al., 2007, 2009; Liu et al., 2010) is, that particular structures have a certain real world size, and thus their size in an image gives rise to the scene depth. We argue that this is a rather weak hypothesis, since structures are likely to exist at different size in reality and perspective projection distorts them. As a consequence it renders the problem of single image depth estimation ill-posed in general. Though, perspective cues are not harmful, but actually helpful, because they carry information about the local surface orientation and allow to reason about the scene, *e.g.*, about the viewpoint of the camera. For example, coarse geometry was already estimated from vanishing points and lines (*e.g.*, Baatz et al., 2011; Cao and McDonald,

2012; Schwing and Urtasun, 2012) and leveraged for viewpoint normalization of extracted features (*e.g.*, Srajer et al., 2014). However, these approaches obviously fail for more general scenes, *e.g.* cluttered indoor environments. Thus, in Ladicky et al. (2014) we argue that it is beneficial to directly estimate first order derivatives of depth, *i.e.*, surface normals, since it provides more accurate results than estimation of absolute depth. We use a discriminative learning approach to estimate pixel-wise surface orientation solely from the image appearance. Our method combines contextual and segment-based cues and builds a regressor in a boosting framework by transforming the problem into the regression of coefficients of a local coding. The strength of our approach stems from the fact that we join both representations and intrinsically learn, when to use which. In addition in Zeisl et al. (2014) we show how to obtain more consistent estimates via the fusion (*e.g.*, of estimates from complimentary methods) and regularization of normals maps in a variational framework. Thereby, the unit norm constraint of surface normals renders the problem non-convex. We propose a local relaxation and an algorithm that is guaranteed to converge. As a result we obtain normal maps as illustrated in Fig 5.16. Since absolute depth is unknown, we will show in the following that local image rectification is accomplishable from surface orientation only.

Note, that the importance of normal estimation has been already recognized long before machine learning methods were available. Due to the lack of training data, proposed approaches (Horn and Brooks, 1986; Mallick et al., 2005; Ikehata and Aizawa, 2014) had to rely purely on the knowledge of underlying physics of light and shading. Resulting methods work only under strong assumptions about the knowledge of locations of light sources and properties of the material (such as the assumption of Lambertian surfaces). However, these approaches do not work in more complex scenarios such as indoor or outdoor scenes, and thus are not applicable for general problems.

Viewpoint Invariance from Normal Directions only Let us assume that for an identified keypoint $\mathbf{p}$ in the image we are given its surface normal $\mathbf{n}$. Then, we are looking for the homography which warps the image patch in the neighborhood of $\mathbf{p}$ to a frontal view, which is achieved via the plane-induced

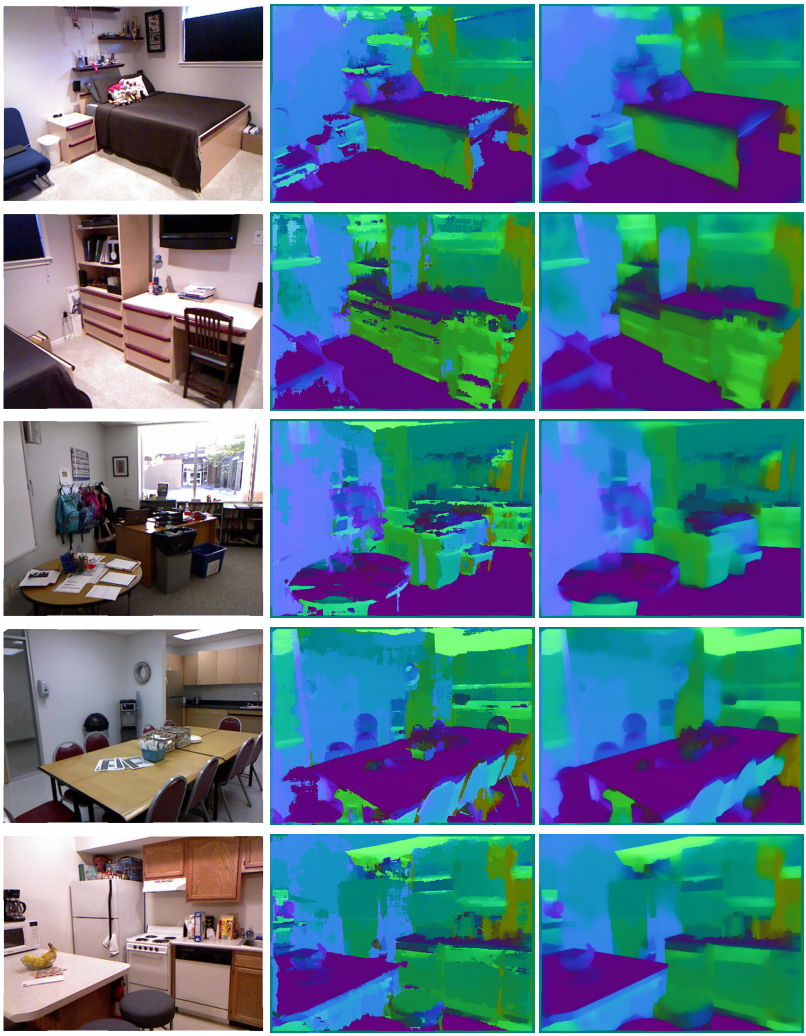


Figure 5.16: Normal estimates (Ladicky et al., 2014) and variational denoising (Zeisl et al., 2014) for images from the NYU2 dataset (Silberman et al., 2012).

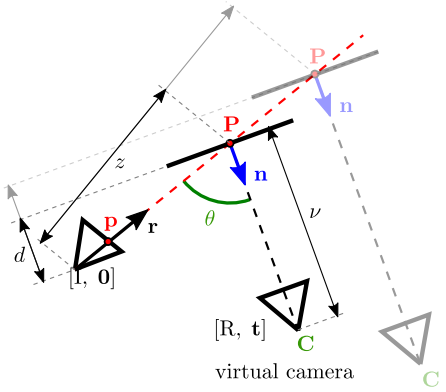


Figure 5.17: Geometric setup for the derivation of the point-normal induced homography presented in Eq. (5.19) and constructed for the keypoint location $\mathbf{p}$ and its normal estimate $\mathbf{n}$. Since the depth z of the point $\mathbf{P}$ is unknown (as is the location of the surface plane), there are infinite many frontal views. For our solution we chose $\nu = z$ and obtain a plane-induced homography independent of d.

homography (cf. Eq. (5.3))

$$\tilde{\mathbf{x}}' = \underbrace{\mathrm{K}'\mathrm{P}' \begin{bmatrix} \mathrm{R} + \mathrm{R}\mathbf{C}\mathbf{n}^\mathsf{T}/d \\ -\mathbf{n}^\mathsf{T}/d \end{bmatrix} \mathrm{K}^{-1}}_{\mathrm{H}} \tilde{\mathbf{x}} \ . \tag{5.16}$$

In the following we will (i) specify the pose in dependence of the particular keypoint $\mathbf{p}$ and its normal $\mathbf{n}$ and (ii) show how to get independent of the scene depth d. Fig. 5.17 illustrates the following derivations. First, the optical axis of the virtual camera will be $-\mathbf{n}$ such that it looks frontal on the plane π. Thus the rotation is again given by Eq. (5.4). Second, we want to translate the virtual camera, such that the 3D point $\mathbf{P}$ lies on its optical axis; *i.e.*, it should hold $\mathbf{C} = \mathbf{P} + \nu\mathbf{n}$, where ν specifies the distance between the 3D point and the virtual camera center. In original camera coordinates $\tilde{\mathbf{P}} = (\mathbf{r}, 1/z)^\mathsf{T}$ where the keypoint ray $\mathbf{r} = \mathrm{K}^{-1}\tilde{\mathbf{p}}$, and z denotes the unknown depth. Since $(\mathbf{n}^\mathsf{T}, d)\tilde{\mathbf{P}} = 0$ we obtain $z = -d/\mathbf{n}^\mathsf{T}\mathbf{r}$. Consequently the virtual camera center

is

$$\mathbf{C} = \mathbf{P} + \nu\mathbf{n} = -\frac{d\mathbf{r}}{\mathbf{n}^{\mathrm{T}}\mathbf{r}} + \nu\mathbf{n} \ . \tag{5.17}$$

For a projective virtual camera (*i.e.*, $\mathrm{P}' = [\mathrm{I},\ \mathbf{0}]$) Eq. (5.16) becomes

$$\mathrm{H} = \mathrm{K}' \left(\mathrm{R} - \mathrm{R} \left(\frac{\mathbf{r}}{\mathbf{n}^{\mathrm{T}}\mathbf{r}} - \frac{\nu}{d} \right) \mathbf{n}^{\mathrm{T}} \right) \mathrm{K}^{-1} \ . \tag{5.18}$$

Because the accurate 3D location of the plane is unknown, our goal is to become independent of d. For example, if we require that the point $\mathbf{P}$ has the same depth in both cameras, *i.e.*, $\nu = z$, we obtain a homography that is in dependence of the particular keypoint and its normal direction, but independent of its actual location:

$$\mathrm{H} = \mathrm{K}' \left(\mathrm{R} - \mathrm{R}(\mathbf{r} + \mathbf{n}) \frac{\mathbf{n}^{\mathrm{T}}}{\mathbf{n}^{\mathrm{T}}\mathbf{r}} \right) \mathrm{K}^{-1} \ . \tag{5.19}$$

This results can be interpreted as follows: Given that both cameras are at the same distance to a 3D point on a plane with known orientation (but unknown location) *for a particular point* $\mathbf{p}$, the actual depth of the plane is neglectable. This results only holds if the (virtual) destination camera is perspective. In case it is affine, the depth of the plane (*i.e.*, of the point $\mathbf{P}$) is important, since it determines the size of the object. Accordingly, the bottom row of H in Eq. (5.16) is non-zero and depends on d.

The important contribution here is, that conceptually it is well possible to obtain geometry normalized features from surface normal information solely, *i.e.*, no scene depth is required.

Rectification based on Normal Estimates Leveraging previous results, we tried to achieve viewpoint rectification of features. Thereby we tested two approaches. First, similar to Köser and Koch (2007) we use an affine detector and warp the local image region via the homography in Eq. (5.19) to obtain a frontal view and extract the descriptor therefrom. Second, segmenting the image in different regions, each supporting a (discrete) normal direction, we detect and extract features in the warped images (using a homography defined by the segment center). With both approaches we aim for more discriminative

matching for wide baseline image pairs. Improved correspondences would also lead to a larger number and more accurate point tracks in a SfM reconstruction, reducing the decomposition into submodels due to missing links.

However, our tests and evaluation showed that obtained results did not improve the performance and often degrade results. Closer investigation of the problem reveals, that estimated normals are not accurate enough *at the position of detected keypoints*. For our segmentation-based approach this means, that the assignment is often wrong for the segments containing keypoints. This is because normal estimation and keypoint detection turn out to be conflicting approaches. While surface normal estimates are fairly accurate for homogeneous areas (such as a white wall), they are erroneous for smaller scale structures and especially in areas with high texture variation. Contrary, it is exactly in those regions where a keypoint will be detected. As a result, local texture warping harms more than it helps.

For future work, it is conceivable to regard dominant normal estimates as salient directions as done for 3D scenes in Sec. 5.5 and detect features in the whole image, rather than only in segments. Though, due to the lack of the underlying 3D structure strong distortions are present in the rectified images and a method for efficient outlier filtering is crucial.

6 Voting Based Camera Pose Estimation

Now that we have discussed how global models can be obtained from partial recordings and with only limited overlap, this chapter moves the attention to the task of localizing a single image within a 3D model. We study the benefits and limitations of spatial verification compared to appearance-based filtering of correspondences. Our novel voting-based pose estimation strategy exhibits $O(n)$ complexity in the number of matches and thus facilitates to consider much more matches than previous approaches – with the direct consequence that we are able to surpass state-of-the-art localization performance for large-scale datasets.

Let us assume that the 3D model has been reconstructed from a set of database images using SfM methods, or that a dense model is augmented with texture information. Then, we can obtain a sparse point cloud, where each 3D point is augmented with a set of local image features and associated with their local descriptors. Establishing 2D-3D correspondences between 2D image observations and 3D points are typically validated via the consecutive application of a n-point-pose solver (Bujnak et al., 2008) inside a RANSAC loop. Though, for large-scale scenarios with inlier rations as small as 1% the executional time will grow beyond what is viable. In addition, Lowe's widely used ratio test (Lowe, 2004) rejects more and more correct matches and thus often fails in these cases. In contrast, we propose to

> *shift the task of finding correct correspondences from the matching stage to the pose estimation step, by leveraging geometric cues extensively, which are local and thus independent of the model size.*

First, instead of using 1^{st} nearest neighbors and only retaining matches that are likely to be inliers, we simplify the matching problem and consider 1-to-many correspondences. This results in a large number of matches with a very small

inlier ratio. Second, we aim to perform extensive spatial verification early on in the pose estimation procedure. As a consequence, geometric verification needs to be scalable to thousands of tentative correspondences to remain applicable. We introduce a voting based spatial verification process that exploits a known gravity direction and an approximate knowledge of the camera height using a setup similar to Svärm et al. (2014). Exemplary results of our voting procedure are illustrated in Fig. 6.1. Our contributions are as follows:

- We formulate spatial verification as a Hough voting problem in pose parameter space, obtaining a run-time that grows only *linearly* in the number of matches.
- We show that we can detect a large fraction of wrong matches using simple but efficient filtering operations based on local (image) geometry.
- Our approach naturally integrates and profits from pose priors, *e.g.*, from GPS data, inertial measurements or vanishing point information, when those are available.
- Our formulation yields a multi-modal distribution over possible camera poses without any additional cost and thus is well suited to handle repetitive scenes.
- We study the applicability of different matching strategies and the influence of allowing 1-to-many matches.

The resulting method localizes considerably more images than current state-of-the-art techniques (Li et al., 2012b; Chen et al., 2011; Torii et al., 2013; Arandjelovic and Zisserman, 2014) for image-based localization on large scale datasets and processes tens of thousands matches with an inlier ratio below 1% in a few seconds – which is well beyond what current methods (Li et al., 2012b; Svärm et al., 2014) can handle. Interestingly, while our results demonstrate that geometric constraints are well suited for outlier filtering, they also clearly indicate that simply using more matches does not automatically lead to a better localization. Thus, one intention of this work is to stimulate further research on defining the quality of matches and how to find good correspondences.

The rest of the chapter is structured as follows. The next section presents a review on pose estimation. Sec. 6.2 outlines our voting method, while Sec. 6.4 explains the computation of spatial votes from matches. Sec. 6.5 shows how to exploit local geometric constraints to filter wrong matches. Finally, Sec. 6.6 discusses our experimental evaluation and Sec. 6.7 concludes the chapter with a

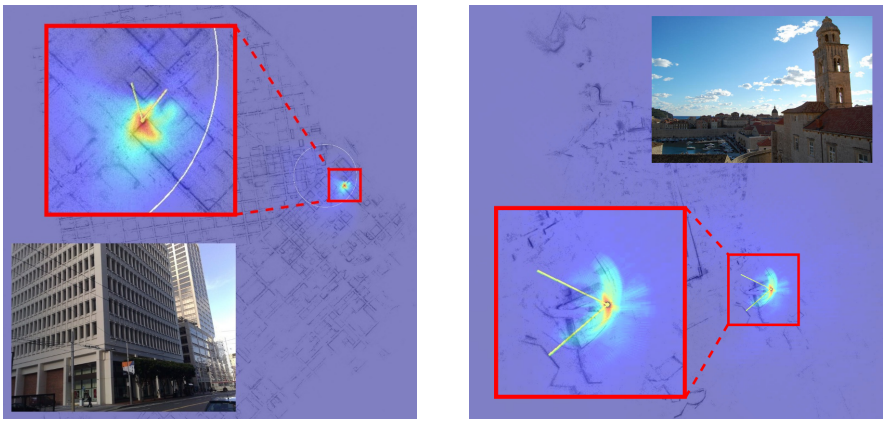


Figure 6.1: Given n 2D-3D matches, our approach makes extensive use of geometric filtering and votes for a 4 DoF camera pose (translation and rotation around the gravity direction) in $\mathcal{O}(n)$ while naturally integrating location priors if available (with circle). The heatmaps encode the number of geometrical correct matches for 2D positions.

discussion. The interested reader is also referred to our accompanying website at `www.cvg.ethz.ch/research/location-voting` for this project.

6.1 Review

There exist two possible approaches to obtain the 2D-3D matches needed for pose estimation. Methods based on *direct matching* perform approximate nearest neighbor search in descriptor space and apply Lowe's ratio test (Lowe, 2004) for outlier filtering. While 2D-to-3D search is inherently more reliable than 3D-to-2D matching (Sattler et al., 2011), state-of-the-art approaches use the latter to recover correspondences missed or rejected during the former (Choudhary and Narayanan, 2012; Li et al., 2012b; Sattler et al., 2012a). This enables them to better counter the problem that the ratio test rejects more and more correct matches for larger datasets due to the increased descriptor space density (Li et al., 2012b; Sattler et al., 2012b). Recently, alternatives (Li et al., 2012b; Svärm et al., 2014) to aggressive outlier filtering during the

matching stage have been proposed. These works are most related to our approach, as they can handle significantly lower inlier ratios. Li et al. (2012b) use co-visibility information to guide RANSAC's sampling process, enabling them to avoid generating obviously wrong camera pose hypotheses. Following a setup equivalent to ours, Svärm et al. (2014) derive a deterministic outlier rejection scheme based on a 2D registration problem. The run-time of their method is $\mathcal{O}(n^2 \log n)$, where n is the number of matches, which severely limits the number of correspondences that can be processed in reasonable time. In contrast, the method proposed in this paper runs in time $\mathcal{O}(n)$, enabling us to solve significantly larger matching problems.

