Constrained Camera Motion Estimation and 3D Reconstruction

Vorgelegt von
Christian Kurz

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Juni 2014

Dekan – Dean

Prof. Dr. Markus Bläser	Universität des Saarlandes	Saarbrücken

Berichterstattende – Reporters

Prof. Dr. Hans-Peter Seidel	Universität des Saarlandes	Saarbrücken
Prof. Dr. Thorsten Thormählen	Philipps-Universität Marburg	Marburg
Prof. Dr. Bodo Rosenhahn	Leibniz Universität Hannover	Hannover

Kolloquium – Colloquium

Datum – Date

28. November 2014

Vorsitzender des Prüfungsausschusses – Chairman of the Examination Board

Prof. Dr. Philipp Slusallek	Universität des Saarlandes	Saarbrücken

Berichterstattende – Reporters

Prof. Dr. Hans-Peter Seidel	Universität des Saarlandes	Saarbrücken
Prof. Dr. Thorsten Thormählen	Philipps-Universität Marburg	Marburg

Akademischer Mitarbeiter – Faculty member

Dr. Christian Schulz	Universität des Saarlandes	Saarbrücken

Abstract

The creation of virtual content from visual data is a tedious task which requires a high amount of skill and expertise. Although the majority of consumers is in possession of multiple imaging devices that would enable them to perform this task in principle, the processing techniques and tools are still intended for the use by trained experts. As more and more capable hardware becomes available, there is a growing need among consumers and professionals alike for new flexible and reliable tools that reduce the amount of time and effort required to create high-quality content.

This thesis describes advances of the state of the art in three areas of computer vision: camera motion estimation, probabilistic 3D reconstruction, and template fitting.

First, a new camera model geared towards stereoscopic input data is introduced, which is subsequently developed into a generalized framework for constrained camera motion estimation. A probabilistic reconstruction method for 3D line segments is then described, which takes global connectivity constraints into account. Finally, a new framework for symmetry-aware template fitting is presented, which allows the creation of high-quality models from low-quality input 3D scans.

Evaluations with a broad range of challenging synthetic and real-world data sets demonstrate that the new constrained camera motion estimation methods provide improved accuracy and flexibility, and that the new constrained 3D reconstruction methods improve the current state of the art.

Kurzfassung

Die Erzeugung virtueller Inhalte aus visuellem Datenmaterial ist langwierig und erfordert viel Geschick und Sachkenntnis. Obwohl der Großteil der Konsumenten mehrere Bildgebungsgeräte besitzt, die es ihm im Prinzip erlauben würden, dies durchzuführen, sind die Techniken und Werkzeuge noch immer für den Einsatz durch ausgebildete Fachleute gedacht. Da immer leistungsfähigere Hardware zur Verfügung steht, gibt es sowohl bei Konsumenten als auch bei Fachleuten eine wachsende Nachfrage nach neuen flexiblen und verlässlichen Werkzeugen, die die Erzeugung von qualitativ hochwertigen Inhalten vereinfachen.

In der vorliegenden Arbeit werden Erweiterungen des Stands der Technik in den folgenden drei Bereichen der Bildverarbeitung beschrieben: Kamerabewegungsschätzung, wahrscheinlichkeitstheoretische 3D-Rekonstruktion und Template-Fitting.

Zuerst wird ein neues Kameramodell vorgestellt, das für die Verarbeitung von stereoskopischen Eingabedaten ausgelegt ist. Dieses Modell wird in der Folge in eine generalisierte Methode zur Kamerabewegungsschätzung unter Nebenbedingungen erweitert. Anschließend wird ein wahrscheinlichkeitstheoretisches Verfahren zur Rekonstruktion von 3D-Liniensegmenten beschrieben, das globale Verbindungen als Nebenbedingungen berücksichtigt. Schließlich wird eine neue Methode zum Fitting eines Template-Modells präsentiert, bei der die Berücksichtigung der Symmetriestruktur des Templates die Erzeugung von Modellen hoher Qualität aus 3D-Eingabedaten niedriger Qualität erlaubt.

Evaluierungen mit einem breiten Spektrum an anspruchsvollen synthetischen und realen Datensätzen zeigen, dass die neuen Methoden zur Kamerabewegungsschätzung unter Nebenbedingungen höhere Genauigkeit und mehr Flexibilität ermöglichen, und dass die neuen Methoden zur 3D-Rekonstruktion unter Nebenbedingungen den Stand der Technik erweitern.

Acknowledgments

My thanks go to Prof. Dr. Hans-Peter Seidel, who has given me the opportunity to join the MPI Informatik and work in a very enjoyable and productive environment, which will not be easy to find again.

My thanks go to Prof. Dr. Thorsten Thormählen and Dr. Michael Wand, my direct supervisors, who have always been encouraging and willing to personally supplement my mathematical education.

My thanks go to Dr. Nils Hasler, Dr. Arjun Jain, Dr. Tobias Ritschel, and Xiaokun Wu, my fellow coauthors, who have substantially contributed to many of the projects that have made this thesis possible.

My thanks go to Sabine Budde and Ellen Fries, the secretaries in our group, whom I have constantly bugged with questions and requests, and to my fellow researchers, who have provided comments and suggestions on many occasions.

My thanks go to Dr. Bertram Somieski, who has been liberal in sharing his experiences and offering advice, in addition to instigating many interesting discussions over lunch.

My thanks go to Martin Grochulla, who has been a good friend and office mate over the past years, and who has always been an excellent, patient discussion partner and willing to offer tactical support.

My thanks go to Thomas Grün, who has suggested valuable corrections.

My thanks go to my parents and to my brother, to Carola, to Severine, and to Achim, who have been there for me and provided me with support, care, and constructive remarks.

Contents

CHAPTER 1

Introduction

Camera motion estimation and 3D reconstruction has been a topic of continuing interest for computer vision and related fields. Its applications today are numerous, ranging from visual effects and augmented reality over video stabilization to camera motion style transfer or 3D scanning. Visual effects relying on *computer generated imagery* form an integral part of movie post-production today. The 3D reconstruction work, which is essential to augment the real footage with virtual objects and characters, is performed by trained and experienced professionals.

But it is not only the professionals and semi-professionals nowadays that are in the focus of the industry. In the wake of modern consumer devices, computer vision has quite literally entered the living room, and the industry strives to make professional tools available to the masses. With the introduction of the *Kinect*®, for instance, Microsoft® has created a hands-free input device for console gaming that transforms into a hand-held 3D scanner simply by plugging it into a PC with capable graphics hardware. Moreover, cell phones – and therefore cameras – are inexpensive and ubiquitous nowadays. Factoring in other sources, DSLRs and camcorders, the amount of image and video data created daily is immense.

The industry has started to put equipment considered professional just a couple of years ago into the hands of the consumer. But even with the right equipment, content creation, especially for high-quality content, still is a tedious, time-consuming process. The consumer usually has neither the time nor the expertise of a trained, paid professional. When transitioning towards the consumer market, the tools and algorithms employed are thus required to become more reliable, and they have to facilitate complicated processes for the use by inexperienced users. The goal is to achieve a high degree of automation, which significantly enlarges the potential user base.

To achieve the goal of creating reliable, user-friendly tools with a high degree of automation, flexible and generalized algorithms are an important component. When using such algorithms, less effort may have to be put into software maintenance and extension, potentially rendering the complicated and expensive process of software development more tractable and rewarding.

This thesis takes a look at several aspects of the camera motion estimation and 3D reconstruction problem. Man-made environments and objects offer a high degree of regularity and structural features, such as planar regions, straight line segments, and specific symmetry relations, which are powerful constraints not recognized by traditional methods. Using constrained estimation and optimization techniques, contributions are made in traditional camera motion estimation, probabilistic 3D reconstruction, and geometry processing. All these contributions are geared towards creating more accurate, reliable, generalized, and easily applicable algorithms, which may facilitate the transition of traditional tools into more consumer-oriented settings.

1.1 Structure and Overview

Visual orientation is not a difficult task for humans and animals. Relying on vision alone, the average human being is able to quickly assess situations, identify dangers, and generally traverse its environment safely at high speed. From a computer vision perspective, this is a tremendous feat. Robot and autonomous vehicle navigation would greatly benefit from visual capabilities on par with their biological counterparts, as well as many other areas of computer vision.

Chapter 2 – Structure from Motion Camera motion estimation and 3D reconstruction aims to provide various areas of application in computer vision with localization abilities akin to those of the human visual system. The relevant techniques are commonly subsumed under the term *structure from motion*, a basic introduction to which is given in Chapter 2.

After the introductory chapter, this thesis is divided into two parts. The first part focuses on constrained camera motion estimation. Camera motion estimation denotes joint estimation of camera position and orientation as well as the structure of the scene. The second part focuses on constrained 3D reconstruction, without conjoint optimization of the camera motion.

Part I – Constrained Camera Motion Estimation

Camera motion estimation yields information about the motion path of a virtual camera with respect to a representation of the real scene, based on the respective input data. The topic of the first part of this thesis is constrained camera motion estimation. Additional constraints arising from stereoscopic image sequences or specific configurations of the reconstructed scene representation are applied in the reconstruction process to achieve higher reconstruction quality and accuracy.

Chapter 3 – Bundle Adjustment for Stereoscopic 3D Stereoscopic image sequences have a long history in computer vision, primarily in autonomous vehicle and robot navigation. The resurgence of stereoscopic 3D technology in the movie industry has renewed the interest in this type of input data for camera motion estimation and 3D reconstruction. In contrast to the task mentioned before, where real-time performance is often a requirement, 3D reconstruction in movie post-production has less restrictions on computation time. Owing to the fact that the traditional tools and pipelines have largely been optimized for monocular data material, new algorithms have to be developed to process and exploit the inherent properties of stereoscopic input data. The most fundamental aspect of those properties is that the cameras in a stereo setup for the generation of stereoscopic 3D footage undergo only dependent motion. Traditional tools are often able to process stereoscopic data by treating it as monocular, but the specific input data characteristics are ignored.

Chapter 3 describes a new camera model for bundle adjustment which is suitable for stereoscopic input data. Stereo setups are explicitly modeled via a base frame and offset transformations, thus respecting the fact that both cameras may not move independent from one another. By reducing the number of spurious degrees of freedom, higher reconstruction quality is achieved.

Chapter 4 – A Generalized Framework for Constrained Bundle Adjustment Constraints have been used to make 3D reconstructions more accurate and reliable for quite some time. Especially for man-made environments, common observations are that reconstructed points are lying in the same plane, or that the walls of a building usually meet at a right angle. There are almost as many areas of application as there are different approaches to enforce the observed constraints, and to be efficiently applicable to several of these different applications, a framework has to allow the simultaneous application of camera and scene constraints with ease and flexibility.

Based on the general stereo camera model introduced in Chapter 3, this chapter develops a generalized framework for constrained bundle adjustment based on hierarchies of Euclidean transformations. This paradigm fulfills the requirements of being easy to apply and flexible enough to model many constraints, such as collinearity, coplanarity, parallelism, and angular relations. Stereo and multi-camera setups may be modeled easily. As will be shown Chapter 4, the hierarchies of Euclidean transformations also provide native support for rigidly moving objects.

Part II – Constrained 3D Reconstruction

The second part of this thesis focuses on constrained 3D reconstruction, with the position and orientation of the camera being either known or no longer required for further processing.

Chapter 5 – Global Connectivity Constraints for 3D Line Segment Reconstruction Straight lines are a predominant feature of man-made objects and environments. Given their prevalence, they are often used for 3D reconstruction. In contrast to point features, which are easy to detect and match, line features are more difficult to handle. The spatial extent of a line is not easily inferred from its image projection, and occlusions may lead to spurious detections and complicate the matching process across images, even if cues about the image geometry are available. In addition, reasoning about the connectivity in a set of disjoint 3D line segments is a complicated matter.

In Chapter 5, a new, probabilistic formulation of the common 3D line reconstruction problem is introduced. It enables joint estimation of line depth and connectivity, as well as line grouping and outlier elimination across frames.

Chapter 6 – Symmetry-aware Template Deformation and Fitting Dense object reconstructions based on structure from motion or other optical reconstruction procedures, such as active 3D scanning, are one possibility to obtain a digital 3D model of a target object. The other possible courses of action are to create the model by hand, or to search for a pre-existing model on the internet.

For the manual creation of a model, the limiting factors are time and expertise. 3D modeling software is usually easy to procure, but the average consumer is inept at using it. Significant amounts of time are required in order to learn the 3D modeling process in addition to the time a professional would need to create the model. The manual creation of a 3D model is not a viable option in many cases.

Using a structure-from-motion-based approach or a 3D scanner to create the dense model entails its own host of problems. The reconstructions contain noise, outliers, and holes in the reconstructed models, and even professional scanning equipment encounters difficulties with certain (reflective) object surfaces. Again, performing the reconstruction procedures and operating the scanning equipment requires a certain level of expertise, especially if some sort of post-processing has to be performed.

Considering the required effort, using a pre-existing model is usually the most economic decision. The internet is littered with 3D models of every level of quality and complexity for any conceivable topic. Whole collections of models are available for purchase. Given the high amount of different models per object class contained in comprehensive shape libraries and similar resources, it is probable that the average user will be able to locate a model similar to the object he/she is trying to create a model of. These pre-existing models may be modified by hand to better approximate the target object, but this approach has the potential to be equally time consuming as modeling from scratch.

The approach presented in Chapter 6 combines the usage of a pre-existing template model with a scan of the actual object to automatically deform the template in a way that makes it closely resemble the scan. In combination with a structural analysis of the template, this enables custom models with high quality to be created in an easy and convenient way.

Chapter 7 – Conclusion A closing discussion and outlook concludes this thesis.

1.2 Contributions

In this thesis, the following contributions are made:

- A flexible model for stereoscopic camera setups for optimization in bundle adjustment is developed. The model is able to accommodate many different stereo configurations with different characteristics and provides improved reconstruction speed and accuracy (Chapter 3, published as Kurz et al. [89]).
- A generalized framework for constrained bundle adjustment based on hierarchies of Euclidean transformations is proposed, based on the aforementioned model for stereoscopic camera setups. This framework provides an elegant and intuitive way to handle many camera and scene constraints and moving objects (Chapter 4).
- A probabilistic formulation for the 3D reconstruction of straight lines from multiple images is introduced. The formulation includes the depth configuration of the individual line segments and also models connectivity. In addition, it permits line grouping and outlier elimination. Line segment depth and connectivity are optimized conjointly (Chapter 5, published as Jain et al. [76]).
- A constrained template deformation and fitting approach is presented. This approach allows the deformation of a template model to fit a given target scan while preserving the detected symmetry structure of the template. As a result, high-quality custom models that closely fit the desired target geometry may be obtained with ease (Chapter 6, published as Kurz et al. [90]).

1.3 List of Publications

- C. Kurz, T. Thormählen, and H.-P. Seidel. Bundle Adjustment for Stereoscopic 3D. In A. Gagalowicz and W. Philips, editors, *Computer Vision/Computer Graphics Collaboration Techniques*, volume 6930 of *Lecture Notes in Computer Science*, pages 1–12. Springer, October 2011. Cited as Kurz et al. [89].
- A. Jain, C. Kurz, T. Thormählen, and H.-P. Seidel. Exploiting Global Connectivity Constraints for Reconstruction of 3D Line Segments from Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2010, pages 1586–1593, San Francisco, CA, USA, June 2010. IEEE. Cited as Jain et al. [76].
- C. Kurz, X. Wu, M. Wand, T. Thormählen, P. Kohli and H.-P. Seidel. Symmetry-aware Template Deformation and Fitting. *Computer Graphics Forum Early View (Online Version of Record published before inclusion in an issue)*, John Wiley & Sons, March 2014. Cited as Kurz et al. [90].

Additional publications

- C. Kurz, T. Thormählen, and H.-P. Seidel. Scene-Aware Video Stabilization by Visual Fixation. In *Proceedings of the 6th European Conference for Visual Media Production*, CVMP 2009, pages 1–6, London, UK, November 2009. IEEE.
- C. Kurz, T. Thormählen, B. Rosenhahn, and H.-P. Seidel. Exploiting Mutual Camera Visibility in Multi-Camera Motion Estimation. In *Advances in Visual Computing*, volume 5875 of *Lecture Notes in Computer Science*, pages 391–402. Springer, November 2009.
- C. Kurz, T. Ritschel, E. Eisemann, T. Thormählen, and H.-P. Seidel. Camera Motion Style Transfer. In *Proceedings of the 7th European Conference on Visual Media Production*, CVMP 2010, pages 9–16, London, UK, November 2010. IEEE.
- C. Kurz, T. Thormählen, and H.-P. Seidel. Visual Fixation for 3D Video Stabilization. *Journal of Virtual Reality and Broadcasting* (CVMP 2009 Special Issue), 8(2):1–12, January 2011.
- C. Kurz, T. Ritschel, E. Eisemann, T. Thormählen, and H.-P. Seidel. Generating Realistic Camera Shake for Virtual Scenes. *Journal of Virtual Reality and Broadcasting*, 10(7):1–13, January 2014.

CHAPTER 2

Structure from Motion

Structure from motion (SfM) is a computer vision approach for 3D reconstruction from multiple images. It is based on the establishment of 2D feature point correspondences between the images, which allow the triangulation of their common 3D object point.

This chapter provides an introductory overview based on the seminal book *Multiple View Geometry* by Hartley and Zisserman [62] and the work of Thormählen [147].

2.1 Introduction and Outline

SfM, which is also commonly referred to as *structure and motion (SaM)*, is a triangulation-based approach for 3D reconstruction from visual data, e. g., image sequences. It has its roots in photogrammetry, and has received considerable attention in computer vision and related fields over the past decades.