Location recognition and *indirect localization* methods apply image retrieval techniques (Chum et al., 2007; Philbin et al., 2007; Cao and Snavely, 2013; Sivic and Zisserman, 2003) to restrict correspondence search to the 3D points visible in a shortlist of retrieved database images. In order to improve the retrieval performance, (Knopp et al., 2010; Schindler et al., 2007a) remove confusing features, (Torii et al., 2013) explicitly handle repetitive structures, and (Irschara et al., 2009) generates synthetic views to increase the robustness to viewpoint changes. Most relevant to this paper are the methods from (Chen et al., 2011; Torii et al., 2013; Zamir and Shah, 2014; Arandjelovic and Zisserman, 2014). Chen et al. (2011) show how to exploit GPS information and viewpoint normalization to boost the retrieval performance. Similar to us, Zamir and Shah (2014) consider multiple nearest neighbors as potential matches, while Arandjelovic and Zisserman (2014) adapt Hamming embedding to account for varying descriptor distinctiveness. We show that our approach achieves superior localization while providing the full camera pose.

Finally, Quennesson and Dellaert (2007) find a density of camera viewpoints from high level visibility constraints in a voting like procedure. Similar to us, Baatz et al. (2012) verify geometric consistency early on in their voting for view directions.

6.2 Pose Estimation as a Voting Problem

In this work we relax the matching filter and aim to exploit geometric cues instead. To handle the massive amount of outliers there exists the need for a fast and scalable outlier filter. To this end, we borrow a setup from Svärm et al. (2014) which facilitates geometric constraints on the camera (gravity direction

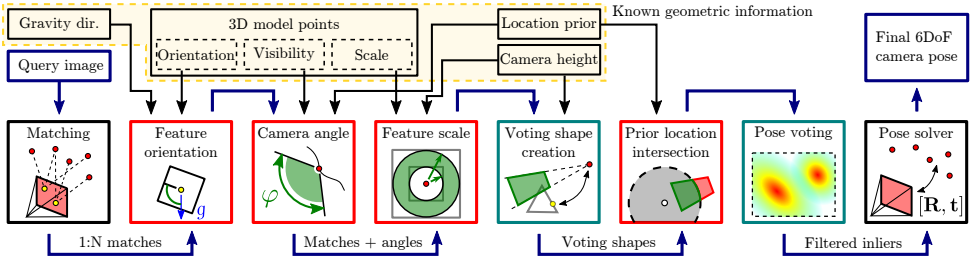


Figure 6.2: Overview of our linear filtering and location voting process (blue path) extensively utilizing spatial verification based on (known) properties. The cyan colored boxes denote steps of the voting procedure and are discussed in Sec. 6.2 and 6.4. Boxes marked in red correspond to the proposed filtering steps based on geometric constraints, which are explained in Sec 6.5.

and approximate height) and transform it to a voting procedure. In addition we augment the voting with other filters utilizing global geometric constraints from the 3D model (feature orientations, visibility and scale of 3D points) and, if available, a positional prior for the camera location. An overview of our linear outlier filter is visualized in Fig. 6.2. It reduces the problem of finding a camera pose that maximizes the number of inliers to several independent 2D voting problems, one for each distinct camera orientation. In the following we will explain the voting procedure in more detail, while Sec. 6.5 covers the proposed geometric filters.

Given the camera gravity direction and assuming that the original camera coordinate system was the identity matrix $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$, we can define a rotation matrix (similar to Eq. (2.11-*p*16))

$$\mathrm{R}_g = \begin{bmatrix} (\mathbf{g} \times \mathbf{e}_z \times \mathbf{g})^\mathrm{T} \\ (-\mathbf{g} \times \mathbf{e}_z)^\mathrm{T} \\ -\mathbf{g}^\mathrm{T} \end{bmatrix} = \begin{bmatrix} -g_x g_z & g_y g_z & g_x^2 g_y^2 \\ -g_y & g_x & 0 \\ -g_x & -g_y & -g_z \end{bmatrix}, \tag{6.1}$$

that transforms the local camera coordinate system into a coordinate frame which is gravity-aligned. Compared to Eq. (2.11-*p*16) here $\mathbf{g}$ defines the vertical direction and the obtained coordinate system has its z-axis point upwards, *i.e.*, it is vertically aligned with the world coordinate system. We assume

that the 3D scene model is gravity-aligned as well. This reduces the pose estimation problem from 6DoF to finding a rotation $\mathrm{R}_\varphi \in \mathbb{R}^{2\times 2}$ around the gravity direction and a translation $\mathbf{t} \in \mathbb{R}^3$. A 2D-3D match between a 3D point $\mathbf{X}$ and a 2D image observation $\mathbf{x}$ is defined to be an inlier, if $\mathbf{X}$ is projected within ε pixels next to $\mathbf{x}$. This is equivalent to the transformed 3D point $\mathbf{X}_g = \mathrm{R}_\varphi \mathbf{X} + \mathbf{t}$ falling into the 3D error cone

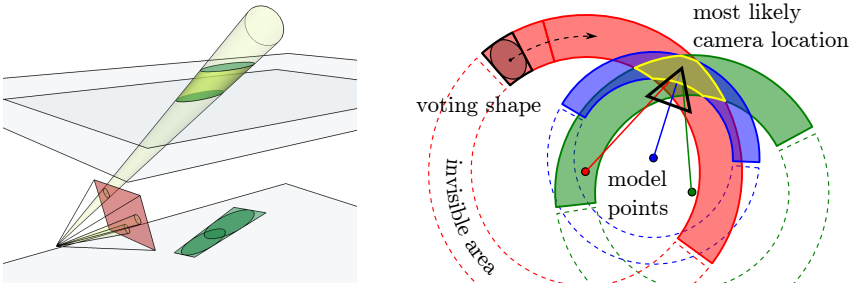
$$\mathbf{c}(\mathbf{x}, \varepsilon) = \nu \cdot \mathbf{r}(\mathbf{x} + \mathbf{u}), \quad \forall \mathbf{u} \in \mathbb{R}^2, \|\mathbf{u}\| = \varepsilon,\ \nu \in \mathbb{R}_{\geq 0} \tag{6.2}$$

defined by the reprojection error ε and $\mathbf{x}$. Here, $\mathbf{r}(\mathbf{x}) = \mathrm{R}_g \mathrm{K}^{-1} (\mathbf{x}\ 1)^\intercal$ is the viewing ray corresponding to $\mathbf{x}$ transformed into the gravity-aligned coordinate frame. We require that the intrinsic calibration K is known. The space of all valid $\mathbf{u}$ can be seen as the offset vectors in the image plane wrt. $\mathbf{x}$, that comprise the area of accepted reprojection error. Assuming that we know the height h of the camera above the ground plane, the problem of registering the 3D point with the cone simplifies to estimating a 2D translation $\mathbf{t}'$ such that

$$\begin{bmatrix} \mathrm{R}_\varphi & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X} + \begin{bmatrix} \mathbf{t}' \\ -h \end{bmatrix} \in \mathbf{c}(\mathbf{x}, \varepsilon), \quad \mathbf{t} = [\mathbf{t}', -h]^\intercal \ . \tag{6.3}$$

As a result the registration problem gets further restricted to the conic section at offset $X_z - h$, *i.e.*, $\mathbf{c}_z(\mathbf{x}, \varepsilon) = X_z - h$ and thus is fully described on a 2D plane.

Obviously we do not know the camera height exactly upfront. However, we can often approximate the ground plane by interpolating the positions of the cameras represented in the model. At the same time, the height of the query camera position is usually close to the ground plane within a certain interval, *e.g.* ±5m. Centering the inverted error cone at the matching 3D point $\mathbf{X}$ and rotating it around gravity direction defines a space in which the camera has to lie (see the following Sec. 6.3 for an explanation of the inversion). Intersecting it with the ground plane thus allows us to estimate the height interval $[h_{min}, h_{max}]$ for the camera pose. This uncertainty in camera height corresponds to intersecting the error cone $\mathbf{c}(\mathbf{x}, \varepsilon)$ by two horizontal planes. As shown by Svärm et al. (2014), we can project these capped error cones onto the ground plane and thus reduce the camera pose estimation to a 2D registration problem between projected error cones (Fig. 6.3a) and projected 3D point positions.



(a) 3D error cone and 2D error shape (b) Intersecting 2D voting shapes

Figure 6.3: (a) Visualization of the 2D error shape generation from a 3D reprojection error cone. (b) Voting shapes are rotated around their 3D points and thus intersect in the most likely camera location (the visualization is marginalized over the set of camera orientations).

Definition 3. *A 2D error shape for a given 2D-3D point correspondence is the union of all projected conic sections between the reprojection error cone* $\mathbf{c}(\mathbf{x}, \varepsilon)$ *and heights in the interval* $[h_{min}, h_{max}]$.

Hence, the uncertainty in camera height is propagated to the camera location, reflected by the larger area covered by the error shape. In case the cone does not intersect the height interval, the correspondence is immediately invalidated.

6.3 Pose Voting

6.3.1 $\Omega(n^2)$ Pose Voting

Assuming that a single match $m = (\mathbf{x}, \mathbf{X})$ is an inlier under a reprojection error of r, there exists a camera pose such that $\mathbf{X}$ perfectly projects to $\mathbf{x}$ while all other inliers have a reprojection error of at most $2r$. In 2D this corresponds to shrinking the error shape $\mathrm{M} \subset \mathbb{R}^2$ of m to contain only its center point $\bar{\mathbf{m}}$ while enlarging the error shapes of all other matches. For a second match m_2, the new error shape $\mathrm{M}_2(m)$ is defined as the Minkowski difference

$$\mathrm{M}_2(m) = \{\mathbf{p}_2 - \mathbf{p}_1 + \bar{\mathbf{m}} \mid \mathbf{p}_2 \in \mathrm{M}_2, \mathbf{p}_1 \in \mathrm{M}\} \ . \tag{6.4}$$

Svärm et al. (2014) use this fact do design a deterministic outlier filter: Given a match m, they propagate the error and determine both an upper and a lower bound on the number of matches that are geometrically consistent with m. Computing these bounds for a single feature takes time $\mathcal{O}(n)$, resulting in an overall run-time of $\Omega(n^2)$ when evaluating each of the n correspondences.

6.3.2 Linear Time Pose Voting

So far the error shapes are defined in the local, gravity-aligned coordinate system of the *camera*. As such, the registration problem can also be imagined as translating and rotating the camera in 2D space and for each unique transformation (discretized in location and angle) count the number of projected 3D points that fall into their voting shape. This procedure is not optimal since the 2D space is unbounded and we would need to test an infinite number of translations.

Thus, we propose to view the problem from a different perspective and to transform the error shapes into the *global* coordinate system; *i.e.*, for a given correspondence we set the projected 3D point position as fixed and by this transform the uncertainty to the camera location. The locations of the transformed error shapes – called voting shapes in the following – thereby also depend on the orientation of the camera. We exploit this fact to design a linear time camera pose estimation algorithm (cf. Fig. 6.3b): Iterating over a fixed set of rotations, each 2D-3D match casts a vote for the region contained in its voting shape. Accumulating these votes in several 2D voting spaces, one per camera orientation, thus enables us to treat every match individually. As a result we obtain a (scaled) probability distribution in the 3-dimensional pose parameter space. The best camera pose is then defined by the orientation and position that obtained most votes. The final 6DoF pose is computed with a 3 point solver inside a RANSAC loop on the voted inlier set. In case of similar structures in the scene, our voting creates a multi-modal distribution. We obtain its modes via non-maximum suppression and verify each of them separately, accepting the pose with most support.

The ideal voting space would be concentric wrt. each matching 3D point (cf. Fig. 6.3b), but this complicates intersection computation significantly. Instead we use a uniform sampling to guarantee $O(n)$ runtime. During voting we account for the quantization by conservatively considering each bin contained in, or intersected by a voting shape.

Proof of Coordinate System Transformation: Consider the error shape M of a match $m = (\mathbf{x}, \mathbf{X})$. Setting the reprojection error for this match to 0 is equivalent to adding uncertainty to the camera position (which is at the camera coordinate system origin $\mathbf{0}$). With $\bar{\mathbf{m}}$ being the center of M, the error shape for the camera location is given by the Minkowski difference $\mathrm{M}_C(m) = \{\mathbf{0} - \mathbf{p} + \bar{\mathbf{m}} \mid \mathbf{p} \in \mathrm{M}\}$. If the match m is correct, the translation from the camera coordinate system to the global world coordinate system (both gravity- and thus axis-aligned) is given as $\mathbf{t}' = \mathbf{X}' - \bar{\mathbf{m}}$, where $\mathbf{X}'$ is the 2D position of the projected point $\mathbf{X}$. Therefore, the camera center in world coordinates has to fall into the global voting shape $\mathrm{V}(m) = \mathrm{M}_C + \mathbf{t}'$, which was obtained without altering the orientation of the camera. For a different orientation, we simply rotate the local camera coordinate system by an angle ϕ before performing a translation $\mathbf{t}'_\phi = \mathbf{X}' - \mathrm{R}_\phi \bar{\mathbf{m}}$. Hence, a rotated voting shape is obtained via

$$\mathrm{V}(m, \phi) = \mathrm{R}_\phi \mathrm{M}_C + \mathbf{t}'_\phi = \{\mathbf{X}' - \mathrm{R}_\phi \mathbf{p} \mid \mathbf{p} \in \mathrm{M}\} \ . \tag{6.5}$$

Eq. (6.5) reveals that changing the camera orientation results in the voting shape being rotated around the 2D position $\mathbf{X}'$ of the matching point (cf. Fig. 6.3b). □

Time Complexity $\mathcal{O}(n)$: First, given a rotation angle, a *single* iteration over all n correspondences is sufficient to aggregate votes for the 2D camera location. Second, to obtain the full distribution and by this the best inlier set also wrt. a discriminative camera angle the procedure needs to be performed separately for k discretized angles. Third, for a large variation in the camera height, the propagated uncertainty leads to less discriminative votes. To avoid this property we quantize the considered height range into l smaller intervals and test for each of them. Consequently, the number of used angle-height pairs is constant, *i.e.*, our method performs $kl \cdot n$ iterations. The size of the voting shapes is bounded as well, thus that we cast a constant number of votes for each shape (In principle a conic section can be unbounded, *e.g.*, a parabola; however, as will be introduced in Sec. 6.5, we leverage the feature scale to constrain its extent). As a result, our approach has an overall computational complexity of $\mathcal{O}(n)$. We will show later that the constant is significantly reduced by our filters, *e.g.*, on average only 8% of the camera orientations need to be tested and as few as 15% of the correspondences survive the geometric tests.

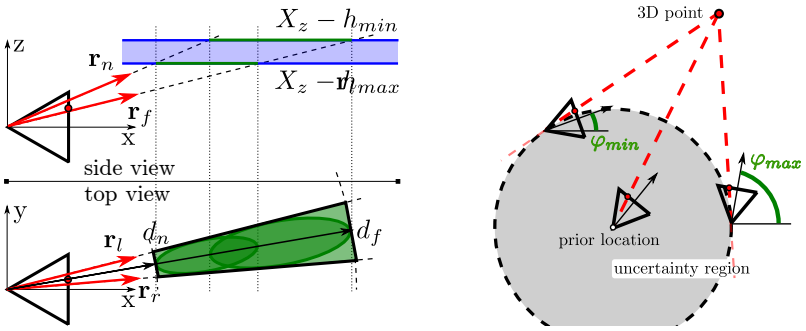
Algorithm 3 Linear time location voting

```
Require: Computed voting shapes, shapes = {M}
  procedure LinearTimeVoting
    for h in heights do
      for φ in angles do
        map ← create new 2d voting map
        for M in shapes[·][h] do
          rotate M by φ
          render M on appropriate resolution
          for cell in rasterization do
            add id(m) to map[cell]
        inliers[h, φ] ← cells with most inliers from map
    return largest set from inliers
```

Implementation Details: Since the size of each voting shape can vary drastically, we use a hierarchical voting approach. For each shape, we select the level in the hierarchy such that all shapes cast at most a fixed number of votes (*e.g.*, for 100 bins). On the finest level, the size of each bin is 0.25m^2. For each level the 2D voting space is implemented as a hash-map (indexed via discretized camera locations) and due to its sparse structure not bounded in space. The height interval is typically ± 5m and discretized in 1m steps. For the angular resolution we chose 2° degrees.
Our approach for linear-time camera pose voting is summarized in Alg. 3. For the creation of voting shapes we refer to Alg. 4 in Sec. 6.5.

6.4 Efficient Voting Shape Computation

In the following we present an efficient computation of the voting shapes and show how to account for the errors introduced by the voting space quantization and gravity direction inaccuracy. In Sec. 6.3.2 we pointed out that a voting- and error-shape only differ by a proper rigid transformation. Thus, we base our derivation on Def. 3 and approximate an error shape via its bounding quadrilateral (cf. Svärm et al., 2014) for efficiency. The quadrilateral can be described via its near and far distance $d_{\text{n|f}}$ to the camera center and the two bounding rays $\mathbf{r}_{\text{l|r}}$ (projected on the ground plane), as illustrated in Fig. 6.4a.