Summary of the Approach The SfM pipeline consists of several steps. First, correspondences are established: 2D feature points are detected in an image and then either tracked to subsequent images (in case of an image sequence) or matched to the other images. The next step is the elimination of outlier feature point tracks or matches using geometric constraints between two or more images. This is followed by the computation of initial values for the camera parameters (position, orientation, and intrinsic parameters like the focal length) and the 3D object points. The initial values for the camera parameters and 3D object points are subsequently optimized by a

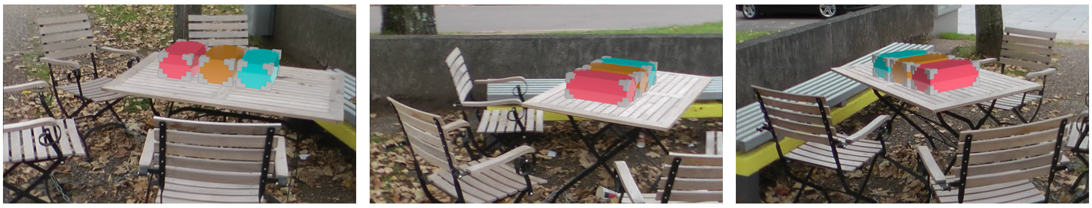


Figure 2.1: Three images of a real image sequence consistently augmented with virtual objects. The virtual camera path and scene geometry corresponding to the real image sequence were reconstructed with SfM. This information was then used to place three virtual boxes in the virtual scene. Finally, renderings of the virtual scene were generated and composited with the real images. Due to the accurate estimation of the camera position and orientation, the virtual objects are consistently positioned on the real table.

maximum-likelihood estimation, which is called *bundle adjustment*. This optimization procedure is a vital part of SfM. The algorithm is typically applied in an incremental fashion, starting from a reconstruction of only two images, which is then extended image by image.

Areas of Application The areas of application of SfM include camera motion recovery in cinematography, where it is called *match moving* or *camera solving*, and augmented reality. The goal in these applications is to determine the unknown motion path of the camera from the visual input data. This information is then used to accurately create a *virtual* camera at the correct position for each image to allow the augmentation of the input data by consistently placed and correctly scaled virtual objects (see Figure 2.1 for an example). The movie industry has numerous predominantly commercial tools at its disposal to accomplish this task, such as PFTrack™, SynthEyes™, or 3DEqualizer™, just to name a few. The process for offline video-based augmented reality is basically SfM, as described by Gibson et al. [55]. For online (i. e., real-time) video-based augmented reality, the computational complexity of SfM is still a challenge.

SfM is also closely related to *Visual Simultaneous Location and Mapping (Visual SLAM)*. SLAM is employed by autonomous robots and vehicles to determine their position in the environment by incrementally creating a map thereof. The term *visual* SLAM was coined by Karlsson et al. [81], who describe the inclusion of camera data into the SLAM process. Real time constraints usually require the location and mapping tasks to be performed concurrently. There is a tight connection between SfM and the (in comparison to the location process) more time consuming mapping process.

Another area of application of SfM is object reconstruction, geared towards creating a virtual representation of an object or environment. The models created may then be used in a variety of ways. Additional steps have to be taken, as traditional SfM yields only sparse, point-based reconstructions, which are not directly suited for such applications. Approaches in this area are also referred to as *multi-view stereo*. The work by Goesele et al. [57], which describes dense reconstruction from community photo collections based on region-growing, is well known. Furukawa and Ponce [49] have introduced a patch-based dense SfM approach that allows the recovery of high-quality models, which is now widely used.

Outline The following sections first introduce the scene model and the camera model used in SfM (Section 2.2 and Section 2.3, respectively). Section 2.4 then describes how an initial reconstruction can be obtained from two images, and Section 2.5 shows how this initial reconstruction can be extended to comprise additional images. Section 2.6 discusses *auto-calibration*, the process of upgrading 3D reconstructions from a projective to a metric frame. The chapter is concluded by a discussion of bundle adjustment in Section 2.7.

2.2 The Scene Model

SfM is based on the relations between the 3D structure of a scene and the 2D image of this scene created on the image plane of a camera. A convenient representation to model the scene is based on 2D and 3D points. Other representations such as lines (see Bartoli and Sturm [11]) are possible, but points will be the sole representation considered in this chapter.

The input data to SfM consists of a set of J images I_j, $j = 1, \dots, J$. The method of acquisition of the images I_j is inconsequential for the scene model. During the reconstruction process, a number of 2D feature points $\mathbf{x}_{j,k}$ is established. The first index j denotes the image the feature point has been detected in. The second index k relates the feature points across images – it establishes correspondences. For an image sequence, the index k identifies the *feature track* a feature point belongs to. In any way, the index $k = 1, \dots, K$ attributes a 3D object point $\mathbf{X}_k$ from the set of K object points to every feature point. This scene model is illustrated in Figure 2.2. To give an example, a 2D feature point $\mathbf{x}_{4,2}$ represents the detected 2D position of 3D object point $\mathbf{X}_2$ in image I_4. Note that in a typical SfM setting usually only a small subset of all object points is visible in every image. In addition, the object points only describe a sparse representation of the scene. This is a consequence of the formulation based on feature points: for traditional SfM settings, feature points typically cannot be established densely with any reliability, which leaves only a sparse distribution over the images.

Rigidity SfM usually makes an implicit assumption about the rigidity of the scene, since the correspondences established for the image points relate all corresponding

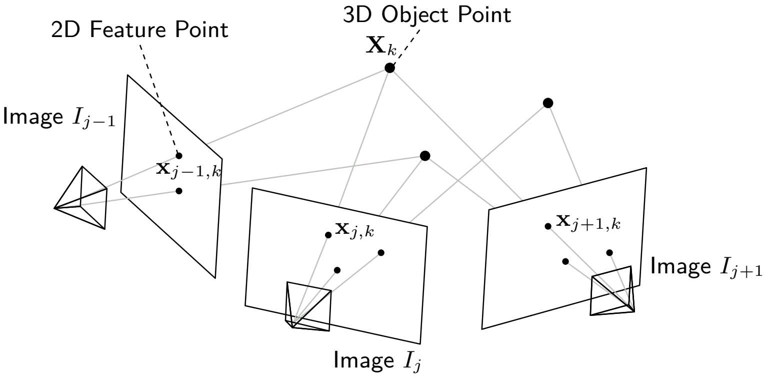


Figure 2.2: The basic principle of SfM is the relation between the corresponding 2D feature points $\mathbf{x}_{j,k}$ of images I_j and the reconstructed 3D object points $\mathbf{X}_k$.

2D points to a single 3D point. Image points corresponding to moving and deforming objects are eliminated in the process. SfM can be extended to handle independent rigid moving objects (see the work by Fitzgibbon and Zisserman [44], for example). The research to address non-rigid objects in SfM has recently culminated in the work by Garg et al. [53], who have introduced a variational formulation for the dense reconstruction of non-rigid objects from video. Although scene rigidity is assumed for the remainder of this thesis, the topic will be revisited in Chapter 4.

2.3 The Pinhole Camera Model

The pinhole camera model describes the projection of points in 3D space onto the 2D image plane of an imaging device. Imaging devices will simply be denoted as cameras henceforth. Assuming the camera center to coincide with the origin and the camera pointing in negative Z-direction, projection of a 3D point $(X, Y, Z)^\top$ to a 2D point $(x, y)^\top$ on the canonical image plane ($Z = 1$) is achieved by perspective division, division by the Z-coordinate:

$$x = \frac{X}{Z} \quad \text{and} \quad y = \frac{Y}{Z} \quad . \tag{2.1}$$

This process is depicted in Figure 2.3, left. Using these relations, the image on the image plane is inverted, as it would be with a real pinhole camera. It is common practice to counter this inconvenience by the introduction of a *virtual* image plane. To project to the virtual image plane at $Z = -1$ and produce an upright image, it is sufficient to divide by $-Z$ instead of Z. This is also illustrated in Figure 2.3.

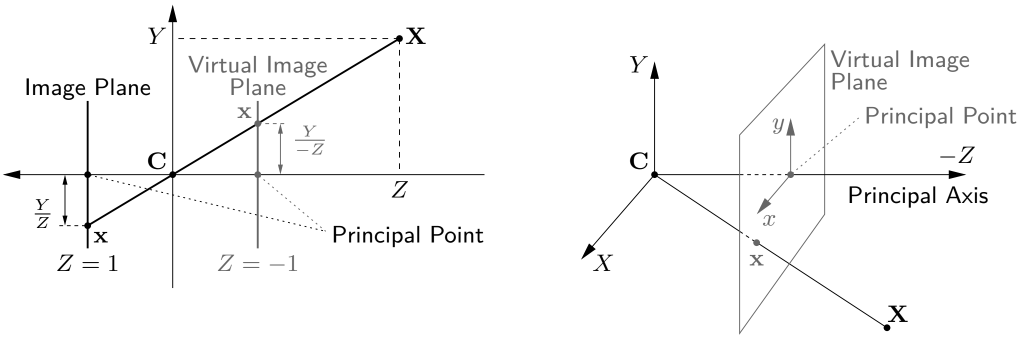


Figure 2.3: The projection of a 3D point $\mathbf{X}$ onto the canonical image plane (located at $Z = 1$) and onto the virtual image plane (located at $Z = -1$) yields a 2D point $\mathbf{x}$ in each case. The image produced on the canonical image plane is inverted, and the one produced on the virtual image plane is upright. The center of projection, which coincides with the origin, is denoted as $\mathbf{C}$.

Linear Mapping The projection of the pinhole camera may be expressed as a linear mapping:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & -1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad , \tag{2.2}$$

where $\simeq$ denotes equality up to scale and the 2D and 3D points are represented in homogeneous coordinates.

The Finite Projective Camera To accurately approximate real cameras with this model, additional parameters have to be taken into account. Substituting $\mathbf{x} = (x, y, 1)^\top$ and $\mathbf{X} = (X, Y, Z, 1)^\top$, the full expression for the *finite projective camera* is

$$\mathbf{x} \simeq \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}^{-1}\mathbf{X} \quad , \tag{2.3}$$

with $\mathbf{x} \in \mathbb{P}^2$ and $\mathbf{X} \in \mathbb{P}^3$ being instances of the 2D feature points and 3D object points introduced in Section 2.2, represented in homogeneous coordinates in projective space. The projection now takes into account the *intrinsic* camera parameters (like the focal length) via the 3×3 camera calibration matrix $\mathtt{K}$ and the *extrinsic* camera parameters (position and orientation) via the 4×4 transformation matrix $\mathtt{T}$. These parameters will be discussed in Section 2.3.1 and Section 2.3.2 respectively. The matrix $[\mathtt{I}|\mathbf{0}]$ composed of the 3×3 identity matrix $\mathtt{I}$ and a null vector $\mathbf{0}$ ensures the proper matrix dimensions.

The Camera Matrix Equation (2.3) suggests that one may replace the individual matrices by a single matrix $\mathtt{P}$:

$$\mathtt{P} = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}^{-1} \quad . \tag{2.4}$$

The homogeneous 3×4 *camera matrix* $\mathtt{P}$ has 11 degrees of freedom, as it is only defined up to an arbitrary scale. To represent a finite camera, it has to fulfill the constraint that its left hand 3×3 submatrix has to be non-singular; otherwise it represents a camera at infinity. If the camera matrix may take the latter form, it represents a *general projective camera*. In either case, projection is simply given by

$$\mathbf{x} \simeq \mathtt{P}\mathbf{X} \quad . \tag{2.5}$$

Distortion Lens distortion is another important aspect to consider in the camera model. It will be treated in Section 2.3.3.

2.3.1 The Intrinsic Camera Parameters

The transfer of the image from the canonical image plane to the desired virtual one is governed by the intrinsic camera parameters focal length (f), skew (s), pixel aspect ratio (η), and principal point offset (o_x, o_y). Together these parameters form the calibration matrix $\mathtt{K}$:

$$\mathtt{K} = \begin{bmatrix} f & s & o_x \\ & \eta f & o_y \\ & & -1 \end{bmatrix} \quad . \tag{2.6}$$

The value -1 in the last row accounts for the projection onto the virtual image plane at $Z = -1$ (as described at the beginning of Section 2.3).

The Focal Length Parameter f The focal length parameter f encodes the distance between the camera center and the image plane. As such it functions as a scaling factor for the transfer of the projected image positions from the image plane at unit distance (where they end up after perspective division) to the desired one.

The Skew Parameter s The skew parameter s models a skewing of the camera coordinate system, i. e., that the x- and y-axes of the coordinate system are not perpendicular to each other. For a digital camera, this would require the image sensor to be manufactured this way. In practice, the default assumption of $s = 0$ should therefore hold, the exception being an image taken from an image (see Hartley and Zisserman [62]). Neither modification nor optimization of this parameter should be considered in a general setting. The skew parameter is proportional to the focal length f, and therefore is not meaningful by itself.

The Pixel Aspect Ratio Parameter η The *pixel aspect ratio (PAR)* parameter η accounts for differences in the pixel pitch[1] in x- and y-direction. For square pixels, the default PAR value is $\eta = 1$. The differences in the pixel pitch may be the result of actual non-square pixels in the image sensors of the capturing hardware, but it is also possible that the format for data storage specifies a certain PAR. For example, many camcorders would record high-definition video in HDV™ format, which specifies a resolution of 1440 × 1080 with a PAR of $1.\overline{3}$ ($\eta = 0.75$). This may occur even if the sensor is able to record material at a higher resolution in order to reduce the storage space required.

If the PAR of the input data is known, the material can be scaled in a way that makes the effective PAR equal to 1. This provides more control than having to deal with parameters $f_x = f$ and $f_y = \eta f$, where the latter implicitly contains the PAR. The introduction of η has several advantages, though: It also allows the use of a single focal length parameter. Furthermore, the input data does not need to be scaled if η is set correctly, thus avoiding artifacts resulting from interpolation. And finally, if there is no information on the PAR available or if the available information is suspected to be inaccurate, the separation of f and η allows greater control over the optimization procedure by making η invariant even for potentially varying f (for varying f, not making η invariant introduces a spurious degree of freedom per calibration matrix).

The Principal Point Offset o_x, o_y The components o_x and o_y of the principal point offset account for inaccuracies in the placement of the image sensor. They model a translation of the sensor in the image plane and describe the deviation of the actual point of intersection of the image plane with the optical axis (the *principal point*) as opposed to the idealized assumption that the optical axis intersects the image plane dead center, which is hard to achieve in practice.

2.3.2 The Extrinsic Camera Parameters

The extrinsic orientation of the camera – its position and viewing direction – can be represented by a 3 × 3 rotation matrix $\mathtt{R}$ that represents the viewing direction of the camera and a 3-dimensional vector $\mathbf{t}$ that represents the position of the camera – the camera's center of projection $\mathbf{C}$.

The matrix $\mathtt{R}$ and the vector $\mathbf{t}$ can be combined into a proper rigid transformation $\mathtt{T}$:

$$\mathtt{T} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad . \tag{2.7}$$

In this form, the transformation matrix $\mathtt{T}$ describes a transformation from a local coordinate system to the world coordinate system. To reflect the nature of the camera

[1] The pixel pitch is the distance between the centers of adjacent pixels (see Fauber [41]).

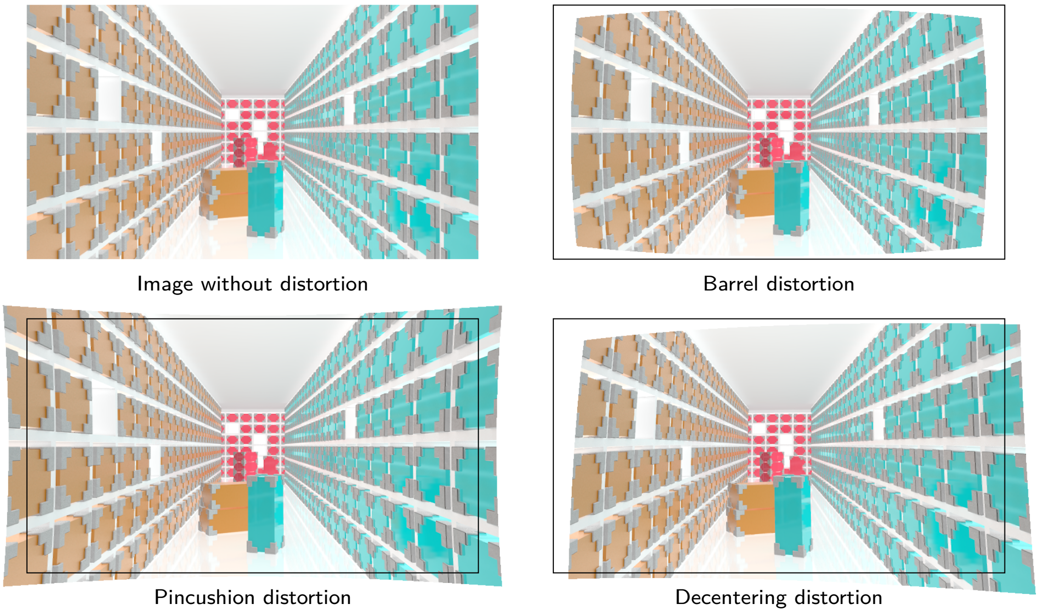


Image without distortion

Barrel distortion

Pincushion distortion

Decentering distortion

Figure 2.4: Illustration of the effect of distortion. The black border superimposed onto the distorted images indicates the shape of the original image for reference. Barrel and pincushion distortion are both types of radial distortion.

(which is to project points from the world coordinate system to the local, canonical camera coordinate system), one has to use the inverse of $\mathtt{T}$, as has already been indicated in Equation (2.3):

$$\mathtt{T}^{-1} = \begin{bmatrix} \mathtt{R}^\top & -\mathtt{R}^\top \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad . \tag{2.8}$$

2.3.3 Distortion

The pinhole camera model is an idealized abstraction of the real imaging hardware. Although conformity to the idealized model is desirable in most cases, manufacturers encounter many challenges in practice. Constraints on quality typically arise from size, weight, and not least cost, and render the production of *perfect* imaging hardware all but infeasible. The pinhole camera model is therefore usually extended by a *distortion model* to compensate some of the manufacturing imprecision in the lens and its placement. Commonly modeled types of distortion are radial distortion (barrel and pincushion distortion) and decentering distortion. Figure 2.4 illustrates the effect these types of distortion have on the projected image.

Lens distortion has already been a major concern back in the 1950s (a good summary is given by Brown [27]), and many distortion models in use today are still based on the same theoretical foundations. The distortion model to calculate the distorted position $(x_d, y_d)^\top$ of a point $(x, y)^\top$ on the image plane proposed then has a radial and a tangential component:

$$x_d = x + \delta_x + \Delta_x \quad \text{and} \quad y_d = y + \delta_y + \Delta_y \quad , \tag{2.9}$$

with radial distortion terms δ_x and δ_y and decentering distortion terms Δ_x and Δ_y.

The above formulation lets the center point of distortion coincide with the principal point. Brown [26] argues that this is a practical and effective compromise for imaging systems with comparatively short focal length.