(a) Voting shape construction (b) Camera orientations

Figure 6.4: (a) Voting shape construction: A quadrilateral can be fully described by the rays $\mathbf{r}_{\mathrm{n|f}}$ intersecting the height interval and the left and right bounding rays $\mathbf{r}_{\mathrm{l|r}}$. (b) The location prior constrains the possible camera orientations for a match to the interval $[\varphi_{min}, \varphi_{max}]$ (if the corresponding 3D point is outside the uncertainty region).

A quadrilateral for a particular camera orientation is then efficiently computed by rotating the projected rays $\mathbf{r}_{\mathrm{n|f}}$ as derived in Eq. (6.5).

Without loss of generality let us define that the camera is shifted to the world coordinate system origin and that the projected optical axis points in the x direction in gravity aligned camera coordinates. The most left and right rays have extremal y value; *i.e.*, we are looking for stationary points of the y-component of $\mathbf{c}(\mathbf{x}, \varepsilon)$. The cone parameterization from Eq. (6.2) for $\nu = 1$ describes points on the image plane with reprojection error ε. Therefore, $\mathbf{r}_{\mathrm{l|r}}$ intersect the image plane at keypoint offsets

$$\mathbf{u}^*_{\mathrm{l|r}} = \arg\min_{\mathbf{u},\lambda} r_y(\mathbf{x}+\mathbf{u}) + \frac{\lambda}{2}(\mathbf{u}^\mathrm{T}\mathbf{u} - \varepsilon^2) \ . \tag{6.6}$$

With the cone formulation from Eq. 6.2 and f denoting the focal length of the camera, the derivations wrt. $\mathbf{u}$ and λ are computed to

$$\begin{pmatrix} r_{21} \\ r_{22} \end{pmatrix} \frac{1}{f} + \lambda\mathbf{u} = 0 \qquad \text{and} \qquad \mathbf{u}^\mathrm{T}\mathbf{u} - \varepsilon^2 = 0 \ , \tag{6.7}$$

such that substituting the former in the latter result in

$$\lambda^2 = \frac{1}{\varepsilon^2 f^2} \begin{pmatrix} r_{21} \\ r_{22} \end{pmatrix}^\mathsf{T} \begin{pmatrix} r_{21} \\ r_{22} \end{pmatrix} \tag{6.8}$$

and solving the optimization problem wrt. to $\mathbf{u}$ finally leads to

$$\mathbf{u}^*_{\mathrm{l|r}} = \mp \varepsilon \begin{pmatrix} r_{21} & r_{22} \end{pmatrix}^\mathsf{T} \Big/ \left\| \begin{pmatrix} r_{21} & r_{22} \end{pmatrix}^\mathsf{T} \right\| . \tag{6.9}$$

In a similar manner the offsets corresponding to the near and far rays are derived as

$$\begin{aligned} \mathbf{u}^*_{\mathrm{n|f}} &= \arg\min_{\mathbf{u},\lambda} r_z(\mathbf{x}+\mathbf{u}) + \frac{\lambda}{2}(\mathbf{u}^\mathsf{T}\mathbf{u} - r^2) \\ &= \mp \varepsilon \begin{pmatrix} r_{31} & r_{32} \end{pmatrix}^\mathsf{T} \Big/ \left\| \begin{pmatrix} r_{31} & r_{32} \end{pmatrix}^\mathsf{T} \right\| \quad . \end{aligned} \tag{6.10}$$

Taking into account that $(r_{21},\, r_{22}) = (-g_y,\, g_x)$ and $(r_{31},\, r_{32}) = -(g_x,\, g_y)$, they turn out to be orthogonal to $\mathbf{u}^*_{\mathrm{l|r}}$. Further, and as one would expected the image plane offsets are (i) along the gravity direction projected into the image or orthogonal to it, respectively, (ii) *independent* of the particular feature location $\mathbf{x}$, and (iii) also independent of the camera intrinsics.

To account for the bounded heights, $\mathbf{r}_{\mathrm{n|f}} = \mathbf{r}(\mathbf{x} + \mathbf{u}^*_{\mathrm{n|f}})$ is intersected at heights $h_{\mathrm{n|f}} = \{X_z - h_{max}, X_z - h_{min}\}$, resulting in the distances of the error shape to the camera, *i.e.*,

$$d_i = \left\| \mathbf{r}_{i_{x:y}} \frac{h_i}{\mathbf{r}_{i_z}} \right\|, \quad \forall i \in \{\mathrm{n}, \mathrm{f}\} \ . \tag{6.11}$$

To account for the discretization in angles, $\mathbf{r}_{\mathrm{l|r}} = \mathbf{r}(\mathbf{x} + \mathbf{u}^*_{\mathrm{l|r}})$ is rotated apart around the z-axis by half the angular resolution.

6.4.1 Accounting for Gravity Direction Uncertainty

The measurement of the camera gravity direction is likely to exhibit a certain amount of noise, which we want to account for during voting. The introduced uncertainty will lead to a roll and tilt of the camera and hence rotate a feature point ray and reprojection error cone. In principle the size of a voting shape

will increase if the gravity direction is not known exactly, *i.e.*, the uncertainty is propagated to the voting shape. Therefore, the union of all conic sections of rotated cones now defines the error shape, which is again approximated by a quadrilateral.

For a fixed gravity orientation the keypoint offsets $\mathbf{u}^*$ have been computed before. What remains is to derive the extremal image plane positions in dependence of the camera tilt and roll. We will mainly present the results of our derivation, while more details are covered in Appendix 6.A. The uncertainty of α degrees in the gravity direction can be represented by a rotation of the given gravity vector. Therefore, under a certain rotation, all possible rays for a feature point $\mathbf{x}$ (in aligned camera coordinates) are given by

$$\tilde{\mathbf{r}}(\mathbf{x}, \mathbf{a}) = \mathbf{R}_\alpha(\mathbf{a})\, \mathbf{r}(\mathbf{x}) \ , \tag{6.12}$$

where the rotation matrix is parameterized via the angle α and an axis $\mathbf{a}$ (which lies in the horizontal plane).

First, let us consider the optimization problem for the near and far extremal positions. The closest and farthest intersection with the height interval correlate with the "steepest" and "flattest" ray. Therefore, for a ray of constant length the stationary points on its z-component under rotation are of interest, such that the two optimal rotation axes compute to

$$\begin{aligned} \mathbf{a}^*_{\mathrm{n|f}} &= \arg\min_{\mathbf{a},\lambda} \tilde{r}_z(\mathbf{a}) + \frac{\lambda}{2}(\mathbf{a}^\mathsf{T}\mathbf{a} - 1) \\ &= \mp\big(-r_y,\, r_x\big)^\mathsf{T} \Big/ \big\|\big(-r_y,\, r_x\big)\big\| \ . \end{aligned} \tag{6.13}$$

Second, for the left and right positions the optimization problem wrt. the extremal y-components of rays reads as

$$\mathbf{a}^*_{\mathrm{l|r}} = \arg\min_{\mathbf{a},\lambda} \tilde{r}_y(\mathbf{a}) + \frac{\lambda}{2}\left(\mathbf{a}^\mathsf{T}\mathbf{a} - 1\right) \tag{6.14}$$

It's derivative wrt. $\mathbf{a}$ forms a 2×2 linear system $A(\lambda)\mathbf{a} = \mathbf{b}$. Solving for $\mathbf{a}$ and evaluating the unit norm constraint on $\mathbf{a}$ results in a fourth order polynomial in λ. We compute its roots as the eigenvalues $\lambda_{1\ldots4}$ of the 4×4 Frobenius companion matrix. They are used to evaluate the original function, Eq. (6.12), wrt. its y-component, where we only consider real valued solutions for λ. The

minimum and maximum value define the two rotation axes as

$$\mathbf{a}^*_{\mathrm{l|r}} = \left\{ \arg\min_{\lambda_{1\ldots4} \cap \mathbb{R}} \tilde{r}_y(\mathbf{a}(\lambda)),\ \arg\max_{\lambda_{1\ldots4} \cap \mathbb{R}} \tilde{r}_y(\mathbf{a}(\lambda)) \right\} \quad \text{with} \quad \mathbf{a}(\lambda) = \mathbf{A}(\lambda)^{-1}\mathbf{b} \ . \tag{6.15}$$

Compared to the case with fixed gravity direction, now the extrema positions are *dependent* on the feature position $\mathbf{x}$. This is intuitive, since the further a keypoint is located from the principal point, the more influence a camera tilt and roll will have.

To account for the reprojection error, results from Eq. (6.9) and (6.10) are added and the extremal positions of a cone under gravity uncertainty are

$$\begin{aligned} \mathbf{c}(\mathbf{x}, \mathbf{u}^*_i, \mathbf{a}^*_i) &= \tilde{\mathbf{r}}(\mathbf{x} + \mathbf{u}^*_i, \mathbf{a}^*_i) = \mathbf{R}_\alpha(\mathbf{a}^*_i)\, \mathbf{r}(\mathbf{x} + \mathbf{u}^*_i) \\ &= \tilde{\mathbf{r}}(\mathbf{x}, \mathbf{a}^*_i) + \mathbf{R}_\alpha(\mathbf{a}^*_i)\mathbf{R}_g\mathbf{K}^{-1} \begin{pmatrix} \mathbf{u}^*_i \\ 0 \end{pmatrix} \qquad \forall i \in \{\mathrm{n}, \mathrm{f}, \mathrm{l}, \mathrm{r}\} \ . \end{aligned} \tag{6.16}$$

6.5 Filtering Based On Geometry Constraints

In the following, we present a set of filters that can be applied individually to each match. They are based on geometric relations between properties of the 3D model and local descriptors and aim to reduce the total number of votes to cast. The advantages are twofold: First, the consideration of different camera orientations introduces a fixed constant in terms of computational complexity. By applying some simple filters we can decrease both, the number of relevant matches and the constant time complexity and gain considerable speedup. Second, eliminating false votes upfront boosts the recall rate of our method by up to 20% as will be shown in Sec 6.6.

Relative Feature Orientation: Usually, local descriptors are defined relative to a feature orientation. Similar to Jegou et al. (2008); Baatz et al. (2012), who use orientations to improve image retrieval, we can use the local feature orientation to reject matches. Given the known gravity direction, we express the query feature orientation in a fixed reference frame and compare it to the feature orientations from the database images. The latter typically form an interval of possible feature orientations. A match is rejected, if the query

orientation differs by more than a fixed threshold from the orientations in the interval belonging to the matching 3D point. Notice that this filtering step works similar to upright-normalized descriptors, only that we do not need to warp the query image. Moreover, our filtering works on established correspondences and allows for a weaker rejection via a conservative, experimentally evaluated threshold of 30° degrees.

3D Point Visibility from SfM Model: Local descriptors are not invariant to viewpoint changes. For each 3D point in the scene model, the set of viewpoints under which it was observed is known. This enables us to determine the minimum and maximum rotation angle under which a 3D point is visible. It is used to bound the interval of camera rotations per correspondence for which voting is performed. To account for the viewpoint robustness of feature descriptors, we extend the bounding camera angles for a match by conservative $\pm 60°$ degrees in each direction[1].

Feature Scale: We also utilize the scale at which a feature was detected in the image to reason about the feasibility of a correspondence. Given a database image with focal length f observing a feature belonging to the 3D point p with scale s_I, we use the concept of similar triangles to obtain the scale $s_{3\mathrm{D}}$ of the 3D point as $s_{3\mathrm{D}} = s_I \cdot d/f$, where d is the depth of p in the local camera coordinate system. All observations of p thus form an interval of 3D scales. Following the same formula, we can use this interval to derive the interval $[d_{min}, d_{max}]$ of possible (camera to 3D point) depth values such that the 3D scales projected into the query image are similar to the scale of the matching feature. As derived in Sec. 6.4 the camera height interval defines the near and far distance (cf. $d_{\mathrm{n|f}}$) between camera and matching 3D point. We can thus limit the extent of the voting shape to the intersection of both distance intervals, rejecting the match if it is empty.

Positional Prior: Besides orientation information, mobile devices often also provide location information (*e.g.*, network-based cell tracking, GPS, etc.). We represent the measured location and an upper bound to its uncertainty as a circular area in the voting space. For each match, we then only need to

[1] We found that a usually used threshold of 30° degrees (Mikolajczyk and Schmid, 2004) rejects too many correct matches.

Algorithm 4 Voting shape creation with pre-filtering

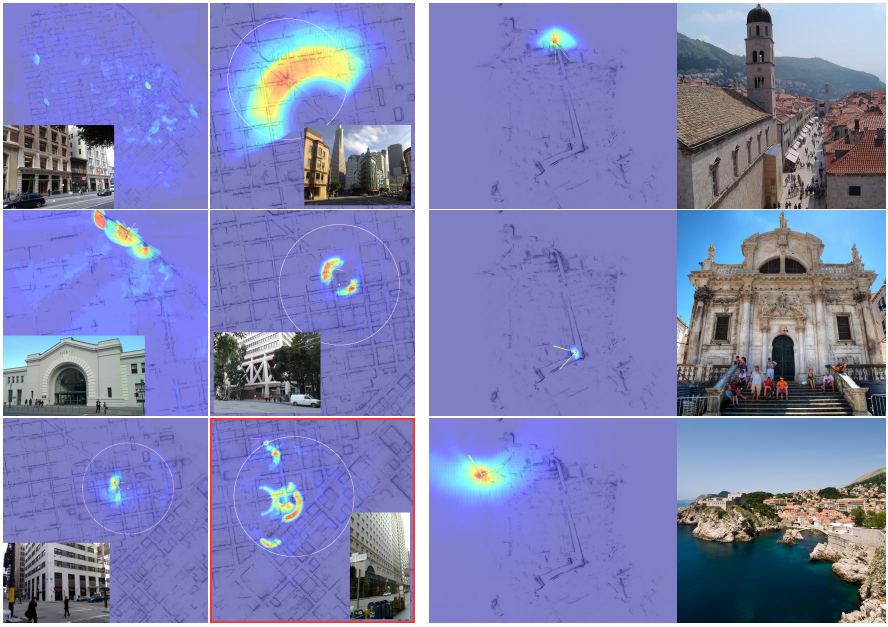
Require: 2D-3D correspondences $\{m\}$
  procedure VOTINGSHAPECREATION
    for m **in** correspondences **do**
      if feature orientations do **not** align **then continue**
      compute $angles[m]$ via visibility & location prior
      if angles are empty **then continue**
      for h **in** $heights$ **do**
        compute $[d_{min}, d_{min}]$ from feature scale
        compute voting shape M
        if M **not** in $[d_{min}, d_{min}]$ **then** reject shape
        else shrink M based on distance limits
          $shapes[m][h] \leftarrow$ M
    return shapes

consider the intersection of its voting shapes with this prior region, usually enabling us to reject many wrong matches early on. This is achieved by our voting formulation in global world coordinates. It allows to directly filter based on the expect camera location and for each correspondence individually, rather than restricting the part of the model to consider (*e.g.* Chen et al., 2011) – which we believe is a much more natural way to include a pose prior. In comparison, Svärm et al. (2014) operate in local camera coordinates where a global location prior is not applicable. In addition there is a strong relation between the orientation of the query camera and its possible locations, which is explained visually in Fig. 6.4b. Using pose priors to limit the set of feasible camera locations thus also restricts the set of feasible rotations for each matching 3D point falling outside the uncertainty region.

Alg. 4 summarizes the use of previously proposed filters in our algorithms.

6.6 Experiments and Results

To evaluate our approach we have conducted experiments on two real-world datasets which are summarized in Tab. 6.1. Exemplary voting results are visualized in Figures 6.5, 6.6 and 6.7.



(a) San Francisco (b) Dubrovnik

Figure 6.5: Exemplary voting results for query images from (a) the San Francisco and (b) the Dubrovnik dataset. Without usage of GPS information votes are cast in the entire map. With GPS (white circles in (a)) the voting is restricted to the uncertainty region. In case of repetitive scenes (2nd row (a)), *e.g.*, similar buildings or symmetric structures, our voting procedure returns a multi-modal distribution. In addition the localization accuracy, *e.g.*, depending on the distance to the scene, is reflected by the size of the returned distribution. The image framed in red shows a failure case.

Figure 6.6: Voting heatmaps for different query images of the San Francisco dataset. The first 9 images were localized correctly. Results marked with a red border depict failure cases, where the bottom left denotes a false positive wrt. the ground truth annotation. We believe it is still correctly localized.

Figure 6.7: Voting heat maps for different query images of the Dubrovnik dataset. All denote a successful localization.

Dataset	San Francisco				Dubrovnik	Aachen
	SF-0	SF-1	PCI	PFI		
DB images	610k	790k	1.06M	638k	6k	3k
3D points	30.34M	75.41M	-	-	1.96M	1.54M
Query images	803	803	803	803	800	273

Table 6.1: Characteristics of the datasets used for evaluation. PCI and PFI are sets of images used for retrieval tasks. SF-0 and SF-1 use parts of PCI to reconstruct a SfM model.