Radial Distortion As summarized by Brown [27], the radial distortion δ_r of a perfectly centered lens can be expressed as an odd powered series of the form

$$\delta_r = k_0 r^3 + k_1 r^5 + k_2 r^7 + \cdots \quad , \quad \text{with} \quad r = \sqrt{(x^2 + y^2)} \tag{2.10}$$

for coordinates x and y expressed in a coordinate system that has the principal point as origin. The observation that the x- and y-components of the distortion can be expressed as

$$\delta_x = \frac{x}{r}\delta_r \quad \text{and} \quad \delta_y = \frac{y}{r}\delta_r \tag{2.11}$$

has given rise to the still widely used formulation

$$x_d = x(1 + k_0 r^2 + k_1 r^4 + k_2 r^6 + \cdots) \quad \text{and} \tag{2.12}$$

$$y_d = y(1 + k_0 r^2 + k_1 r^4 + k_2 r^6 + \cdots) \quad . \tag{2.13}$$

The effect of two common types of radial distortion, barrel distortion and pincushion distortion, is illustrated in Figure 2.4. Barrel distortion generally has negative values for the parameter series, while pincushion distortion has positive ones; mixed series of values are also possible, though.

Decentering Distortion The effects of decentering in the lens placement are tangential distortion and asymmetric radial distortion (illustrated in Figure 2.4), summarized as decentering distortion by Brown [26]. The decentering distortion Δ is characterized as

$$\Delta_x = \left[2p_0 xy + p_1\left(r^2 + 2x^2\right)\right]\left(1 + p_2 r^2 + p_3 r^4 + \cdots\right) \quad \text{and} \tag{2.14}$$

$$\Delta_y = \left[2p_1 xy + p_0\left(r^2 + 2y^2\right)\right]\left(1 + p_2 r^2 + p_3 r^4 + \cdots\right) \quad . \tag{2.15}$$

Distortion Model The distortion model considered in this thesis is that of the popular open source computer vision library OpenCV [23], which is a modified version of the model by Brown [26] given in Equation (2.9). All the higher order decentering distortion parameters (p_2 and onwards) are assumed to be zero, and the symmetric radial distortion is extended by a rational term apparently inspired by the work of Claus and Fitzgibbon [36] to better model wide-angle lenses:

$$\begin{aligned} x_d &= x\frac{1+k_0r^2+k_1r^4+k_2r^6}{1+k_3r^2+k_4r^4+k_5r^6}+2p_0xy+p_1\left(r^2+2x^2\right) && (2.16)\\ y_d &= y\frac{1+k_0r^2+k_1r^4+k_2r^6}{1+k_3r^2+k_4r^4+k_5r^6}+2p_1xy+p_0\left(r^2+2y^2\right) \quad . && (2.17)\end{aligned}$$

Application The effect of distortion can be applied with matrix expressions by using a *lifting* of the point coordinates, which would be consistent with the projection process of Section 2.3. This is beyond practicality for the higher-order distortion models described in the previous section, however. Distortion is therefore expressed by a mapping $\mathfrak{d}$:

$$\mathfrak{d} : \mathbb{P}^2 \mapsto \mathbb{P}^2 \quad , \quad \mathbf{x}_d = \mathfrak{d}(\mathbf{x}_u) \quad , \tag{2.18}$$

where the undistorted point in the image plane is given as $\mathbf{x}_u = (x, y, 1)^\top$, and the distorted point as $\mathbf{x}_d = (x_d, y_d, 1)^\top$. Equation (2.3) is then reformulated as

$$\mathbf{x} \simeq \mathtt{K}\,\mathfrak{d}\Big([\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}^{-1}\mathbf{X}\Big) \quad . \tag{2.19}$$

A drawback of this formulation is that requires perspective division to be performed before distortion is applied. If the 3D points are to be projected onto the virtual image plane at $Z = -1$, this has to be taken into account, and accordingly the lower right value of the calibration matrix $\mathtt{K}$ should contain the value 1 instead of -1 in this case.

2.4 Initialization

This section describes how an initial 3D reconstruction may be obtained from two images. The initial reconstruction consists of estimates for the camera matrices related to the images and estimates for the common 3D object points, and serves as a base for the extension to multiple images, as discussed in Section 2.5.

First, 2D feature points are detected and matched or tracked to generate correspondences, followed by outlier elimination. The fundamental matrix created as a by-product during outlier elimination then allows camera and structure recovery, followed by a maximum-likelihood-estimation, the bundle adjustment.

Key Frame Selection The approach described in this section assumes that the camera positions used to capture the two images are separated by an adequate translational offset. This is a requirement for the structure recovery, the triangulation of 3D object points, to work. To select suitable images (*key frames*) for reconstruction, an information criterion, such as the *geometric robust information criterion (GRIC)* by Torr [149], may be used.

2.4.1 Correspondences

Correspondences are of fundamental importance in SfM. A correspondence establishes a relation between specific points of two or more images: it signifies that the respective image positions are observations of the same 3D points of the scene. Correspondences may thus be used to triangulate 3D points from the observed image points, which is the very basis of SfM. This is only true if the scene is static, of course. If an object in the scene has moved between frames, the established correspondences may be correct, but since they are not observations of the same 3D points in space (as the observed object has moved), they cannot be processed by traditional SfM. In this case, the correspondences in questions are outliers and have to be treated accordingly, as discussed in Section 2.4.2. The notation $\mathbf{x} \leftrightarrow \mathbf{x}'$ will be used to denote a 2D feature point correspondence between two images I and I' in this context.

Finding corresponding image locations is commonly subsumed under the term *correspondence problem*. The problem may be solved automatically, semi-automatically, or completely manually. The latter two cases are more common in the industry, as they are potentially very accurate (which may be paramount) but also very time consuming.

Methods for the automatic creation of correspondences can generally be divided into one of two categories: feature point tracking and feature point matching. Feature points for matching or tracking may be detected by a variety of methods, such as the popular Harris corner detector by Harris and Stephens [61] and variants thereof, or *features from accelerated segment test (FAST)* by Rosten and Drummond [126], which has recently gained considerable traction.

Feature Point Tracking Feature point tracking describes methods for the creation of correspondences that rely on the displacements of corresponding points in the image plane to be small for pairs of subsequent images. It is therefore the method of choice when the change in camera position and orientation between exposures is small, as is the case for video data and certain image sequences in very controlled settings.

The traditional method for feature point tracking in SfM (aside from a window-based search using *normalized cross correlation* or similar) is KLT tracking, based on the publications by Lucas and Kanade [96] and Shi and Tomasi [135]. KLT is a gradient-based optimization approach that computes the motion-introduced displacement of a feature point starting from the position in the previous image as initial estimate.

The formulations and methods of KLT are similar to *optical flow*, although classical optical flow does not create consistent feature point tracks over longer sequences – while the result is typically dense in contrast to KLT. This has been addressed by Sand and Teller [131] with the introduction of *Particle Video (PV)* and by Brox and Malik [28] with *Large Displacement Optical Flow (LDOF)*, among others. The method by Rubinstein et al. [127] (which requires PV or LDOF as preprocessing) appears to yield the best tracking results so far. However, much like PV, it is not suitable for online processing, and therefore has areas of application different from KLT and LDOF.

Although LDOF is superior to KLT in terms of track density and accuracy, KLT is still a common and popular choice for feature point tracking. This may be attributed to the fact that, in addition to being conceptionally simpler, KLT has a potential processing rate in excess of 200 frames per second and is therefore at least two orders of magnitude faster in a GPGPU setting for image material of comparable resolution (the implementations in question being the one by Zach et al. [166] for GPU KLT and the one by Sundaram et al. [141] for GPU LDOF).

LDOF and similar algorithms in optical flow may be seen as combination approaches between feature point tracking and matching, as they make use of feature point matching to obtain longer trajectories.

Feature Point Matching In contrast to feature point tracking, feature point matching does not rely on small displacements in the feature positions in the input data. Instead it *matches* detected features across images by the application of a *feature descriptor*.

Since its introduction in 1999, *scale-invariant feature transform (SIFT)* by Lowe [95] was (and often still is) considered to be state-of-the-art in feature point matching. SIFT features are invariant to rotation and scale, desirable properties for matching input data with strong variation. Many contestants have been inspired by SIFT, such as the vastly more efficient *speeded up robust features (SURF)* by Bay et al. [12], *center surround extrema (CenSurE)* by Agrawal et al. [1], or lately KAZE[2] features by Alcantarilla et al. [4].

Binary descriptors, a new type of feature point descriptors introduced only recently, offer significant advantages in terms of processing time, while still being able to provide matching performance similar to SIFT in many cases (for a performance evaluation, see the work by Heinly et al. [66]). The first binary descriptor, *binary robust independent elementary features (BRIEF)*, was introduced by Calonder et al. [29], followed by several other works that show descriptors with different sampling patterns, sampling pairs, and improved matching characteristics: *oriented FAST and rotated BRIEF (ORB)* by Rublee et al. [128] (rotation invariant), *binary robust invariant scalable keypoints (BRISK)* by Leutenegger et al. [91] (rotation and scale invariant), and *fast retina keypoint (FREAK)* by Alahi et al. [3] (rotation and scale invariant).

[2] KAZE is not an abbreviation, but rather Japanese for *wind*.

2.4.2 Outlier Elimination

The set of correspondences created by feature point tracking or feature point matching typically contains a certain number of outliers. Feature points may be poorly located due to motion blur, drift over time, or just be plainly mismatched. But even if the tracking or matching is flawless, feature points may be considered as outliers if they violate any of the assumptions the model for scene description is based on. For rigid SfM, feature points that correspond to an object in motion relative to the static scene have to be considered as outliers.