The *San Francisco* dataset (Chen et al., 2011) contains street-view like database images, while query images were captured on mobile devices and provided with (coarse) GPS locations. It is the most challenging dataset for image localization published so far, thus we base our analysis mostly on it. The datasets comes in four different types. Our evaluation is based on SF-0, which has the smallest size and thus represents the most challenging case for localization (unfortunately we could not obtain the SF-1 model). For each query image, its gravity direction is derived from the vertical vanishing point; thereby considering an uncertainty of 2° degrees in the voting procedure (cf. Sec 6.4.1). Fig. 6.8 shows examples of warped images according to the estimated vertical vanishing point (the needed homography was obtained according to. Eq. (2.12)). Note, that within our algorithm we do *not* use the warped images, but the original photos. These images are for illustrative purposes. The alignment of vertical structures in the images demonstrates that estimating the gravity vector from vanishing points is fairly accurate, but not perfect. Consequently, there is a need to handle uncertainty in the gravity direction.

Similar to Chen et al. (2011), we evaluate the performance of our method as recall rate given a fixed precision of 95%. An image is considered to be correctly localized if it registers to points of the correct building ID according to the ground truth annotation; this is the same evaluation criterion as used by Li et al. (2012b). Note that for SF-0, there exists an upper bound on the recall rate of 91.78%, since for 66 query images the corresponding building IDs are missing in the reconstructed model.

Second, we evaluate on the *Dubrovnik* dataset (Li et al., 2010) which is a

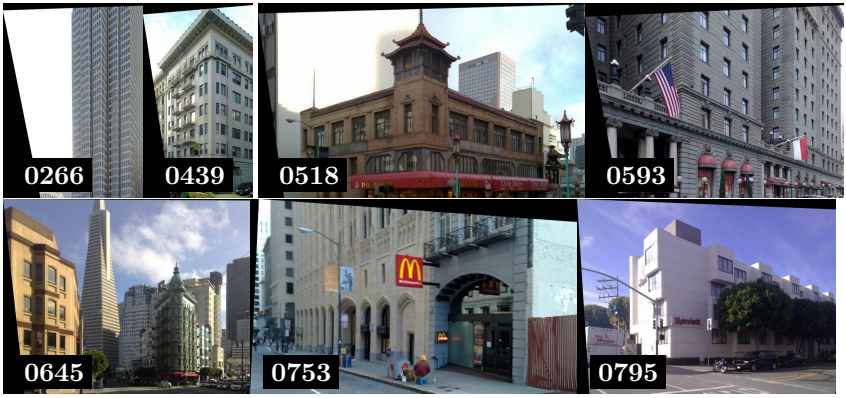


Figure 6.8: Examples of warped, *i.e.*, vertically aligned images according to the computed gravity direction given the estimated vertical vanishing point. The numbers denote the query image ID as defined in the San Francisco dataset.

typical example for a 3D model build from image collections and has been widely used in the literature. As such database and query image follow a similar spatial distribution, which makes pose estimation easier. Consequently localization can be regarded as solved on the dataset, which especially Li et al. (2012b) has shown recently.

Third, we chose the *Aachen* dataset (Sattler et al., 2012c) to study the influence of integrating camera gravity direction uncertainty on the localization performance. In order to do so, we captured 273 query images via a mobile phone and obtained accurate (up to 1° error) camera orientation information from the inertial sensor. Due to the small model size and large number of extracted features, correct localization is guaranteed and enables evaluation for different noise levels.

Correspondence Generation: Similar as others (Li et al., 2012b; Sattler et al., 2012a; Svärm et al., 2014) we use SIFT features for keypoint matching where descriptor entries are in the range 0-255. Matching is performed by approximated nearest neighbor search in a kd-tree structure, which is build from the descriptors belonging to all model observations. For each query feature up to N nearest neighbors are retrieved. To avoid biasing towards a particular

matching strategy, we leverage and evaluate several of them. The studied matching schemes are:

(A) A *ratio test* on descriptor distances for retrieved nearest neighbors with a threshold of 0.7 (baseline) and 0.9 (as used by Li et al., 2012b; Svärm et al., 2014). For 1:N matches the ratio test is performed wrt. the N+1$^{\text{th}}$ neighbor (cf. Zamir and Shah, 2014).

(B) Retrieval of a *constant number of nearest neighbors*.

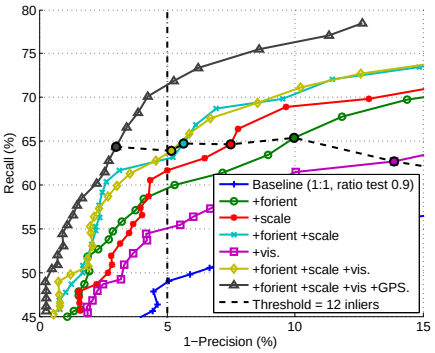
(C) *Absolute thresholding* on the descriptor distance of nearest neighbors to suppress wrong correspondence generation in sparsely populated feature space regions. The threshold of 224 was experimentally obtained from the model by evaluating corresponding descriptors of 3D points (similar to Cao and Snavely, 2014), such that 95% of correct matches survive.

(D) A *variable radius search*, where the search radius is defined by 0.7 times the distance to the nearest neighbor in the query image itself.

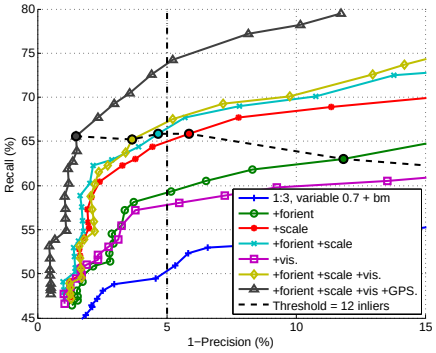
Methods (B) and (C) follow the idea to be independent of the model feature space density, while for (D) an adaptive threshold is estimated via the descriptor density in the query image, which serves as an approximation to the model characteristics. The latter typically returns many correspondence, except for query images containing repetitive structures. All methods can be augmented with a *back-matching* step which verifies that the retrieved 3D point shares the query feature as nearest neighbor in image descriptor space.

6.6.1 Influence of Filters

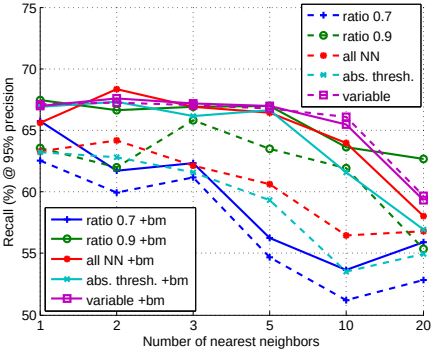
First, we would like to study the influence of our proposed filters. We chose matching (A) with 1 nearest neighbor and a ratio test threshold of 0.9 as our baseline, making it comparable to (Li et al., 2012b; Svärm et al., 2014). Then we apply the different filters individually and sequentially, see Fig. 6.9a. Most impact is observed by constraining the voting shape size to accord to the feature scale, followed by the consistency check on feature orientations. Restricting camera orientations has only limited influence if applied as last filter; however, it successfully serves the purpose of accelerating the voting procedure (the average angular range results in only 28° degrees) without any degradation of the pose estimation results. If a location prior is employed another performance boost of approx. 7% is noticed. In total only 15% of all correspondences survive



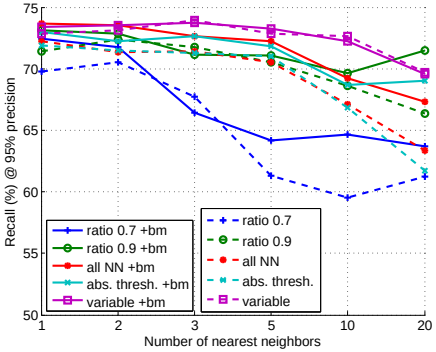
(a) Filter influence, baseline method (1:1 matches, ratio threshold = 0.9)



(b) Filter influence, top performing method (1:3 matches, Matching (D) with back-matching)



(c) Precision over number of matches *without* GPS



(d) Precision over number of matches *with* GPS

Figure 6.9: (a-b) Ablation study for the proposed filters on the SF-0 dataset with the baseline and top performing matching scheme (Legend for filters: forient = feature orientation, scale = feature scale, vis = 3D point visibility, GPS = location prior). (c-d) Recall rate at 95% precision for localization on the SF-0 dataset with different matching strategies and varying location prior (bm = matches additionally verified via back-matching). The marked data points denote the results of Tab. 6.2.

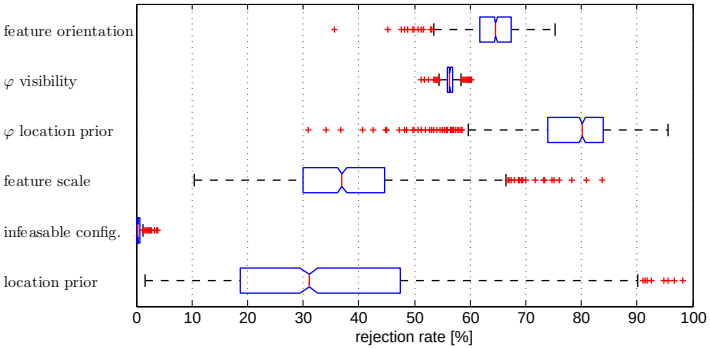


Figure 6.10: Sequential correspondence rejection rates of the different filter stages employed during voting for the San Francisco dataset and 1-3 matching with strategy (D) (variable search radius with image ratio threshold 0.7). For explanation please see the text.

the filtering steps. Often 12 inliers are used as measure to indicate a correct pose (Li et al., 2012b). Employing this threshold, one can notice that recall stays at about 65%, while precision drops significantly without the filters. Summarizing, employing filters based on geometric constraints can lead to a performance increase of more than 20% and a speedup of up to factor 80. For other matching methods a similar influence of filters can be demonstrated, *e.g.*, for the top performing method according to the following paragraph the influence of filters is illustrated in Fig. 6.9b

The rejection rates of the filter stages, *i.e.*, the percentage of correspondences which are eliminated, are illustrated in Fig. 6.10. The numbers denote rejection in sequential order, that means, given all correspondences a median of 64% is rejected based on unaligned feature orientations between query and 3D point features. Out of the surviving 36%, another 56% are not visible assuming a certain camera orientation, *i.e.*, approximately half of the angular range is eliminated upfront – which is explained by the fact that feature points should not be visible on the backside of structures. Another 80% of these remaining angular intervals is neglected due to the constraint posed on the camera orientation given a location prior. 37% of previously surviving correspondences are then rejected based on their invalid feature scale in reference to

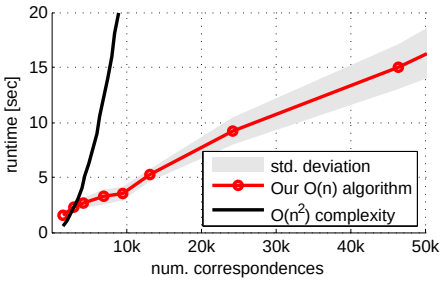


Figure 6.11: Runtime of our voting algorithm for different number of correspondences, showing our $\mathcal{O}(n)$ complexity.

the matching 3D point scales; this is tested individually for each height interval. As a result only 0.3% of remaining correspondences represent an infeasible configuration where a feature reprojection cone and 3D point will never align in 3D (assuming a certain camera height). Finally, 31% of the constructed voting shapes are not considered because they do not intersect with the area defined by the location prior. The remaining correspondences finally vote for a location. On average these are 15% from the initial correspondences within a median angular range of 28.8° degree (8% from 360°). Consequently, with an angular discretization of 2° degrees this accounts for 15 discrete angles to test for (instead of 180 for a full rotation) and leads to a significant speedup of our procedure.

6.6.2 Scalability

In our second experiment we want to study the influence of the different matching procedures providing the input to our algorithm. Results are illustrated in Fig. 6.9c and Fig. 6.9d with varying application of back-matching and a GPS prior. We run our algorithm with 1-3,5,10 and 20 nearest neighbors per keypoint. In the extreme case this accounts for up to 50k correspondences for a single query image. Due to the linear time complexity of our voting procedure the worst case runtime is still only at 16 seconds (cf. Fig 6.11). The obtained results show that the standard ratio test of 0.7 results in considerably worse performance. While a relaxed ratio test of 0.9 is doing significantly better, no significant difference can be noticed towards the other matching schemes.

Method	SF-0		SF-1	Retrieval (PCI)	
	No GPS	GPS	No GPS	No GPS	GPS
Voting without filters	31.0				
— " — + P3P	50.3				
Voting with filters	68.4	73.7			
— " — + **P3P**	**67.5**	**74.2**			
Li et al. (2012b)	54.2		62.5		
Chen et al. (2011)				41 (59)	49 (65)
Torii et al. (2013)	50.9			63	
Arandjelovic and Zisserman (2014)	56.5			78	

Table 6.2: Comparison of our recall rate on the SF-0 dataset for a precision between 94.5% and 95.2%, using matching strategy (D) with at most 3 nearest neighbors. For completeness we also list the results of comparing methods on the SF-1 model and for image retrieval (using histogram equalization and upright features). Results of Chen et al. (2011) on the PCI+PFI images are given in brackets. For (Torii et al., 2013; Arandjelovic and Zisserman, 2014), retrieval results correspond the top ranked image *without* taking precision into account. We compute results for 95% precision via geometric verification on the top 20 candidate images.

This suggests that the ratio test is useless in large-scale localization scenarios and strong geometric filtering is superior by a large margin (cf. Fig 6.9a,6.9b). For matching strategies (B)-(D) and considering different numbers of nearest neighbors, we can notice that the performance is roughly constant up to 5 neighbors and starts to drop only beyond. This proves the effectiveness of our algorithm; *e.g.*, 2000 query keypoints and a required minimum of 12 inliers relate to an inlier ratio as low as 0.12%. However, it is also an interesting result, as it suggest that not necessarily more matches are better, but that there exists a trade-off between rejecting correspondences early on in matching and introducing to much noise in the pose estimation stage.

Method	avg #matches	Registration #imgs	t[sec]	Errors, Quartiles [m] median	1^{st}	3^{rd}	#imgs w/ error <18.3m	>400m
Voting	11265	798	3.78	1.69	0.75	4.82	725	2
RANSAC	56	796	-	0.56	0.19	2.09	744	7
Robust BA	49	794	-	0.47	0.18	1.73	749	13
Svärm et al. (2014)	4766	798	5.06	0.56	-	-	771	3
Sattler et al. (2012a)	≤100	795.5	0.25	1.4	0.4	5.3	704	9

Table 6.3: Comparison of registration performance on the Dubrovnik dataset (no location prior, matching method (D) with 3 nearest neighbors). Li *et al.* Li et al. (2012b) 800 register images, but use an additional guided 3D-2D correspondences search, if the initial 2D-3D matching fails.

6.6.3 Comparison to State-of-the-Art

Finally, we compare our results to state-of-the-art in image based localization and retrieval. Tab. 6.2 lists the evaluation of various forms of our algorithm with and without the usage of GPS information. As can be seen the geometric filters have a significant impact and our approach considerably improves over state-of-the-art. In particular the final P3P pose solver does not improve the localization performance, but provides a refined 6DoF pose. The average inlier ratio was at 0.9%, where RANSAC sample generation and hypothesis evaluation is obviously infeasible. Retrieval methods of Torii et al. (2013) and Arandjelovic and Zisserman (2014) list their recall results without considering precision; *e.g.*, 78% recall also relates to only 78% precision, as each query returns a positive result. For comparability to our method, we leverage the scores after geometric verification as computed by Arandjelovic and Zisserman (2014). For (Torii et al., 2013) no scores are provided. Thus, we establish matches for their top 20 candidates (also contained in the SF-0 model) and run geometric verification with a 3 point pose solver. The estimated camera pose supported by most inliers is then used to compute the precision-recall rate.

For Dubrovnik, our evaluation criterion is equivalent to (Svärm et al., 2014); *i.e.*, an image is considered correctly registered if the estimated pose is sup-

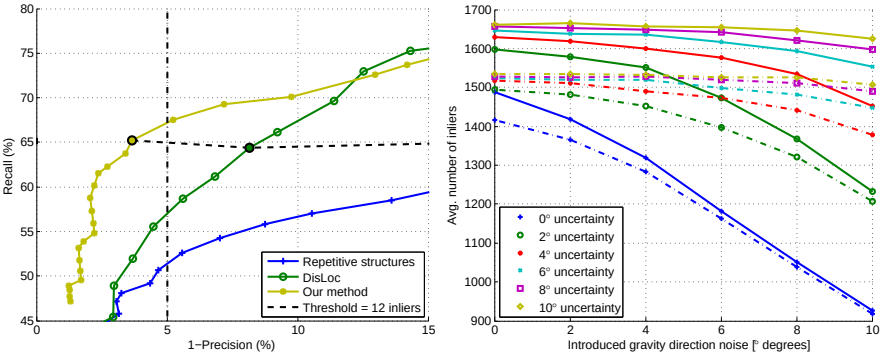
ported by 12 or more inliers under a reprojection error of 6 pixel. Tab. 6.3 lists the results and shows that we achieve state-of-the-art performance. The slightly better numbers of Svärm et al. (2014) wrt. registered images and on the error bounds stem from the fact, that they use an optimal pose solver (which we were not able to obtain from the authors), while we leverage standard 3 point pose RANSAC (Fischler and Bolles, 1981). We also perform final 6DoF pose estimation directly via bundle adjustment on the voted inlier set with a robust Cauchy cost function. The results are convincing: we achieve the smallest median location error and quartile errors reported on the dataset so far. This suggests that the inlier votes reflect a close upper bound on the true inlier set.