Robust Estimation Outlier elimination requires the use of robust estimation techniques that are able to cope with these very outliers in the input data. To provide a reliable estimate, the *random sample consensus (RANSAC)* family of algorithms is a popular choice. The *m-estimator sample consensus (MSAC)* algorithm by Torr and Zisserman [150], a straightforward extension of the original algorithm by Fischler and Bolles [42], has improved performance at no additional implementation complexity or computational cost. It iteratively optimizes the cost function

$$R = \sum_i \rho\left(e_i^2\right) \quad , \quad \text{with} \quad \rho\left(e^2\right) = \begin{cases} e^2 & \text{if } e^2 < \tau^2 \\ \tau^2 & \text{if } e^2 \geq \tau^2 \end{cases} \quad , \tag{2.20}$$

where e_i is a per-datum error function and τ a threshold value that decides whether a datum is an outlier or not.

During each iteration, the algorithm proceeds by first selecting a number of β random samples from the input data. The model parameters are then estimated from the selected samples and the error function e_i is evaluated for each datum, which allows the computation of R. If the value of R is lower than any previously calculated cost, the current model is considered as the best solution so far.

The number of iterations is crucial, as the algorithm is non-deterministic: The higher the number of iterations, the higher the probability of a reasonable result, i. e., a result calculated from inlier samples only. If m is the desired probability of obtaining a reasonable result, the corresponding number of iterations required q is given by

$$q = \frac{\log(1 - m)}{\log(1 - n^\beta)} \quad , \tag{2.21}$$

where n is the probability that any selected data point is an inlier. The probability n is replaced by a very conservative estimate in practice, but it can be updated on the fly as soon as a model is found whose set of inliers comprises more data points than previously assumed as probable by the choice of n.

The number of random samples β selected per iteration has a major influence on the number of iterations required to get a reasonable result with a certain probability,

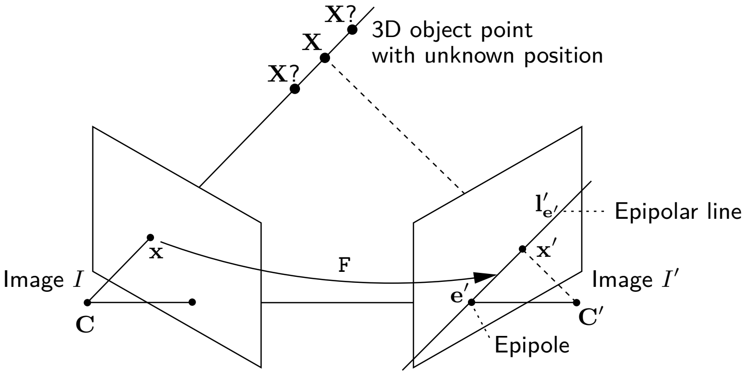


Figure 2.5: Illustration of the epipolar geometry relating a pair of images I and I'. The fundamental matrix $\mathtt{F}$ transforms a 2D point $\mathbf{x}$ onto the epipolar line $\mathbf{l}'_{\mathbf{e}'}$, which is the image projection of the line of sight from the center of projection $\mathbf{C}$ through $\mathbf{x}$, on which the 3D point $\mathbf{X}$ is located. The epipole $\mathbf{e}'$ is the projection of $\mathbf{C}$ into the image plane and forms the intersection of all epipolar lines.

as indicated by Equation (2.21). If possible, it is desirable to choose β as the minimum number of samples required for the estimation of the model in order to reduce the number of iterations required.

Fundamental Matrix Outlier Elimination The fundamental matrix $\mathtt{F}$ is an algebraic representation of the epipolar geometry of two images. An illustration of the epipolar geometry is provided in Figure 2.5. The fundamental matrix $\mathtt{F}$ describes the transfer of a point $\mathbf{x}$ in image I to the corresponding *epipolar line* $\mathbf{l}'_{\mathbf{e}'} = \mathtt{F}\mathbf{x}$ in image I'. For a point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ between the two images, there is the corresponding relation $\mathbf{l}_{\mathbf{e}} = \mathtt{F}^{\top}\mathbf{x}'$, which allows the fundamental matrix to be characterized by the equation

$$\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0 \tag{2.22}$$

for any point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$. By stacking expressions of the form

$$\left(x'x, x'y, x', y'x, y'y, y', x, y, 1\right)\mathbf{f} = 0, \tag{2.23}$$

where $\mathbf{f}$ is the 9-vector created from the entries of $\mathtt{F}$ in row-major order, a system of linear equations $\mathtt{A}\mathbf{f} = \mathbf{0}$ can be created. At least $\beta = 7$ point correspondences are required to determine the fundamental matrix up to scale. It may be recovered from the right null-space of $\mathtt{A}$ using *singular value decomposition (SVD)* in combination with the requirement that $\mathtt{F}$ has rank 2, i. e., $\det \mathtt{F} = 0$.

The per-datum error function e_i for robust estimation is given as

$$e_i = \mathrm{d}(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i)^2 + \mathrm{d}\Big(\mathbf{x}_i, \mathtt{F}^\top \mathbf{x}'_i\Big)^2 \quad , \tag{2.24}$$

where $\mathrm{d}(\mathbf{x}, \mathbf{l_e})$ is the distance of $\mathbf{x}$ to the epipolar line $\mathbf{l_e}$.

Distortion is considered to be negligible in the above formulation. If significant distortion is present in the input data, it may affect the estimation and eliminate many valid correspondences. In such a case, more elaborate methods, such as presented by Claus and Fitzgibbon [36], should be used.

2.4.3 Camera And Structure Recovery

Camera Matrices The previous section has introduced the fundamental matrix $\mathtt{F}$ as a tool for outlier elimination. Once the robust estimation is completed, the best estimate of $\mathtt{F}$, which has been created as a by-product in the estimation process, can be used to create initial camera matrices $\mathtt{P}$ and $\mathtt{P}'$:

$$\mathtt{P} = [\,\mathtt{I}\,|\,\mathbf{0}\,] \quad \text{and} \quad \mathtt{P}' = \Big[\,[\mathbf{e}']_\times \mathtt{F}\,\Big|\,\mathbf{e}'\Big] \quad , \tag{2.25}$$

where $\mathbf{e}'$ is the *epipole* in the second image (see Figure 2.5) and $[\cdot]_\times$ denotes the skew-symmetric matrix for the expression of the cross product in matrix form. The epipole $\mathbf{e}'$, the point of intersection of all epipolar lines $\mathbf{l}'_{\mathbf{e}'}$, can be computed from $\mathbf{e}'^\top \mathtt{F} = \mathbf{0}$.

Triangulation Once camera matrices $\mathtt{P}$ and $\mathtt{P}'$ have been determined for two images, initial 3D object points may be triangulated. This is accomplished by creating an equation of the form $\mathtt{A}\mathbf{X} = \mathbf{0}$ from the relations $\mathbf{x} = \mathtt{P}\mathbf{X}$ and $\mathbf{x}' = \mathtt{P}'\mathbf{X}$. From the linearly independent entries of the cross product expressions $\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}$ and $\mathbf{x}' \times (\mathtt{P}'\mathbf{X}) = \mathbf{0}$, $\mathtt{A}$ can be created as

$$\mathtt{A} = \begin{bmatrix} x\mathbf{P}^{3\top} - \mathbf{P}^{1\top} \\ y\mathbf{P}^{3\top} - \mathbf{P}^{2\top} \\ x'\mathbf{P}'^{3\top} - \mathbf{P}'^{1\top} \\ y'\mathbf{P}'^{3\top} - \mathbf{P}'^{2\top} \end{bmatrix} \quad , \tag{2.26}$$

where $\mathbf{P}^{i\top}$ and $\mathbf{P}'^{i\top}$ are the rows of the camera matrices $\mathtt{P}$ and $\mathtt{P}'$. The 3D object point $\mathbf{X}$ is then given by the unit singular vector corresponding to the smallest singular value of $\mathtt{A}$, which can determined by SVD.

2.4.4 Maximum-Likelihood Estimation

The last step to perform for the initial reconstruction is a maximum-likelihood estimation to minimize the reprojection error. The appropriate formulation is

$$\underset{\mathtt{P},\mathtt{P}',\mathbf{X}}{\arg\min} \quad \sum_{k=1}^{K} \mathrm{d}(\mathbf{x}_k, \mathtt{P}\mathbf{X}_k)^2 + \mathrm{d}(\mathbf{x}'_k, \mathtt{P}'\mathbf{X}_k)^2 \quad . \tag{2.27}$$

The optimization procedure – bundle adjustment – will be described in detail in Section 2.7.

2.5 Extension

The previous section has sketched an approach that yields an initial 3D reconstruction from two images: Correspondences between suitable images are found, outlier correspondences are eliminated and the fundamental matrix is obtained, camera matrices are calculated from this fundamental matrix, and finally 3D points are triangulated, followed by a bundle adjustment.

This initial reconstruction for two images is a basic building block for a system that handles image sequences of arbitrary length (or image collections of arbitrary size). In principle, there are two possible ways to proceed with the reconstruction: hierarchically or incrementally.