6.6.4 Sensitivity to Camera Gravity Direction Uncertainty

To study the influence of the consideration of a certain amount of uncertainty in the gravity direction of a query image, we perform an additional experiment on the Aachen dataset. We introduce artificial noise of up to 10° degrees in the gravity direction used for localization and run experiments with different amounts of uncertainty considered in the voting. As can be seen in Fig. 6.12b, if no uncertainty is accounted for the number of inliers drops significantly with the amount of introduced noise, whereas for a higher assumed uncertainty the average number of inliers stays constant. The disadvantage of considering a lager uncertainty is that a) the upper bound on the inliers is less tight (*i.e.*, more iterations are needed in RANSAC) and b) the localization accuracy of the voted poses decreases.
The Aachen dataset is small and localization is rather easy (*e.g.*, all 273 query images can be localized well) – which can also be noticed by means of the high numbers of obtained inliers. On the other hand, on the San Francisco or Dubrovnik datasets the number of inliers is considerably smaller. However, there are no ground truth gravity directions provided along with the query images, making the validity and expressiveness of an evaluation wrt. the influence of gravity direction uncertainty on these datasets limited.

6.7 Discussion

In this work, we have proposed a novel camera pose estimation technique based on Hough voting, including a set of simple filtering operations, all



(a) Localization performance for image retrieval results

(b) Gravity direction uncertainty

Figure 6.12: (a) Comparison of recall rates over precision of our method against the results we obtained for the image retrieval methods of Torii *et al.* Torii et al. (2013) (Repetitive structures) and Arandjelović *et al.* Arandjelovic and Zisserman (2014) (DisLoc). (b) Influence of considering uncertainty in the gravity direction on the number of voted inliers (solid lines) and inliers after RANSAC (dashed lines) for different levels of introduced noise in the gravity direction of the query images. Localization was performed on the Aachen dataset Sattler et al. (2012c), for which we captured query images with a mobile phone and thus had accurate orientation information obtained from the built in inertial sensors (which is not available for the San Fransisco or Dubrovnik dataset).

employing strong geometric constraints. The run-time of our method grows linearly with the number of matches and thus allows to handle huge amounts of correspondences in a reasonable time. Consequently, we have been able to study the influence of spatial filtering vs. aggressive rejection during matching and have shown the advantages of the former by achieving superior localization performance compared to state-of-the-art. Even though we are able to handle thousands of matches, our results also demonstrate that using more matches does not necessarily lead to a better localization performance.

The runtime requirements of the initial matching between extracted image features and model points represent a bottleneck of our method, since it dependents on the actual model size. In this regard, fast matching, *e.g.*, via hamming embedding of descriptors, or even constant time correspondences search by means of an inverted file index is desirable. However, preliminary results that we have obtained in this direction were not promising: the sheer number of potential correspondences (up to the size of the model itself) distort the extracted inlier sets considerably, which prohibits correct pose estimated in many cases. Hence, we regard the initial matching still as an important step, that can not simply be replaced via outlier filtering.

6.A Derivations Regarding Gravity Direction Uncertainty

In the following section we want to give a detailed derivation for our solution of the computation of voting shapes under the influence of gravity direction uncertainty. In principle the size of a voting shape will increase if the gravity direction is not known exactly, *i.e.*, the uncertainty is propagated to the voting shape. The two rays denoted as $\mathbf{r}_n$ and $\mathbf{r}_f$ intersect with the height interval and by this define the near and far distance of the error shape from the camera. The left and right ray, named $\mathbf{r}_l$ and $\mathbf{r}_r$, are projected onto the ground plane to retrieve the side faces of the quadrilateral.

Under a certain rotation, all possible rays for a feature point $\mathbf{x}$ (in aligned camera coordinates) are given by

$$\tilde{\mathbf{r}}(\mathbf{x}, \mathbf{a}) = \mathbf{R}_\alpha(\mathbf{a})\, \mathbf{r}(\mathbf{x}) \ . \tag{6.17}$$

In the following we drop the dependency of $\mathbf{r}$ on the feature location $\mathbf{x}$ for clarity and denote the components of the rotated ray as $\tilde{\mathbf{r}} = (\tilde{r}_x, \tilde{r}_y, \tilde{r}_z)^\mathsf{T}$. The rotation matrix $\mathbf{R}_\alpha(\mathbf{a})$ is parameterized via the angle α and an axis $\mathbf{a}$ which lies in the horizontal plane ($a_z = 0$), and thus reads as

$$\begin{aligned} \mathbf{R}_\alpha(\mathbf{a}) &= \cos\alpha\, \mathbf{I} + \sin\alpha\, [\tilde{\mathbf{a}}]_x + (1 - \cos\alpha)\, \tilde{\mathbf{a}}\tilde{\mathbf{a}}^\mathsf{T} \\ &\text{s.t.} \quad \mathbf{a} = (a_x, a_y)^\mathsf{T}, \quad \tilde{\mathbf{a}} = (\mathbf{a}, 0)^\mathsf{T}. \quad \|\mathbf{a}\| = 1 \\ &\text{with } [\tilde{\mathbf{a}}]_x = \begin{bmatrix} 0 & 0 & a_y \\ 0 & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \text{ and } \tilde{\mathbf{a}}\tilde{\mathbf{a}}^\mathsf{T} = \begin{bmatrix} a_x^2 & a_x a_y & 0 \\ a_x a_y & a_y^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{6.18}$$

As a result, for a given feature point $\mathbf{x}$ the bounding rays will depend on the rotation and by this on a certain rotation axis $\mathbf{a}$.

First, for the near and far extremal position stationary points of the z-component of rays are of interest, such that the two extremal rotation axes are

$$\begin{aligned} \mathbf{a}^*_{\mathrm{n|f}} &= \arg\min_{\mathbf{a},\lambda} \tilde{r}_z(\mathbf{a}) + \frac{\lambda}{2}(\mathbf{a}^\mathsf{T}\mathbf{a} - 1) \\ &= \arg\min_{\mathbf{a},\lambda} \sin\alpha \left(-a_y, a_x, \frac{\cos\alpha}{\sin\alpha}\right) \mathbf{r} + \frac{\lambda}{2}\left(\mathbf{a}^\mathsf{T}\mathbf{a} - 1\right), \end{aligned} \tag{6.19}$$

where we have included the constraint that $\mathbf{a}$ is expected to have unit length via a Lagrangian multiplier. It's derivatives wrt. $\mathbf{a}$ and λ result in

$$\sin\alpha \begin{pmatrix} r_y \\ -r_x \end{pmatrix} + \lambda \mathbf{a} = 0 \qquad \text{and} \qquad \mathbf{a}^\mathsf{T}\mathbf{a} - 1 = 0 \ . \tag{6.20}$$

Solving the first wrt. $\mathbf{a}$ and substituting the result in the latter, gives

$$\lambda^2 = \sin^2\alpha \begin{pmatrix} r_y \\ -r_x \end{pmatrix}^\mathsf{T} \begin{pmatrix} r_y \\ -r_x \end{pmatrix} \ . \tag{6.21}$$

Thus, the two rotation axes leading to rays for the computation of the near and far quadrilateral boundaries are

$$\mathbf{a}^*_{\mathrm{n|f}} = \mp\big(-r_y,\, r_x\big)^\mathsf{T} \Big/ \big\|\big(-r_y,\, r_x\big)\big\| \, . \tag{6.22}$$

As one would expect, the rotation axis is orthogonal to the feature ray and thus a rotation by α corresponds to a maximum tilting of the camera.

Second, for the left and right positions the optimization problem wrt. the extremal y-components of rays reads as

$$\begin{aligned} \mathbf{a}^*_{\mathrm{l|r}} &= \arg\min_{\mathbf{a},\lambda} \tilde{r}_y(\mathbf{a}) + \frac{\lambda}{2}\left(\mathbf{a}^\mathsf{T}\mathbf{a} - 1\right) \qquad (6.23) \\ &= \arg\min_{\mathbf{a},\lambda} \left(a_x a_y,\, \frac{\cos\alpha}{1-\cos\alpha} + a_y^2,\, \frac{-\sin\alpha}{1-\cos\alpha} a_x \right) \mathbf{r} + \frac{\lambda}{2}\left(\mathbf{a}^\mathsf{T}\mathbf{a} - 1\right) \ . \end{aligned}$$

It's derivative wrt. $\mathbf{a}$ forms a 2×2 linear system

$$\underbrace{\begin{bmatrix} \lambda & r_x \\ r_x & 2r_y + \lambda \end{bmatrix}}_{\mathbf{A}(\lambda)} \mathbf{a} = \underbrace{\begin{pmatrix} \frac{\sin\alpha}{1-\cos\alpha} r_z \\ 0 \end{pmatrix}}_{\mathbf{b}=(b_1,\, b_2)} \ . \tag{6.24}$$

By solving for $\mathbf{a}$ and evaluating the unit norm constraint on $\mathbf{a}$ we obtain

$$\mathbf{a}^\mathsf{T}\mathbf{a} = \mathbf{b}^\mathsf{T}\mathbf{A}^{-\mathrm{T}}\mathbf{A}^{-1}\mathbf{b} = \frac{b_1^2}{\big(\lambda\,(2r_y + \lambda) - r_x^2\big)^2} \begin{pmatrix} 2r_y + \lambda \\ -r_x \end{pmatrix}^\mathsf{T} \begin{pmatrix} 2r_y + \lambda \\ -r_x \end{pmatrix} = 1 \ . \tag{6.25}$$

This results in a fourth order polynomial in λ according to

$$p(\lambda) = \lambda^4 + c_3\lambda^3 + c_2\lambda^2 + c_1\lambda + c_0 \tag{6.26}$$
$$\text{with} \quad \begin{matrix} c_3 = 4r_y & c_2 = 4r_y^2 - 2r_x^2 - b_1^2 \\ c_1 = 2r_y b_1^2 - 4r_x^2 r_y & c_0 = r_x^4 - 4r_x^2 r_y^2 b_1^2 \end{matrix} .$$

We compute its roots as the eigenvalues $\lambda_{1\ldots4}$ of the 4×4 Frobenius companion matrix

$$\mathbf{C}(p) = \begin{bmatrix} 0 & 0 & 0 & -c_0 \\ 1 & 0 & 0 & -c_1 \\ 0 & 1 & 0 & -c_2 \\ 0 & 0 & 1 & -c_3 \end{bmatrix} . \tag{6.27}$$

As mentioned in Sec. 6.4.1 we only consider real valued solutions for λ, and use them to evaluate the original objective function in Eq. (6.12) wrt. its y-component. Consequently, the minimum and maximum value define the two rotation axes as

$$\mathbf{a}^*_{\mathrm{l|r}} = \left\{ \arg\min_{\lambda_{1\ldots4}\cap\mathbb{R}} \tilde{r}_y(\mathbf{a}(\lambda)),\ \arg\max_{\lambda_{1\ldots4}\cap\mathbb{R}} \tilde{r}_y(\mathbf{a}(\lambda)) \right\} \tag{6.28}$$
$$\text{with} \quad \mathbf{a}(\lambda) = \mathbf{A}(\lambda)^{-1}\mathbf{b} .$$

7 Conclusion

This thesis has presented novel algorithms to model indoor environments from images, to automatically calibrate cameras and depth sensors jointly, as well as to register images or partial models against another 3D model. Our attention has been on cases where existing algorithms were prone to fail due to different reasons, *e.g.*, texture-less regions in stereo vision or wide-base settings in registration tasks, or required artificial landmarks and sometimes even manual interaction so far. We have been guided by the ambition to develop algorithms that not only work robustly in these cases, but also exhibit a low computational complexity while at the same time increasing the solution accuracy. To that end, we have followed the idea to leverage geometric priors and exploit available 3D structure information where possible. The experimental evaluation of our proposed solutions has demonstrated that the usage of geometry information is well applicable in the different application scenarios, and that it can help to overcome problems of previous approaches.

With the current trend of computer vision applications playing a major role in diverse industrial applications as well as consumer products, we see the need for algorithms that are robust to failure cases and are guaranteed to provide results with high accuracy in general. This is a major challenge for general purpose formulations; however, in many cases the particular application scenario and its operating environment are constrained. We believe that especially for these use cases the utilization of prior knowledge or available information from complementary sensors is a crucial step to advance the performance of computer vision systems, and we envision that major contributions will be achieved based on this principles. The results of thesis are steps in this direction for different application scenarios in 3D computer vision.

In the following, we summarize our work and point out the main contributions of the individual chapters. We then conclude this work by identifying open problems and directions for future research.

7.1 Summary and Contributions

In Chapter 3 we have considered the classical two-view problem and proposed an algorithm for stereo reconstruction of building interiors. Since indoor environments typically exhibit only little texture, the dense correspondence problem is hard to solve. We bootstrap the extraction of a meaningful geometry by enforcing a strong geometry prior that favors vertical wall elements. Our main contributions are the mathematical representation of this prior, its incorporation in the stereo estimation problem, and an efficient optimization formulation via dynamic programming. The resulting algorithm was shown to run at interactive frame-rates and to provide visually pleasing results.

Next, we have accounted for the recent advance of cheap and easy to use RGB-D sensors – that can replace computational stereo in indoor settings – in Chapter 4. We have presented a structure-based auto-calibration approach which utilizes a sparse model as calibration target for determining the extrinsic pose (between the color camera and depth sensor), as well as the present distortion pattern in the depth measurements. It makes the previously used artificial calibration targets unnecessary, and allows for automatic (re-)calibration or 3D modeling from already captured datasets.

For the registration of RGB-D scans, we have considered a wide-baseline setting between individual scans in Chapter 5. Image distortions resulting from perspective effects thereby hinder the straight forward application of standard features for correspondence search. We have proposed to exploit the observed scene geometry to generate viewpoint independent image representations. Besides the demonstration of the practicability of developable surfaces as well as salient directions (extracted from the 3D model) for this task, our contribution is the presentation of a fully automatic registration approach for scans with only limited overlap.

Finally, Chapter 6 has illustrated the usage of simple, but efficient geometric filters for image-based localization from many tentative correspondences. To make spatial verification scalable and robust to low inlier ratios, we have introduced a camera pose voting procedure which exploits the known camera gravity direction and an approximate height of the camera. Our novel formulation has a linear time complexity in the number of matches and is well suited fro large-scale models containing repetitive structures. Consequently, our approach has been shown to outperform state-of-the art on one of the currently most challenging datasets (San Francisco) for camera pose estimation.

7.2 Future Work

The reconstruction and localization algorithms in this thesis are based on a rigid world assumption and will fail in dynamic environments. For visual modeling from images only, this is still considered a very hard problem and attempts in this direction restrict either the allowed model deformations (*e.g.*, template-based mesh deformations (Perriollat et al., 2011), or articulated motions (Jacquet et al., 2013)) or constrain the camera path (*e.g.*, known static camera poses in motion capture systems (Starck and Hilton, 2007; Joo et al., 2014), or planar motion trajectories (Angst and Pollefeys, 2010)). Clearly, in this setting priors on the geometry and also its temporal deformation will be of great help. On the other hand, RGB-D sensors facilitate the reconstruction of dynamic environments (*e.g.*, Newcombe et al., 2015; Dou et al., 2015) due to their accurate capture of the world configuration at every instant. Though, arbitrary motions and larger-scale reconstructions are still an open research problem. Non-rigid registration between two deformed 3D models is a well studied topic; however, it requires a good initialization of the point-to-point data associations. For visually changing environments (*e.g.*, seasonal changes, structural alterations, or simply different illumination) correspondence search is a long standing, but yet unsolved problem. Higher level reasoning about the temporal order of images and models (Schindler et al., 2007b; Matzen and Snavely, 2014) is an interesting direction. However, for image-based localization the matching itself needs to be invariant to the visual changes. Feature and descriptor learning or learning the matching function itself are seen as promising approaches to advance state-of-the-art.

On the other hand, reconstruction, registration and calibration are not solved in rigid environments either, and there obviously is space for improvement in the proposed algorithms.

Geometric Priors for 3D Modeling: Our proposed priors are quite limited in the sense that they model only vertical or locally planar structures. For the reconstruction of more accurate models, the variety of allowed geometric shapes needs to increase, while at the same time still penalizing wrong configurations. Semantic information (*e.g.*, Häne et al., 2014) or a hierarchy or topology enforced on the prior could help in this regard. Learning the desired shape from training data is a promising direction (also see Sec 3.6) compared to manual modeling, especially with the recent advance of deep learning methods.

Though, the actual parameterization of the prior and its incorporation into the optimization objective is a crucial step. In this regard, we believe that variants of active shape and appearance models (Cootes et al., 1995, 2001) exhibit and interesting direction for 3D reconstruction.

RGB-D Sensor Calibration: Our calibration model is limited by the assumption, that the sensor measurements need to be synchronized. This is often the case for industrial applications where the hardware itself is accessible; however, special care needs to be taken for data captured with asynchronous (commodity) sensors. Thus, future work should consider a calibration model that naturally handles unsynchronized cameras. In addition the consideration of rolling shutter effects for all cameras would allow for an accurate calibration even at fast motions. Finally, it is conceivable to extend the self-calibration for multi-sensor setups, potentially containing non-overlapping cameras in the spirit of Heng et al. (2014).

Registration Problems: Knowledge of the observed scene geometry constitutes a great potential for feature normalization as demonstrated in Sec. 5. It is desirable to achieve equivalently stable matching and alignment without explicitly measuring depth. As sketched in Sec. 5.9, the knowledge of surface normals is sufficient; though, our experiments with estimated pixel-wise surface orientations obtained from a surface normal classifier were of little success. Higher order cues and geometric constraints (*e.g.*, Srajer et al. (2014) in indoor settings) should allow for a more stable and repeatable normalization. A different direction for future research could be to bypass local feature matching at all and regress data associates or even the camera pose directly from image data similar to Shotton et al. (2013).

Camera Pose Estimation: The initial 2D-3D matching in image localization is typically dependent on the model size. Simpler, but still discriminative matching schemes would allows for fast pose estimation. Further improvements could be achieved by using additional filters (Hartmann et al., 2014) or better descriptors (Simonyan et al., 2014). Other interesting directions for future work are the generalization of voting shapes, *e.g.*, via kernel or soft voting (Li, 2006), and to construct a continues voting space that does not require quantization. In addition, the localization of short image sequences (rather than a single

image), *e.g.*, modeled as generalized camera (Pless, 2003), should constrain the pose estimation problem more and allow to resolve ambiguous cases.

Personal Publications

[1] Zeisl, B., Pollefeys, M. (2016). *Structure-Based Auto-Calibration of RGB-D Sensors.* In *ICRA*.

[2] Zeisl, B., Sattler, T., Pollefeys, M. (2015). *Camera Pose Voting for Large-Scale Image-Based Localization.* In *ICCV*, pp. 1–8.

[3] Zeisl, B., Zach, C., Pollefeys, M. (2014). *Variational Regularization and Fusion of Surface Normal Maps.* In *3DV*, pp. 601–608.

[4] Ladicky, L., Zeisl, B., Pollefeys, M. (2014). *Discriminatively Trained Dense Surface Normal Estimation.* In *ECCV*, pp. 468–484.

[5] Zeisl, B., Saurer, O., Sattler, T., Pollefeys, M. (2014). *Using Photographs to Build and Augment 3D Models.* In *Int. Conf. on Information Technology in Landscape Architecture: Digital Landscape Architecture (DLA)*. Zurich.

[6] Zeisl, B., Köser, K., Pollefeys, M. (2013). *Automatic Registration of RGB-D Scans via Salient Directions.* In *ICCV*, pp. 2808–2815.

[7] Zeisl, B., Köser, K., Pollefeys, M. (2012). *Viewpoint Invariant Matching via Developable Surfaces.* In *ECCV, Workshop on consumer depth cameras*, pp. 62–71.

[8] Haene, C., Zach, C., Zeisl, B., Pollefeys, M. (2012). *A Patch Prior for Dense 3D Reconstruction in Man-Made Environments.* In *3DIMPVT*, pp. 563–570.

[9] Zeisl, B., Zach, C., Pollefeys, M. (2011). *Stereo Reconstruction of Building Interiors with a Vertical Structure Prior.* In *3DIMPVT*, pp. 366–373.

Bibliography

Abdel-Aziz, Y., Karara, H. (1971). *Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry.* Proc. ASP/UI Symp. Close-Range Photogrammetry, pp. 1–18.

Agarwal, S., Mierle, K., Others (2015). *Ceres Solver.* `http://ceres-solver.org`. Accessed: 2015-09–5.

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R. (2009). *Building Rome in a day.* In *ICCV*, pp. 105–112.

Aharon, M., Elad, M., Bruckstein, A. (2006). *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation.* Transaction on Signal Processing, 54(11):pp. 4311–4322.

Aiger, D., Mitra, N., Cohen-Or, D. (2008). *4-Points Congruent Sets for Robust Pairwise Surface Registration.* SIGGRAPH, 27(3):pp. 85.1–85.10.

Albl, C., Kukelova, Z., Pajdla, T. (2015). *R6P - Rolling Shutter Absolute Pose Problem.* In *CVPR*, pp. 2292–2300.

Amini, A.A., Weymouth, T.E., Jain, R.C. (1990). *Using Dynamic Programming for Solving Variational Problems in Vision.* TPAMI, 12(9):pp. 855–867.

Angst, R., Pollefeys, M. (2010). *5D motion subspaces for planar motions.* In *ECCV*, pp. 144–157.

Ansar, A., Daniilidis, K. (2003). *Linear pose estimation from points or lines.* TPAMI, 25(5):pp. 578–589.

Arandjelovic, R., Zisserman, A. (2014). *DisLocation : Scalable descriptor distinctiveness for location recognition.* In *ACCV*, pp. 188–204.

Arun, K.S., Huang, T.S., Blostein, S.D. (1987). *Least-squares fitting of two 3-d point sets.* TPAMI, 9(5):pp. 698–700.

Ask, E., Enqvist, O., Kahl, F. (2013). *Optimal Geometric Fitting Under the Truncated L 2 -Norm.* In *CVPR*, pp. 1722–1729.

Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M. (2011). *Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition.* IJCV, 96(3):pp. 315–334.

Baatz, G., Saurer, O., Köser, K., Pollefeys, M. (2012). *Large scale visual geo-localization of images in mountainous terrain.* In *ECCV*, volume 7573, pp. 517–530.

Ballard, D. (1981). *Generalizing the Hough transform to detect arbitrary shapes.* Pattern Recognition, 13(2):pp. 111–122.

Barnea, S., Filin, S. (2008). *Keypoint based autonomous registration of terrestrial laser point-clouds.* ISPRS Journal of Photogrammetry and Remote Sensing, 63(1):pp. 19–35.

Bay, H., Ess, A., Tuytelaars, T., Vangool, L. (2008). *Speeded-Up Robust Features (SURF).* Computer Vision and Image Understanding (CVIU), 110(3):pp. 346–359.

Besl, P.J., McKay, N.D. (1992). *A Method for Registration of 3-D Shapes.* TPAMI, 14(2):pp. 239–256.

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., Hertzberg, J. (2008). *Globally consistent 3D mapping with scan matching.* Robotics and Autonomous Systems, 56(2):pp. 130–142.

Bouguet, J.Y. (2004). *Camera calibration toolbox for matlab.*

Boykov, Y., Veksler, O., Zabih, R. (2001). *Fast approximate energy minimization via graph cuts.* TPAMI, 23(11):pp. 1222–1239.

Bredies, K., Kunisch, K., Pock, T. (2010). *Total Generalized Variation.* SIAM Journal on Imaging Sciences, 3(3):pp. 492–526.

Brown, B.J., Rusinkiewicz, S. (2007). *Global non-rigid alignment of 3-D scans.* In *SIGGRAPH*, p. 21. ACM Press, New York, New York, USA.

Brown, D.C. (1958). *A solution to the general problem of multiple station analytical stereotriangulation.* RCA-MTP Data Reduction Technical Report No. 43.

Brown, D.C. (1966). *Decentering Distortion of Lenses.* Photometric Engineering, 32(3):pp. 444–462.

Bujnak, M., Kukelova, Z., Pajdla, T. (2008). *A general solution to the P4P problem for camera with unknown focal length.* In *CVPR*, pp. 1–8.

Bujnak, M., Kukelova, Z., Pajdla, T. (2011). *New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion.* In *ACCV*, volume 6492, pp. 11–24.

Calonder, M., Lepetit, V., Strecha, C., Fua, P. (2010). *BRIEF : Binary Robust Independent Elementary Features.* In *ECCV*, pp. 778–792. Springer.

Cao, S., Snavely, N. (2013). *Graph-based discriminative learning for location recognition.* In *CVPR*, pp. 700–707.

Cao, S., Snavely, N. (2014). *Minimal Scene Descriptions from Structure from Motion Models.* In *CVPR*, pp. 461–468.

Cao, Y., McDonald, J. (2012). *Improved feature extraction and matching in urban environments based on 3D viewpoint normalization.* Computer Vision and Image Understanding (CVIU), 116(1):pp. 86–101.

Cao, Y., Yang, M., McDonald, J. (2011). *Robust alignment of wide baseline terrestrial laser scans via 3D viewpoint normalization.* In *WACV*, pp. 455–462.

Catmull, E., Rom, R. (1974). *A class of local interpolating splines.* Computer Aided Geometric Design, pp. 317–326.

Chambolle, A., Pock, T. (2010). *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging.* Journal of Mathematical Imaging and Vision, 40(1):pp. 120–145.

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A. (2014). *Return of the Devil in the Details: Delving Deep into Convolutional Nets.* In *BMVC*.

Chen, D.M., Baatz, G., Tsai, S.S. (2011). *City-Scale Landmark Identification on Mobile Devices.* In *CVPR*, pp. 737–744.

Chen, Y., Medioni, G. (1991). *Object Modeling by Registration of Multiple Range Images.* In *ICRA*, pp. 2724–2729.

Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H. (2014). *Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction.* In *CVPR*, pp. 1–8.

Chin, T.J., Purkait, P., Eriksson, A., Suter, D. (2015). *Efficient Globally Optimal Consensus Maximisation with Tree Search.* In *CVPR*, pp. 1–9.

Choudhary, S., Narayanan, P.J. (2012). *Visibility probability structure from SfM datasets and applications.* In *ECCV*, volume 7576, pp. 130–143.

Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A. (2007). *Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval O. Chum, et al.* In *ICCV*, pp. 1–8.

Civera, J., Davison, A., Montiel, J. (2012). *Structure from Motion Using the Extended Kalman Filter.* Springer.

Civera, J., Davison, A.J., Montiel, J.M.M. (2008). *Inverse depth parametrization for monocular SLAM.* IEEE Transactions on Robotics, 24(5):pp. 932–945.

Collins, R.T. (1996). *A space-sweep approach to true multi-image matching.* In *CVPR*, volume 15, pp. 358–363.

Comaniciu, D., Meer, P. (2002). *Mean shift: a robust approach toward feature space analysis.* TPAMI, 24(5):pp. 603–619.

Cootes, T., Taylor, C., Cooper, D., Graham, J. (1995). *Active Shape Models-Their Training and Application.* CVIU, 61(1):pp. 38–59.

Cootes, T.F., Edwards, G.J., Taylor, C.J. (2001). *Active appearance models.* TPAMI, 23(6):pp. 681–685.

Cornelis, N., Leibe, B., Cornelis, K., Van Gool, L. (2008). *3D urban scene modeling integrating recognition and reconstruction.* IJCV, 78(2-3):pp. 121–141.

Corsini, M., Dellepiane, M., Ganovelli, F., Gherardi, R., Fusiello, A., Scopigno, R. (2013). *Fully automatic registration of image sets on approximate geometry.* IJCV, 102(1-3):pp. 91–111.

Crandall, D.J., Owens, A., Snavely, N., Huttenlocher, D.P. (2013). *SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion.* TPAMI, 35(12):pp. 2841–2853.

Curless, B., Levoy, M. (1996). *A volumetric method for building complex models from range images.* In *SIGGRAPH*, pp. 303–312.

Dalal, N., Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection.* In *CVPR*, pp. 886–893.

Davison, A.J. (2003). *Real-time simultaneous localisation and mapping with a single camera.* In *ICCV*, pp. 1403–1410.

Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O. (2007). *MonoSLAM: Real-time single camera SLAM.* TPAMI, 29(6):pp. 1052–1067.

Deng, Y., Lin, X. (2006). *A fast line segment based dense stereo algorithm using tree dynamic programming.* In *ECCV*, volume 3953 LNCS, pp. 201–212.

Dou, M., Taylor, J., Fuchs, H., Fitzgibbon, A., Izadi, S. (2015). *3D Scanning Deformable Objects with a Single RGBD Sensor.* In *CVPR*, pp. 493–501.

Eggert, D., Lorusso, A., Fisher, R. (1997). *Estimating 3-D rigid body transformations: a comparison of four major algorithms.* Machine Vision and Applications, 9(5-6):pp. 272–290.

Elad, M., Aharon, M. (2006). *Image denoising via sparse and redundant representations over learned dictionaries.* Transactions on Image Processing, 15(12):pp. 3736–45.

Engel, J., Schoeps, T., Cremers, D. (2014). *LSD-SLAM: Large-Scale Direct Monocular SLAM.* In *ECCV*, pp. 1–16.

Enqvist, O., Ask, E., Kahl, F., Aström, K. (2012). *Robust Fitting for Multiple View Geometry.* In *ECCV*, pp. 738–751.

Enqvist, O., Josephson, K., Kahl, F. (2009). *Optimal correspondences from pairwise constraints.* In *ICCV*, pp. 1295–1302.

Enqvist, O., Kahl, F. (2008). *Robust Optimal Pose Estimation.* In *ECCV*, pp. 141–153.

Faugeras, O., Luong, Q., Maybank, S. (1992). *Camera self-calibration: Theory and experiments.* In *ECCV*, pp. 321–334.

Faugeras, O.D. (1992). *What can be seen in three dimensions with an uncalibrated stereo rig?* In *ECCV*, volume 588, pp. 563–578.

Faugeras, O.D., Hebert, M. (1986). *The Representation, Recognition, and Locating of 3-D Objects.* Int. Journal of Robotics Research, 5(3):pp. 27–52.

Felzenszwalb, P.F., Huttenlocher, D.P. (2006). *Efficient Belief Propagation for Early Vision.* IJCV, 70(1):pp. 41–54.

Felzenszwalb, P.F., Veksler, O. (2010). *Tiered Scene Labeling with Dynamic Programming.* In *CVPR*, pp. 3097–3104.

Felzenszwalb, P.F., Zabih, R. (2011). *Dynamic programming and graph algorithms in computer vision.* TPAMI, 33(4):pp. 721–40.

Fischler, M.a., Bolles, R.C. (1981). *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Communications of the ACM, 24(6):pp. 381–395.

Fitzgibbon, A.W. (2003). *Robust registration of 2D and 3D point sets.* Image and Vision Computing, 21(13-14):pp. 1145–1153.

Flint, A., Mei, C., Murray, D., Reid, I. (2010a). *A Dynmiac Programming Approach to Reconstructing Building Interiors.* In *ECCV*, pp. 394–407. Springer.

Flint, A., Mei, C., Reid, I., Murray, D. (2010b). *Growing semantically meaningful models for visual SLAM.* In *CVPR*, pp. 467–474.

Forster, C., Pizzoli, M., Scaramuzza, D. (2014). *SVO : Fast Semi-Direct Monocular Visual Odometry.* In *ICRA*, pp. 15–22.

Fraundorfer, F., Tanskanen, P., Pollefeys, M. (2010). *A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles.* In *ECCV*, volume 6314, pp. 269–282.

Fredriksson, J., Enqvist, O., Kahl, F. (2014). *Fast and Reliable Two-View Translation Estimation.* In *CVPR*, pp. 1606–1612.

Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R. (2009a). *Manhattan-World Stereo.* In *CVPR*, pp. 1422–1429.

Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R. (2009b). *Reconstructing building interiors from images.* In *ICCV*, pp. 80–87.

Gallup, D., Frahm, J.M., Mordohai, P., Pollefeys, M. (2008). *Variable baseline/resolution stereo.* In *CVPR*, pp. 1–8.

Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M. (2007). *Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions.* In *CVPR*, pp. 1–8.

Gallup, D., Frahm, J.M., Pollefeys, M. (2010). *Piecewise Planar and Non-Planar Stereo for Urban Scene Reconstruction.* In *CVPR*, pp. 1418–1425.

Ganapathy, S. (1984). *Decomposition of transformation matrices for robot vision.* In *ICRA*, volume 1, pp. 130–139.

Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F. (2003). *Complete solution classification for the perspective-three-point problem.* TPAMI, 25(8):pp. 930–943.

Gee, A., Mayol-Cuevas, W. (2012). *6D Relocalisation for RGBD Cameras Using Synthetic View Regression.* In *BMVC*, pp. 113.1–113.11.

Geiger, A., Moosmann, F., Car, O., Schuster, B. (2012). *Automatic camera and range sensor calibration using a single shot.* In *ICRA*, pp. 3936–3943.

Geiger, A., Roser, M., Urtasun, R. (2010). *Efficient Large-Scale Stereo Matching.* In *ACCV*, pp. 25–38.

Glocker, B., Izadi, S., Shotton, J., Criminisi, A. (2013). *Real-Time RGB-D Camera Relocalization.* In *ISMAR*, pp. 173–179.

Govindu, V.M. (2001). *Combining Two-view Constraints for Motion Estimation.* In *CVPR*, pp. 218–225.

Gower, J.C., Dijksterhuis, G.B. (2004). *Procrustes problems*, volume 3. Oxford University Press.

Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W. (2010). *A tutorial on graph-based SLAM.* Intelligent Transportation Systems Magazine, 2(4):pp. 31–43.

Grunert, J.A. (1841). *Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in Geodäsie.* Grunerts Archiv für Mathematik und Physik, 1:pp. 238–248.

Haene, C., Zach, C., Zeisl, B., Pollefeys, M. (2012). *A Patch Prior for Dense 3D Reconstruction in Man-Made Environments.* In *3DIMPVT*, pp. 563–570.

Häne, C., Savinov, N., Pollefeys, M. (2014). *Class Specific 3D Object Shape Priors Using Surface Normals.* In *CVPR*, pp. 652–659.

Haralick, R.M., Lee, C.n., Ottenberg, K., Nölle, M. (1994). *Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem.* IJCV, 13(3):pp. 331–356.