2.5.1 Hierarchical Reconstruction

A hierarchical reconstruction relies on the (recursive) partition of the input data until overlapping subsets of the desired size for initial reconstruction are obtained. The partition may either be carried out until further partition is impossible (see Fitzgibbon and Zisserman [43] and Lhuillier and Quan [92], for example), or only as required (see Nistér [114] and Gibson et al. [55], for instance). The individual initial reconstructions are merged pairwise, and a bundle adjustment is performed, followed by the next merge, until the reconstruction comprises all images. If the sequence is not fully partitioned, the reconstruction for the intermediate images can be obtained using the incremental reconstruction approach described in the next paragraph.

Hierarchical reconstruction requires all data to be present when the reconstruction is started, which makes it unsuitable for online processing.

2.5.2 Incremental Reconstruction

An incremental reconstruction extends an initial reconstruction by adding images one after another. First, correspondences are established as described in Section 2.4.1. The next step is again outlier elimination, but since 3D object points are already available, a camera matrix-based approach may be used instead of the fundamental matrix outlier elimination of Section 2.4.2. In this case, the outlier elimination directly yields the initial camera matrix for the newly added image. Triangulation proceeds as before, complementing the set of 3D object points using any new correspondences, provided there is enough baseline for reliable estimation. The last step is again a maximum-likelihood estimation, bundle adjustment. It is usually prudent to first optimize the newly created camera matrix and object points, before optimizing all estimates obtained so far.

A drawback of the incremental reconstruction is that the computational burden of the bundle adjustment increases with each added image.

Projection Matrix Outlier Elimination In Section 2.4.2, the fundamental matrix is used for outlier elimination, since no 3D objects are available at that stage. When a pre-existing reconstruction is extended, however, one may directly use the 2D-to-3D relations $\mathbf{x} \leftrightarrow \mathbf{X}$ to estimate a camera matrix for the new image and eliminate outlier correspondences in the process.

Two linearly independent equations can be derived from Equation (2.5) for each correspondence $\mathbf{x} \leftrightarrow \mathbf{X}$:

$$\begin{bmatrix} \mathbf{0}^\top & -\mathbf{X}^\top & y\mathbf{X}^\top \\ \mathbf{X}^\top & \mathbf{0}^\top & -x\mathbf{X}^\top \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \quad , \tag{2.28}$$

with $\mathbf{P}^{i\top}$ begin the i-th row of $\mathtt{P}$. Stacking equations from $\beta = 6$ correspondences yields a linear system $\mathtt{A}\mathbf{p} = \mathbf{0}$, where the vectors $\mathbf{P}^i$ are concatenated into $\mathbf{p}$. Strictly speaking, only $5\frac{1}{2}$ correspondences are necessary, as the camera matrix $\mathtt{P}$ has only 11 degrees of freedoms, since it is only defined up to scale.

The per-datum error function e_i for the camera matrix estimation is given by

$$e_i = \mathrm{d}(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 \quad . \tag{2.29}$$

As for outlier elimination based on the fundamental matrix in Section 2.4.2, this formulation does not include lens distortion, which may be an issue if distortion is not negligible.

Bundle Adjustment The cost function of Equation (2.27) can be extended to incorporate an arbitrary number of images:

$$\underset{\mathtt{P},\mathbf{X}}{\arg\min} \quad \sum_{j=1}^{J}\sum_{k=1}^{K} \mathrm{d}(\mathbf{x}_{j,k} \, , \, \mathtt{P}_k\mathbf{X}_j)^2 \tag{2.30}$$

Optimization of this cost function is discussed in Section 2.7.

2.6 Auto-Calibration

The formulations provided so far only permit a *projective* reconstruction. The structure of the scene is usually distorted, as angular relations between lines are not respected during the reconstruction process: angles are not invariant under projective transformations. Projective reconstructions are not directly suited for reconstruction and augmentation tasks; instead a *metric* reconstruction – a reconstruction in Euclidean space – is required. Metric reconstructions may easily be acquired by calibrating the camera offline with the help of a calibration pattern or object.

Auto-calibration refers to the process of upgrading a projective reconstruction to a metric one without the use of calibration patterns or objects, which greatly increases the flexibility of the reconstruction approach. This is achieved by the calculation of a rectifying homography or upgrading transformation $\mathtt{H}$ to transform a projective reconstruction $\{\mathtt{P}_j, \mathbf{X}_k\}$ to a metric reconstruction $\{\mathtt{P}_j\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_k\}$.

There are different approaches available for auto-calibration, most notable the approach by Maybank and Faugeras [100] based on the Kruppa equations, and methods based on the absolute dual quadric, introduced by Triggs [151]. The former approach requires only the fundamental matrix $\mathtt{F}$ between a pair of images to be known, but the equations are difficult to solve and lead to ambiguities, and the approach does not generalize well to more than two views. The remainder of this section therefore sketches the practical auto-calibration approach by Pollefeys et al. [124] based on the absolute dual quadric.

Absolute Dual Quadric The *absolute dual quadric (ADQ)* $\mathtt{Q}^*_\infty$ is a degenerate dual quadric that encodes both the *absolute conic* and the plane it is located on, the *plane at infinity.* It may be represented by a homogeneous 4×4 matrix of rank 3. Its projection is given by

$$\boldsymbol{\omega}^* = \mathtt{P}\mathtt{Q}^*_\infty\mathtt{P}^\top \quad , \tag{2.31}$$

which means that it projects to the *dual image of the absolute conic*:

$$\boldsymbol{\omega}^* = \mathtt{K}\mathtt{K}^\top \quad . \tag{2.32}$$

The above equations in conjunction allow the transfer of constraints on $\boldsymbol{\omega}^*$ to constraints on $\mathtt{Q}^*_\infty$ using the projective camera matrices $\mathtt{P}$. A thorough discussion of this topic including further information and a comprehensive description of the terminology is given by Hartley and Zisserman [62].

To obtain the rectifying homography $\mathtt{H}$, Pollefeys et al. [124] suggest to proceed by first normalizing the camera matrices as follows:

$$\mathtt{P}_\mathrm{N} = \mathtt{K}_\mathrm{N}^{-1}\mathtt{P} \quad , \text{with} \quad \mathtt{K}_\mathrm{N} = \begin{bmatrix} w+h & 0 & \frac{w}{2} \\ & w+h & \frac{h}{2} \\ & & 1 \end{bmatrix} \quad , \tag{2.33}$$

with w and h being the width and height of the corresponding image, and the subscript N denoting normalization. This normalization tries to ensure that the principal point is close to the origin and the focal length is approximately unity. Due to the normalizations, they are able to express $\boldsymbol{\omega}^*$ in terms of approximate values and reasonable standard deviations:

$$\boldsymbol{\omega}^* = \begin{bmatrix} f^2+s^2+o_x^2 & s\eta f+o_x o_y & o_x \\ s\eta f+o_x o_y & \eta^2 f^2+o_y^2 & o_y \\ o_x & o_y & 1 \end{bmatrix} \approx \begin{bmatrix} 1\pm 9.01 & 0\pm 0.01 & 0\pm 0.1 \\ 0\pm 0.01 & 1\pm 9.01 & 0\pm 0.1 \\ 0\pm 0.1 & 0\pm 0.1 & 1 \end{bmatrix} \quad . \tag{2.34}$$

The uncertainties are taken into account when constructing the constraint equations:

$$\boldsymbol{\omega}^*_{12} = 0 \quad \Rightarrow \quad \frac{1}{0.01}\left(\mathbf{P}_{\mathrm{N}}^{1\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{2}\right) = 0 \tag{2.35}$$

$$\boldsymbol{\omega}^*_{13} = 0 \quad \Rightarrow \quad \frac{1}{0.1}\left(\mathbf{P}_{\mathrm{N}}^{1\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{3}\right) = 0 \tag{2.36}$$

$$\boldsymbol{\omega}^*_{23} = 0 \quad \Rightarrow \quad \frac{1}{0.1}\left(\mathbf{P}_{\mathrm{N}}^{2\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{3}\right) = 0 \tag{2.37}$$

$$\boldsymbol{\omega}^*_{11} = \boldsymbol{\omega}^*_{22} \quad \Rightarrow \quad \frac{1}{0.2}\left(\mathbf{P}_{\mathrm{N}}^{1\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{1} - \mathbf{P}_{\mathrm{N}}^{2\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{2}\right) = 0 \tag{2.38}$$

$$\boldsymbol{\omega}^*_{11} = \boldsymbol{\omega}^*_{33} \quad \Rightarrow \quad \frac{1}{9.01}\left(\mathbf{P}_{\mathrm{N}}^{1\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{1} - \mathbf{P}_{\mathrm{N}}^{3\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{3}\right) = 0 \tag{2.39}$$

$$\boldsymbol{\omega}^*_{22} = \boldsymbol{\omega}^*_{33} \quad \Rightarrow \quad \frac{1}{9.01}\left(\mathbf{P}_{\mathrm{N}}^{2\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{2} - \mathbf{P}_{\mathrm{N}}^{3\top}\mathtt{Q}^*_\infty\mathbf{P}_{\mathrm{N}}^{3}\right) = 0 \quad , \tag{2.40}$$

with $\mathbf{P}_N^{i\top}$ being the i-th row of the normalized camera matrix $\mathtt{P}_N$. A system of linear equations of the form $\mathtt{A}\mathbf{q} = \mathbf{0}$ can be constructed from the above equations, where $\mathbf{q}$ contains the 10 unique entries of the symmetric 4×4 matrix $\mathtt{Q}^*_\infty$. The system of linear equations can be solved by SVD. The canonical form of the ADQ $\mathtt{Q}^*_\infty$ in Euclidean space is

$$\tilde{\mathtt{I}} = \begin{bmatrix} \mathtt{I}_{3\times 3} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \quad , \tag{2.41}$$

which allows the expression of the ADQ in a projective coordinate frame as

$$\mathtt{Q}^*_\infty = \mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\top \quad , \tag{2.42}$$

with $\mathtt{H}$ being a homography in projective space. Thus the upgrading transformation $\mathtt{H}$ may be obtained by performing an SVD of $\mathtt{Q}^*_\infty$. The smallest singular value of $\mathtt{Q}^*_\infty$ has to be forced to zero in order to enforce the rank-3 constraint.