Harris, C., Stephens, M. (1988). *A combined corner and edge detector.* In *Alvey vision conference*, volume 15, pp. 147–151.

Hartley, R., Trumpf, J., Dai, Y., Li, H. (2013). *Rotation averaging.* IJCV, 103(3):pp. 267–305.

Hartley, R., Zisserman, A. (2004). *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2nd editio edition.

Hartley, R.I. (1992). *Estimation of Relative Camera Positions for Uncalibrated Cameras.* In *ECCV*, pp. 579–587.

Hartley, R.I. (1997). *In defense of the eight-point algorithm.* TPAMI, 19(6):pp. 580–593.

Hartmann, W., Havlena, M., Schindler, K. (2014). *Predicting Matchability.* In *CVPR*, pp. 9–16.

Heckbert, P.S. (1989). *Fundamentals of Texture Mapping and Image Warping.* Master thesis, University of California at Berkeley.

Heinly, J., Dunn, E., Frahm, J. (2012). *Comparative evaluation of binary features.* In *ECCV*, pp. 759–773.

Heng, L., Furgale, P., Pollefeys, M. (2014). *Leveraging Image-based Localization for Infrastructure-based Calibration of a Multi-camera Rig.* Journal of Field Robotics, pp. 775–802.

Henry, P., Fox, D., Bhowmik, A., Mongia, R. (2013). *Patch Volumes: Segmentation-Based Consistent Mapping with RGB-D Cameras.* In *3DV*, pp. 398–405.

Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D. (2012). *RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments.* The International Journal of Robotics Research, 31(5):pp. 647–663.

Heo, Y., Lee, K., Lee, S. (2011). *Robust Stereo Matching Using Adaptive Normalized Cross Correlation.* PAMI, 33(4):pp. 807–822.

Herrera, C., Kannala, J. (2012). *Joint depth and color camera calibration with distortion correction.* TPAMI, 34(10):pp. 2058–2064.

Hirschmüller, H. (2008). *Stereo processing by semiglobal matching and mutual information.* TPAMI, 30(2):pp. 328–341.

Hoiem, D., Efros, A.a., Hebert, M. (2008a). *Closing the loop in scene interpretation.* In *CVPR*, pp. 1–8.

Hoiem, D., Efros, A.a., Hebert, M. (2008b). *Putting objects in perspective.* IJCV, 80(1):pp. 3–15.

Holzer, S., Shotton, J., Kohli, P. (2012). *Learning to efficiently detect repeatable interest points in depth data.* In *ECCV*, volume 7572, pp. 200–213.

Horn, B.K., Brooks, M.J. (1986). *The Variational Approach to Shape from Shading.* Comput. Vision Graph. Image Process., 33:pp. 174–208.

Horn, B.K.P. (1987). *Closed-form solution of absolute orientation using unit quaternions.* Journal of the Optical Society of America A, 4(4):pp. 629–642.

Horn, B.K.P., Hilden, H.M., Negahdaripour, S. (1988). *Closed-form solution of absolute orientation using orthonormal matrices.* Journal of the Optical Society of America A, 5(7):pp. 1127–1135.

Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M. (2013). *Fast cost-volume filtering for visual correspondence and beyond.* TPAMI, 35(2):pp. 504–511.

Huber, D.D.F., Hebert, M. (2003). *Fully automatic registration of multiple 3D data sets.* Image and Vision Computing, 21(7):pp. 637–650.

Huber, P.J., Ronchetti, E.M. (2009). *Robust Statistics.* Wiley-Blackwell, 2nd edition.

Ikehata, S., Aizawa, K. (2014). *Photometric Stereo using Constrained Bivariate Regression for General Isotropic Surfaces.* In *CVPR*, pp. 2187–2194.

Ikeuchi, K., Oishi, T., Takamatsu, J., Sagawa, R., Nakazawa, A., Kurazume, R., Nishino, K., Kamakura, M., Okamoto, Y. (2007). *The Great Buddha Project: Digitally Archiving, Restoring, and Analyzing Cultural Heritage Objects.* IJCV, 75(1):pp. 189–208.

Irschara, A., Zach, C., Frahm, J.M., Bischof, H. (2009). *From structure-from-motion point clouds to fast location recognition.* In *CVPR*, pp. 2599–2606.

Isack, H., Boykov, Y. (2012). *Energy-based geometric multi-model fitting.* IJCV, 97(2):pp. 123–147.

Jacquet, B., Angst, R., Pollefeys, M. (2013). *Articulated and Restricted Motion Subspaces and Their Signatures.* In *CVPR*, pp. 1506—-1513.

Jegou, H., Douze, M., Schmid, C. (2008). *Hamming embedding and weak geometric consistency for large scale image search.* In *ECCV*, volume 5302, pp. 304–317.

Jiang, N., Cui, Z., Tan, P. (2013). *A global linear method for camera pose registration.* In *ICCV*, pp. 481–488.

Jiang, N., Tan, P., Cheong, L.F. (2012). *Seeing double without confusion: Structure-from-motion in highly ambiguous scenes.* In *CVPR*, pp. 1458–1465.

Johnson, A.E., Hebert, M. (1999). *Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes.* TPAMI, 21(5):pp. 433–449.

Joo, H., Park, H.S., Sheikh, Y. (2014). *MAP Visibility Estimation for Large-Scale Dynamic 3D Reconstruction.* In *CVPR*, pp. 1122–1129.

Josephson, K., Byröd, M. (2009). *Pose estimation with radial distortion and unknown focal length.* In *CVPR*, pp. 2419–2426.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F. (2012). *iSAM2: Incremental smoothing and mapping using the Bayes tree.* The International Journal of Robotics Research, 31(2):pp. 216–235.

Kaess, M., Ranganathan, A., Dellaert, F. (2008). *iSAM: Incremental Smoothing and Mapping.* IEEE Transactions on Robotics, 24(6):pp. 1365–1378.

Kahl, F., Hartley, R. (2008). *Multiple-View Geometry Under the {L_infty}-Norm.* TPAMI, 30(9):pp. 1603–1617.

Kalantari, M., Hashemi, A., Jung, F., Guedon, J.P. (2011). *A new solution to the relative orientation problem using only 3 points and the vertical direction.* Journal of Mathematical Imaging and Vision, 39(3):pp. 259–268.

Kälviäinen, H., Hirvonen, P., Xu, L., Oja, E. (1995). *Probabilistic and non-probabilistic Hough transforms: overview and comparisons.* Image and Vision Computing, 13(4):pp. 239–252.

Kanatani, K. (1994). *Analysis of 3-D rotation fitting.* TPAMI, 16(5):pp. 543–549.

Kaneva, B., Torralba, A., Freeman, W.T. (2011). *Evaluation of image features using a photorealistic virtual world.* In *ICCV*, pp. 2282–2289.

Kerl, C., Sturm, J., Cremers, D. (2013). *Dense Visual SLAM for RGB-D Cameras.* In *IROS*, pp. 2100–2106.

King, B., Malisiewicz, T., Stewart, C. (2005). *Registration of multiple range scans as a location recognition problem: Hypothesis generation, refinement and verification.* In *3D Digital Imaging and Modelin*, pp. 180–187.

Klaus, A., Sormann, M., Karner, K. (2006). *Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure.* In *ICPR*, volume 3, pp. 15–18.

Kneip, L., Scaramuzza, D., Siegwart, R. (2011). *A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation.* In *CVPR*, pp. 2969–2976.

Knopp, J., Sivic, J., Pajdla, T. (2010). *Avoiding confusing features in place recognition.* In *ECCV*, volume 6311, pp. 748–762.

Kolmogorov, V., Zabih, R. (2004). *What energy functions can be minimized via graph cuts?* TPAMI, 26(2):pp. 147–59.

Kosecka, J., Zhang, W. (2002). *Video Compass.* In *ECCV*, volume 2353, pp. 476–490.

Köser, K., Koch, R. (2007). *Perspectively Invariant Normal Features.* In *ICCV, Workshop on 3D Representation and Recognition*, pp. 1–8.

Köser, K., Zach, C., Pollefeys, M. (2011). *Dense 3D reconstruction of symmetric scenes from a single image.* In *DAGM*, volume 6835, pp. 266–275.

Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). *Imagenet classification with deep convolutional neural networks.* In *NIPS*, pp. 1097–1105.

Kruppa, E. (1913). *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung.* Sitz.-Ber. d. Math. Naturw. Kaiserlichen Akad. d. Wiss., 122:pp. 1939–1948.

Kühnel, W. (2006). *Differential Geometry: Curves-Surfaces-Manifolds.* American Mathematical Soc.

Kukelova, Z., Bujnak, M., Pajdla, T. (2011). *Closed-form solutions to minimal absolute pose problems with known vertical direction.* In *ACCV*, volume 6493, pp. 216–229.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W. (2011). *g2o: A general framework for graph optimization.* In *ICRA*, pp. 3607–3613.

Ladický, L., Shi, J., Pollefeys, M. (2014). *Pulling Things out of Perspective.* In *CVPR*, pp. 89–96.

Ladicky, L., Zeisl, B., Pollefeys, M. (2014). *Discriminatively Trained Dense Surface Normal Estimation.* In *ECCV*, pp. 468–484.

Lalonde, J.F., Narasimhan, S.G., Efros, A.a. (2010). *What do the sun and the sky tell us about the camera?* IJCV, 88(1):pp. 24–51.

Land, A.H., Doig, A.G. (1960). *An Automatic Method of Solving Discrete Programming Problems.* Econometrica, 28(3):pp. 497–520.

Lee, D.C., Hebert, M., Kanade, T. (2009). *Geometric reasoning for single image structure recovery.* In *CVPR*, pp. 2136–2143.

Lee, G.H., Faundorfer, F., Pollefeys, M. (2013). *Motion estimation for self-driving cars with a generalized camera.* In *CVPR*, pp. 2746–2753.

Lee, G.H., Pollefeys, M., Fraundorfer, F. (2014). *Relative Pose Estimation for a Multi-Camera System with Known Vertical Direction.* In *CVPR*, pp. 540–547.

Lepetit, V., Fua, P. (2006). *Keypoint recognition using randomized trees.* TPAMI, 28(9):pp. 1465–1479.

Lepetit, V., Moreno-Noguer, F., Fua, P. (2009). *EPnP: An accurate O(n) solution to the PnP problem.* IJCV, 81(2):pp. 155–166.

Leutenegger, S., Chli, M., Siegwart, R.Y. (2011). *BRISK: Binary Robust invariant scalable keypoints.* In *ICCV*, pp. 2548–2555.

Levinson, J., Thrun, S. (2013). *Automatic Online Calibration of Cameras and Lasers.* In *RSS*.

Li, B., Heng, L., Lee, G.H., Pollefeys, M. (2013). *A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle.* In *IROS*, pp. 1595–1601.

Li, H. (2006). *A Simple Solution to the Six-Point Two-View Focal-Length Problem.* In *ECCV*, pp. 200–213.

Li, H. (2009). *Consensus set maximization with guaranteed global optimality for robust geometry estimation.* In *ICCV*, pp. 1074–1080.

Li, H., Hartley, R. (2007). *The 3D-3D registration problem revisited.* In *ICCV*, pp. 1–8.

Li, H., Sumner, R.W., Pauly, M. (2008). *Global correspondence optimization for non-rigid registration of depth scans.* Eurographics, 27(5):pp. 1421–1430.

Li, S., Xu, C., Xie, M. (2012a). *A robust O(n) solution to the perspective-n-point problem.* TPAMI, 34(7):pp. 1444–1450.

Li, Y., Snavely, N., Huttenlocher, D.P. (2010). *Location Recognition Using Prioritized Feature Matching.* In *ECCV*, pp. 791–804.

Li, Y., Snavely, N., Huttenlocher, D.P., Fua, P. (2012b). *Worldwide Pose Estimation using 3D Point Clouds.* In *ECCV*, volume 7572, pp. 15–29.

Lindeberg, T. (1994). *Scale-space theory: a basic tool for analyzing structures at different scales.* Journal of Applied Statistics, 21(1):pp. 225–270.

Liu, B., Gould, S., Koller, D. (2010). *Single image depth estimation from predicted semantic labels.* In *CVPR*, pp. 1253–1260.

Liu, L., Stamos, I. (2005). *Automatic 3D to 2D Registration for the Photorealistic Rendering of Urban Scenes.* In *CVPR*, volume 2, pp. 137–143.

Liu, L., Stamos, I. (2007). *A systematic approach for 2D-image to 3D-range registration in urban environments.* In *ICCV*, pp. 1–8.

Liu, L., Stamos, I., Yu, G., Wolberg, G., Zokai, S. (2006). *Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data.* In *CVPR*, volume 2, pp. 2293–2300.

Lo, T.W.R., Siebert, J.P. (2009). *Local feature extraction and matching on range images: 2.5D SIFT.* Computer Vision and Image Understanding, 113(12):pp. 1235–1250.

Longuet-Higgins, H.C. (1981). *A computer algorithm for reconstructing a scene from two projections.* Nature, 293(5828):pp. 133–135.

Lorensen, W.E., Cline, H.E. (1987). *Marching Cubes: A high resolution 3D surface construction algorithm.* Computer Graphics, 21(4):pp. 163–169.

Lowe, D.G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints.* IJCV, 60(2):pp. 91–110.

Lu, C.P., Hager, G.D., Mjolsness, E. (2000). *Fast and globally convergent pose estimation from video images.* TPAMI, 22(6):pp. 610–622.

Lu, F., Milios, E. (1997). *Globally Consistent Range Scan Alignment for Environment Mapping.* Autonomous Robots, 4(4):pp. 333–349.

Luong, Q.T., Faugeras, O.D. (1996). *The Fundamental Matrix: Theory, Algorithms, and Stability Analysis.* IJCV, 17(1):pp. 43–75.

Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S. (2004). *An Invitation to 3D Vision.* Springer, 1st edition.

Maier, R., Sturm, J., Cremers, D. (2014). *Submap-based Bundle Adjustment for 3D Reconstruction from RGB-D Data.* In *GCPR*, pp. 54–65.

Mairal, J. (2010). *Sparse coding for machine learning, image processing and computer vision.* Ph.D. thesis, Ecole Normale Superieure de Cachan.

Makadia, A., Patterson, A.I., Daniilidis, K. (2006). *Fully Automatic Registration of 3D Point Clouds.* In *CVPR*, volume 1, pp. 1297–1304.

Mallick, S.P., Zickler, T.E., Kriegman, D.J., Belhumeur, P.N. (2005). *Beyond Lambert: Reconstructing specular surfaces using color.* In *CVPR*, volume 2, pp. 619–626.

Mastin, A., Kepner, J. (2009). *Automatic registration of LIDAR and optical images of urban scenes.* In *CVPR*, pp. 2639–2646.

Matas, J., Chum, O., Urban, M., Pajdla, T. (2004). *Robust wide-baseline stereo from maximally stable extremal regions.* Image and Vision Computing, 22(10 SPEC. ISS.):pp. 761–767.

Matzen, K., Snavely, N. (2014). *Scene Chronology.* In *ECCV*, pp. 615–630.

Mei, X., Sun, X., Dong, W., Wang, H., Zhang, X. (2013). *Segment-tree based cost aggregation for stereo matching.* In *CVPR*, pp. 313–320.

Mičušík, B., Košecká, J. (2010). *Multi-view Superpixel Stereo in Urban Environments.* IJCV, 89(1):pp. 106–119.

Micusik, B., Wildenauer, H. (2014). *Structure from Motion with Line Segments under Relaxed Endpoint Constraints.* In *3DV*, pp. 13–19.

Mikolajczyk, K., Schmid, C. (2004). *Scale & Affine Invariant Interest Point Detectors.* IJCV, 60(1):pp. 63–86.

Mikolajczyk, K., Schmid, C. (2005). *A Performance evaluation of local descriptors.* TPAMI, 27(10):pp. 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L. (2005). *A comparison of affine region detectors.* IJCV, 65(1-2):pp. 43–72.

Mirzaei, F.M., Kottas, D.G., Roumeliotis, S.I. (2012). *3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization.* Int. Journal of Robotics Research, 31(4):pp. 452–467.

Moghadam, P., Bosse, M., Zlot, R. (2013). *Line-based extrinsic calibration of range and image sensors.* In *ICRA*, pp. 3685–3691.

Moulon, P., Monasse, P., Marlet, R. (2013). *Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion.* In *ICCV*, pp. 3248–3255.

Muja, M., Lowe, D.G. (2013). *Scalable Nearest Neighbour Algorithms for High Dimensional Data.* TPAMI, 36(11):pp. 2227–2240.

Napier, A., Corke, P., Newman, P. (2013). *Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes.* In *ICRA*, pp. 3679–3684.

Newcombe, R., Lovegrove, S., Davison, A. (2011a). *DTAM: Dense Tracking and Mapping in Real-Time.* In *ICCV*, volume 1, pp. 2320–2327.

Newcombe, R.a., Davison, A.J. (2010). *Live dense reconstruction with a single moving camera.* In *CVPR*, pp. 1498–1505.

Newcombe, R.A., Fox, D., Seitz, S.M. (2015). *DynamicFusion : Reconstruction and Tracking of Non-rigid Scenes in Real-Time.* In *CVPR*, pp. 343–352.

Newcombe, R.A., Molyneaux, D., Kim, D., Davison, A.J., Shotton, J., Hodges, S., Fitzgibbon, A. (2011b). *KinectFusion : Real-Time Dense Surface Mapping and Tracking.* In *ISMAR*, pp. 127–136.

Ni, K., Steedly, D., Dellaert, F. (2007). *Out-of-Core Bundle Adjustment for Large-Scale 3D Reconstruction.* In *ICCV*, pp. 1–8.

Nishino, K., Ikeuchi, K. (2002). *Robust simultaneous registration of multiple range images.* In *ACCV*, pp. 23–25.

Nistér, D. (2004). *An efficient solution to the five-point relative pose problem.* TPAMI, 26(6):pp. 756–777.

Nistér, D., Stewénius, H. (2007). *A minimal solution to the generalised 3-point pose problem.* Journal of Mathematical Imaging and Vision, 27(1):pp. 67–79.

Novak, D., Schindler, K. (2013). *Approximate Registration of Point Clouds with Large Scale Differences.* ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5/W2:pp. 211–216.

Olsson, C., Kahl, F., Oskarsson, M. (2009). *Branch-and-bound methods for euclidean registration problems.* TPAMI, 31(5):pp. 783–794.

Ozuysal, M., Calonder, M., Lepetit, V., Fua, P. (2010). *Fast Keypoint Recognition Using Random Ferns.* TPAMI, 32(3):pp. 448–461.

Pandey, G., Mcbride, J.R., Savarese, S., Eustice, R.M. (2012). *Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information.* Journal of Field Robotics, 0(0):pp. 1–27.

Pandey, G., Savarese, S., McBride, J.R., Eustice, R.M. (2011). *Visually bootstrapped generalized ICP.* In *ICRA*, pp. 2660–2667.

Papert, S. (1966). *The summer vision project.*

Perriollat, M., Hartley, R., Bartoli, A. (2011). *Monocular Template-based Reconstruction of Inextensible Surfaces.* IJCV, 95(2):pp. 124–137.

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A. (2007). *Object retrieval with large vocabularies and fast spatial matching.* In *CVPR*, pp. 1–8.

Pless, R. (2003). *Using many cameras as one.* In *CVPR*, volume 2, pp. 587–593.

Pollefeys, M. (1999). *Self-calibration and metric 3D reconstruction from uncalibrated image sequences.* Phd thesis, KU Leuven.

Pollefeys, M., Koch, R., Van Gool, L. (1999). *Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters.* IJCV, 32(1):pp. 7–25.

Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., Others, Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Wel, G., Towles, H. (2008). *Detailed real-time urban 3d reconstruction from video.* IJCV, 78(2):pp. 143–167.

Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R. (2004). *Visual modeling with a hand-held camera.* IJCV, 59(3):pp. 207–232.

Pulli, K. (1999). *Multiview registration for large data sets.* In *3Dim*, pp. 160–168.

Quan, L., Lan, Z. (1999). *Linear N-point camera pose determination.* TPAMI, 21(8):pp. 774–780.

Quennesson, K., Dellaert, F. (2007). *Rao-blackwellized importance sampling of camera parameters from simple user input with visibility preprocessing in line space.* In *3DPVT*, pp. 893–899.

Ramalingam, S., Brand, M., Electric, M. (2013). *Lifting 3D Manhattan Lines from a Single Image*. In *ICCV*, pp. 497–504.

Ranftl, R., Gehrig, S., Pock, T., Bischof, H. (2012). *Pushing the limits of stereo using variational stereo estimation*. 2012 IEEE Intelligent Vehicles Symposium, pp. 401–407.

Roberts, R., Sinha, S.N., Szeliski, R., Steedly, D. (2011). *Structure from motion for scenes with large duplicate structures*. In *CVPR*, pp. 3137–3144.

Robertson, D., Cipolla, R. (2004). *An Image-Based System for Urban Navigation*. In *BMVC*, volume 1, pp. 84.1–84.10.

Rosten, E., Porter, R., Drummond, T. (2010). *Faster and better: A machine learning approach to corner detection*. TPAMI, 32(1):pp. 105–119.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (2011). *ORB: An efficient alternative to SIFT or SURF*. In *ICCV*, pp. 2564–2571.

Rusinkiewicz, S., Levoy, M. (2001). *Efficient variants of the ICP algorithm*. In *3Dim*, pp. 145–152.

Rusu, R., Blodow, N., Beetz, M. (2009a). *Fast point feature histograms (FPFH) for 3D registration*. In *ICRA*, pp. 3212–3217.

Rusu, R., Holzbach, A., Blodow, N., Beetz, M. (2009b). *Fast geometric point labeling using conditional random fields*. In *IROS*, pp. 7–12.

Sattler, T., Leibe, B., Kobbelt, L. (2011). *Fast image-based localization using direct 2D-to-3D matching*. In *ICCV*, pp. 667–674.

Sattler, T., Leibe, B., Kobbelt, L. (2012a). *Improving Image-Based Localization by Active Correspondence Search*. In *ECCV*, pp. 752–765.

Sattler, T., Leibe, B., Kobbelt, L. (2012b). *Towards fast image-based localization on a city-scale*. Lecture Notes in Computer Science, 7474:pp. 191–211.

Sattler, T., Sweeney, C., Pollefeys, M. (2014). *On Sampling Focal Length Values to Solve the Absolute Pose Problem*. In *ECCV*, pp. 828–843.

Sattler, T., Weyand, T., Leibe, B., Kobbelt, L. (2012c). *Image Retrieval for Image-Based Localization Revisited*. In *BMVC*, pp. 76.7–76.12.

Saxena, A., Chung, S.H., Ng, A.Y. (2007). *3-D Depth Reconstruction from a Single Still Image*. IJCV, 76(1):pp. 53–69.

Saxena, A., Sun, M., Ng, A.Y. (2009). *Make3D: Learning 3D Scene Structure from a Single Still Image.* TPAMI, 31(5):pp. 824–40.

Scaramuzza, D. (2011). *1-point-RANSAC Structure from Motion for Vehicle-Mounted Cameras by exploiting Non-holonomic Constraints.* IJCV, 95(1):pp. 74–85.

Scaramuzza, D., Harati, A., Siegwart, R. (2007). *Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes.* In *IROS*, pp. 4164–4169.

Scharstein, D., Szeliski, R. (2002). *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.* IJCV, 47(1):pp. 131–140.

Schindler, G., Brown, M., Szeliski, R. (2007a). *City-scale location recognition.* In *CVPR*, pp. 1–7.

Schindler, G., Dellaert, F., Kang, S.B. (2007b). *Inferring Temporal Order of Images From 3D Structure.* 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7.

Schindler, G., Krishnamurthy, P., Dellaert, F. (2007c). *Line-based structure from motion for urban environments.* In *3DPVT*, pp. 846–853.

Schmid, C., Mohr, R., Bauckhage, C. (2000). *Evaluation of interest point detectors.* IJCV, 37(2):pp. 151–172.

Schmidt, T., Newcombe, R., Fox, D. (2014). *DART: Dense Articulated Real-Time Tracking.* RSS.

Schönemann, P.H. (1966). *A generalized solution of the orthogonal procrustes problem.* Psychometrika, 31(1):pp. 1–10.

Schönemann, P.H., Carroll, R.M. (1970). *Fitting one matrix to another under choice of a central dilation and a rigid motion.* Psychometrika, 35(2):pp. 245–255.

Schwing, A.G., Urtasun, R. (2012). *Efficient Exact Inference for 3D Indoor Scene Understanding.* In *ECCV*, volume 1, pp. 299–313.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R. (2006). *A comparison and evaluation of multi-view stereo reconstruction algorithms.* In *CVPR*, pp. 519–528.

Shi, J., Tomasi, C. (1994). *Good Features to Track.* In *CVPR*, pp. 593–600.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A. (2011). *Real-time human pose recognition in parts from single depth images.* In *CVPR*, pp. 116–124.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A. (2013). *Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images.* In *CVPR*, pp. 2930–2937.

Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D. (2011). *Introduction to autonomous mobile robots.* MIT Press.

Silberman, N., Hoiem, D., Kohli, P., Fergus, R. (2012). *Indoor segmentation and support inference from RGBD images.* In *ECCV*, pp. 1–14.

Simonyan, K., Vedaldi, A., Zisserman, A. (2014). *Using Convex Optimisation.* TPAMI, 36(8):pp. 1–14.

Sinha, S.N., Steedly, D., Szeliski, R. (2009). *Piecewise planar stereo for image-based rendering.* In *ICCV*, pp. 1881–1888.

Sivic, J., Zisserman, A. (2003). *Video Google: a text retrieval approach to object matching in videos.* In *ICCV*, volume 2, pp. 1470–1477.

Smisek, J., Jancosek, M., Pajdla, T. (2011). *3D with Kinect.* In *ICCV, Workshop.*

Smith, P., Reid, I., Davison, A. (2006). *Real-Time Monocular SLAM with Straight Lines.* In *BMVC*, volume 1, pp. 17–26.

Snavely, N., Seitz, S.M., Szeliski, R. (2007). *Modeling the World from Internet Photo Collections.* IJCV, 80(2):pp. 189–210.

Srajer, F., Schwing, A.G., Pollefeys, M., Pajdla, T. (2014). *MatchBox : Indoor Image Matching via Box-like Scene Estimation.* In *3DV*, volume 1, pp. 705–712.

Stamos, I., Leordeanu, M. (2003). *Automated feature-based range registration of urban scenes of large scale.* In *CVPR*, pp. 555–561.

Stamos, I., Liu, L., Chen, C., Wolberg, G., Yu, G., Zokai, S. (2008). *Integrating Automated Range Registration with Multiview Geometry for the Photorealistic Modeling of Large-Scale Scenes.* IJCV, 78(2-3):pp. 237–260.

Starck, J., Hilton, A. (2007). *Surface capture for performance-based animation.* IEEE Computer Graphics and Applications, 27:pp. 21–31.

Steinbrücker, F., Sturm, J., Cremers, D. (2011). *Real-time visual odometry from dense RGB-D images.* In *ICCV, Workshop*, pp. 719–722.

Stewénius, H., Engels, C., Nistér, D. (2006). *Recent developments on direct relative orientation.* ISPRS Journal of Photogrammetry and Remote Sensing, 60(4):pp. 284–294.

Sturm, P. (2005). *Multi-view geometry for general camera models.* In *CVPR*, volume 1, pp. 206–212.

Sun, J., Li, Y., Kang, S.B., Shum, H.Y. (2005). *Symmetric stereo matching for occlusion handling.* In *CVPR*, volume 2, pp. 399–406.

Svärm, L., Enqvist, O., Oskarsson, M., Kahl, F. (2014). *Accurate Localization and Pose Estimation for Large 3D Models.* In *CVPR*, pp. 532–539.

Taylor, C., Kriegman, D. (1995). *Structure and motion from line segments in multiple images.* TPAMI, 17(11).

Teichman, A., Miller, S., Thrun, S. (2013). *Unsupervised intrinsic calibration of depth sensors via SLAM.* In *RSS*.

Telea, A. (2004). *An Image Inpainting Technique Based on the Fast Marching Method.* Journal of Graphics Tools, 9(1):pp. 23–34.

Theiler, P.W., Wegner, J.D., Schindler, K. (2014). *Keypoint-based 4-Points Congruent Sets – Automated marker-less registration of laser scans.* ISPRS Journal of Photogrammetry and Remote Sensing, 96:pp. 149–163.

Theiler, P.W., Wegner, J.D., Schindler, K. (2015). *Globally consistent registration of terrestrial laser scans via graph optimization.* ISPRS Journal of Photogrammetry and Remote Sensing, 109:pp. 126–138.

Tola, E., Lepetit, V., Fua, P., Member, S. (2010). *DAISY: An efficient dense descriptor applied to wide-baseline stereo.* TPAMI, 32(5):pp. 815–830.

Toldo, R., Fusiello, A. (2008). *Robust multiple structures estimation with J-linkage.* In *ECCV*, pp. 537–547.

Torii, A., Sivic, J., Pajdla, T., Okutomi, M. (2013). *Visual place recognition with repetitive structures.* In *CVPR*, pp. 883–890.

Triggs, B. (1999). *Camera pose and calibration from 4 or 5 known 3D points.* In *ICCV*, volume 1, pp. 1–7.

Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A. (2000). *Bundle adjustment—a modern synthesis.* In *Vision Algorithms: Theory and Practice*, pp. 298–372.

Umeyama, S. (1991). *Least-squares estimation of transformation parameters between two point patterns.* TPAMI, 13(4):pp. 376–380.

Valgaerts, L., Bruhn, A., Mainberger, M., Weickert, J. (2011). *Dense versus Sparse Approaches for Estimating the Fundamental Matrix.* IJCV, 96(2):pp. 212–234.

Van Gool, L., Moons, T., Pauwels, E., Oosterlinck, A. (1995). *Vision and Lie's approach to invariance.* Image and Vision Computing, 13(4):pp. 259–277.

Vanden Wyngaerd, J., Van Gool, L. (2002). *Automatic Crude Patch Registration: Toward Automatic 3D Model Building.* Computer Vision and Image Understanding, 87(1-3):pp. 8–26.

Vasconcelos, F., Barreto, J.P., Nunes, U. (2012). *A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder.* TPAMI, 34(11):pp. 2097–2107.

Veksler, O. (2005). *Stereo correspondence by dynamic programming on a tree.* In *CVPR*, pp. 384–390.

Walker, M.W., Shao, L., Volz, R.a. (1991). *Estimating 3-D location parameters using dual number quaternions.* CVGIP: Image Understanding, 54(3):pp. 358–367.

Wang, H., Chin, T.J., Suter, D. (2012). *Simultaneously fitting and segmenting multiple-structure data with outliers.* TPAMI, 34(6):pp. 1177–1192.

Wang, Z.F., Zheng, Z.G. (2008). *A region based stereo matching algorithm using cooperative optimization.* In *CVPR*, pp. 1–8.

Wilson, K., Snavely, N. (2014). *Robust Global Translations with 1DSfM.* In *ECCV*, pp. 61–75.

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y. (2009). *Robust face recognition via sparse representation.* TPAMI, 31(2):pp. 210–27.

Wu, C. (2013). *Towards linear-time incremental structure from motion.* In *3DV*, pp. 127–134.

Wu, C. (2015). *P3 . 5P : Pose Estimation with Unknown Focal Length.* In *CVPR*.

Wu, C., Clipp, B., Li, X., Frahm, J. (2008). *3d model matching with viewpoint-invariant patches (vip).* In *CVPR*, pp. 1–8.

Wyngaerd, J., van Gool, L. (2003). *Combining texture and shape for automatic crude patch registration.* In *3Dim*, pp. 179–186.

Xu, L., Oja, E., Kultanen, P. (1990). *A new curve detection method: Randomized Hough transform (RHT).* Pattern Recognition Letters, 11(5):pp. 331–338.

Yamany, S.M., Farag, A.a. (2002). *Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching.* TPAMI, 24(8):pp. 1105–1120.

Yang, R., Pollefeys, M. (2003). *Multi-resolution real-time stereo on commodity graphics hardware.* In *CVPR*, pp. 1–7.

Zach, C. (2014). *Robust Bundle Adjustment Revisited.* In *ECCV*, pp. 772–787.

Zach, C., Irschara, A., Bischof, H. (2008). *What can missing correspondences tell us about 3D structure and motion?* In *CVPR*, pp. 1–8.

Zach, C., Klopschitz, M., Pollefeys, M. (2010). *Disambiguating visual relations using loop constraints.* In *CVPR*, pp. 1426–1433.

Zach, C., Niethammer, M., Frahm, J.M. (2009). *Continuous maximal flows and wulff shapes: Application to MRFs.* In *CVPR*, pp. 1911–1918.

Zach, C., Pock, T., Bischof, H. (2007). *A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration.* In *ICCV*, pp. 1–8.

Zamir, A.R., Shah, M. (2014). *Image Geo-localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs.* TPAMI, 36(8):pp. 1–1.

Zeisl, B., Georgel, P., Schweiger, F., Steinbach, E., Navab, N. (2009). *Estimation of Location Uncertainty for Scale Invariant Feature Points.* In *BMVC*, pp. 1–12.

Zeisl, B., Zach, C., Pollefeys, M. (2014). *Variational Regularization and Fusion of Surface Normal Maps.* In *3DV*, pp. 601–608.

Zhang, C., Zhang, Z. (2011). *Calibration between depth and color sensors for commodity depth cameras.* In *International Conference on Multimedia and Expo (ICME)*, pp. 47–64.

Zhang, W., Kosecka, J. (2002). *Efficient computation of vanishing points.* In *ICRA*, volume 1, pp. 223–228.

Zhang, Z. (1994). *Iterative point matching for registration of free form curves and surfaces.* IJCV, 12(2):pp. 119–152.

Zhang, Z. (1997). *Parameter estimation techniques: a tutorial with application to conic fitting.* Image and Vision Computing, 15(1):pp. 59–76.

Zhao, W., Nister, D., Hsu, S. (2005). *Alignment of continuous video onto 3D point clouds.* PAMI, 27(8):pp. 1305–18.

Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K., Okutomi, M. (2013). *Revisiting the PnP problem: A fast, general and optimal solution.* In *ICCV*, pp. 2344–2351.

Zhou, Q., Koltun, V. (2014). *Simultaneous Localization and Calibration: Self-Calibration of Consumer Depth Cameras.* In *CVPR*, pp. 454–460.

Zhou, Q.y., Miller, S. (2013). *Elastic Fragments for Dense Scene Reconstruction.* In *ICCV*, pp. 473–480.