2.7 Bundle Adjustment

Bundle adjustment is a maximum-likelihood estimation approach used in most SfM scenarios to ensure that the model parameters are accurate and reliable. It aims

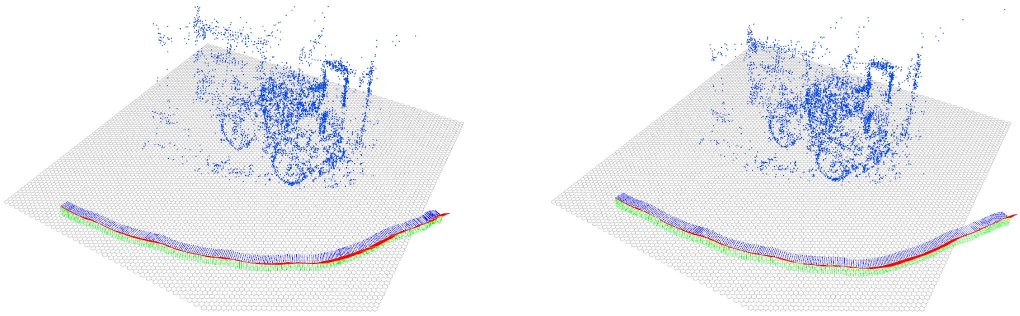


Figure 2.6: SfM-based 3D reconstruction before (left) and after (right) bundle adjustment immediately after auto-calibration has been performed. After the optimization, there are no signs of projective distortion (outward expansion of the scene parts far from the camera positions) remaining in the cloud of 3D object points. The individual positions of the camera path are indicated by the small colored coordinate systems.

to minimize the error metric of a given parametric model and the corresponding measurement. This error metric is typically the reprojection error of the reconstructed 3D object points $\mathbf{X}$, which are projected to the image plane of the virtual camera, with respect to the detected 2D feature points $\mathbf{x}$, such as given in Equation (2.27) and Equation (2.30). During optimization, the error is evenly distributed over all measurements by conjointly optimizing all the parameters involved. Figure 2.6 depicts a 3D reconstruction produced by SfM (after auto-calibration) before and after bundle adjustment in metric space.

Reprojection Error The total reprojection error that bundle adjustment aims to minimize has already been given for projective space in Equation (2.30). After auto-calibration (Section 2.6) has been performed, this formulation is no longer adequate. Instead, the cost function has to respect the projection of Equation (2.3), or rather Equation (2.19) if distortion is present. In metric space, the total reprojection error is thus

$$\operatorname*{arg\,min}_{\mathtt{K},\mathfrak{d},\mathtt{T},\mathbf{X}} \quad \sum_{j=1}^{J}\sum_{k=1}^{K} \mathrm{d}\Big(\mathbf{x}_{j,k}\,,\ \mathtt{K}_j\,\mathfrak{d}_j\Big(\left[\mathtt{I}|\mathbf{0}\right]\mathtt{T}_j^{-1}\mathbf{X}_k\Big)\Big)^2 \quad . \tag{2.43}$$

Note that the use of the Euclidean distance $\mathrm{d}(\cdot)$ implies that all observed 2D feature points have the same uncertainty (additive uniform Gaussian noise). This is a classical assumption, but it may be invalid for certain types of feature points. Scale invariant feature points, for example, typically have different accuracy in the x- and the y-coordinate, as shown by Zeisl et al. [167]. If such feature points are used, this has to be taken into account accordingly.

Least-Squares Formulation Equation (2.30) and Equation (2.43) have introduced the reprojection errors, the least-squares cost functions optimized by bundle adjustment. This section introduces a simplified, abstract formulation of these cost functions to facilitate the following discussions. To this end, let $\boldsymbol{\epsilon}$ be the residual vector of some functional relation $\mathbf{v} = \mathbf{h}(\mathbf{p})$:

$$\boldsymbol{\epsilon}(\mathbf{p}) = \mathbf{h}(\mathbf{p}) - \mathbf{v} \quad , \tag{2.44}$$

with $\mathbf{v}$ being the measurement vector and $\mathbf{p}$ being the parameter vector. The abstract least-squares cost function is then

$$g(\mathbf{p}) = \frac{\boldsymbol{\epsilon}(\mathbf{p})^\top \boldsymbol{\epsilon}(\mathbf{p})}{2} \quad . \tag{2.45}$$

The factor 2 in the denominator simplifies the following equations.

2.7.1 Algorithms

There are several algorithms available to solve least-squares optimization problems, some of which are presented in this section. When selecting one of these methods, there is generally a tradeoff between convergence behavior and computational burden or additional requirements. Newton's method, for example, converges rapidly but requires the evaluation of second derivatives, while gradient descent converges slowly but is easier to compute. More details are given by Nocedal and Wright [116] and Hartley and Zisserman [62].

Newton Newton's method in optimization is based on Taylor series expansion. For the least-squares cost function given in Equation (2.45), it casts the optimization problem as

$$\nabla^2 g(\mathbf{p}_i)\, \boldsymbol{\delta}_i^{\mathrm{N}} = -\nabla g(\mathbf{p}_i) \quad . \tag{2.46}$$

Starting from an initial guess $\mathbf{p}_0$, which has to be reasonable close to the final minimum, the update vector $\boldsymbol{\delta}_i^{\mathrm{N}}$ is used to refine the parameter vector in an iterative fashion:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \boldsymbol{\delta}_i^{\mathrm{N}} \quad . \tag{2.47}$$

Gauss-Newton The Gauss-Newton method is similar to Newton's method, but uses the approximation $\nabla^2 g_i \approx \mathtt{J}_i^\top \mathtt{J}_i$ for the *Hessian matrix* $\nabla^2 g$, the matrix of second-order partial derivatives of g with respect to $\mathbf{p}$. The matrix $\mathtt{J}$ is the *Jacobian matrix*, the matrix of all first-order partial derivatives of g with respect to $\mathbf{p}$. Using this approximations in the standard Newton equations from Equation (2.46) yields the following system:

$$\mathtt{J}_i^\top \mathtt{J}_i \boldsymbol{\delta}_i^{\mathrm{GN}} = -\mathtt{J}_i^\top \boldsymbol{\epsilon}_i \quad . \tag{2.48}$$

The relation $\nabla g(\mathbf{p}_i) = \mathtt{J}_i^\top \boldsymbol{\epsilon}_i$ is used to simplify the notation. The Gauss-Newton method thus avoids the evaluation of the Hessian matrix while still showing the same rapid convergence behavior as Newton's method close to the solution.

Gradient Descent The gradient of the cost function defines the direction of the cost function's most rapid decrease, and may therefore be used for iterative minimization. This minimization scheme is known as *gradient descent*. The length of the step ν_i may be obtained through a line search; the formulation for the optimization then being

$$\nu_i \boldsymbol{\delta}_i^{\mathrm{GD}} = -\mathtt{J}_i^\top \boldsymbol{\epsilon}_i \quad . \tag{2.49}$$

Gradient descent has slow convergence and therefore is not a good minimization strategy by itself.

Levenberg-Marquardt The Levenberg-Marquardt method combines the Gauss-Newton method and gradient descent. Its optimization scheme is given by

$$\left(\mathtt{J}_i^\top \mathtt{J}_i + \kappa_i \mathtt{I}\right) \boldsymbol{\delta}_i = -\mathtt{J}_i^\top \boldsymbol{\epsilon}_i \quad . \tag{2.50}$$

The parameter κ controls the influence of gradient descent on the whole iteration. It is usually initialized to a small value and then varied depending on whether the iteration was successful or not: if the current update does not lead to a decrease in the cost function, i. e., the iteration failed, κ is increased to give more emphasis to the gradient descent component.

2.7.2 The Sparse Levenberg-Marquardt Algorithm

Typical bundle adjustment problems may contain thousands of sets of extrinsic and intrinsic camera parameters, and hundreds of thousands of 3D object points. The time complexity of the Levenberg-Marquardt algorithm is cubic in general, which precludes arbitrary increases in the problem size. To alleviate this limitation on the performance, implementations of bundle adjustment usually exploit the sparse structure of the underlying problem: While the camera parameters depend on the parameters of the 3D object points they all observe, the 3D object points do not influence one another. The problem may thus be partitioned into a distinct block structure that, when exploited appropriately, dramatically reduces the required computation time. A sample illustration of this block structure for a toy example is provided in Figure 2.7. For further information, consult the work by Triggs et al. [153] or Hartley and Zisserman [62].

Block Structure The parameter vector $\mathbf{p}$ is partitioned according to the structure of the problem into sub-vectors $\mathbf{p} = (\mathbf{a}^\top, \mathbf{b}^\top)^\top$, where $\mathbf{a}$ comprises the extrinsic and

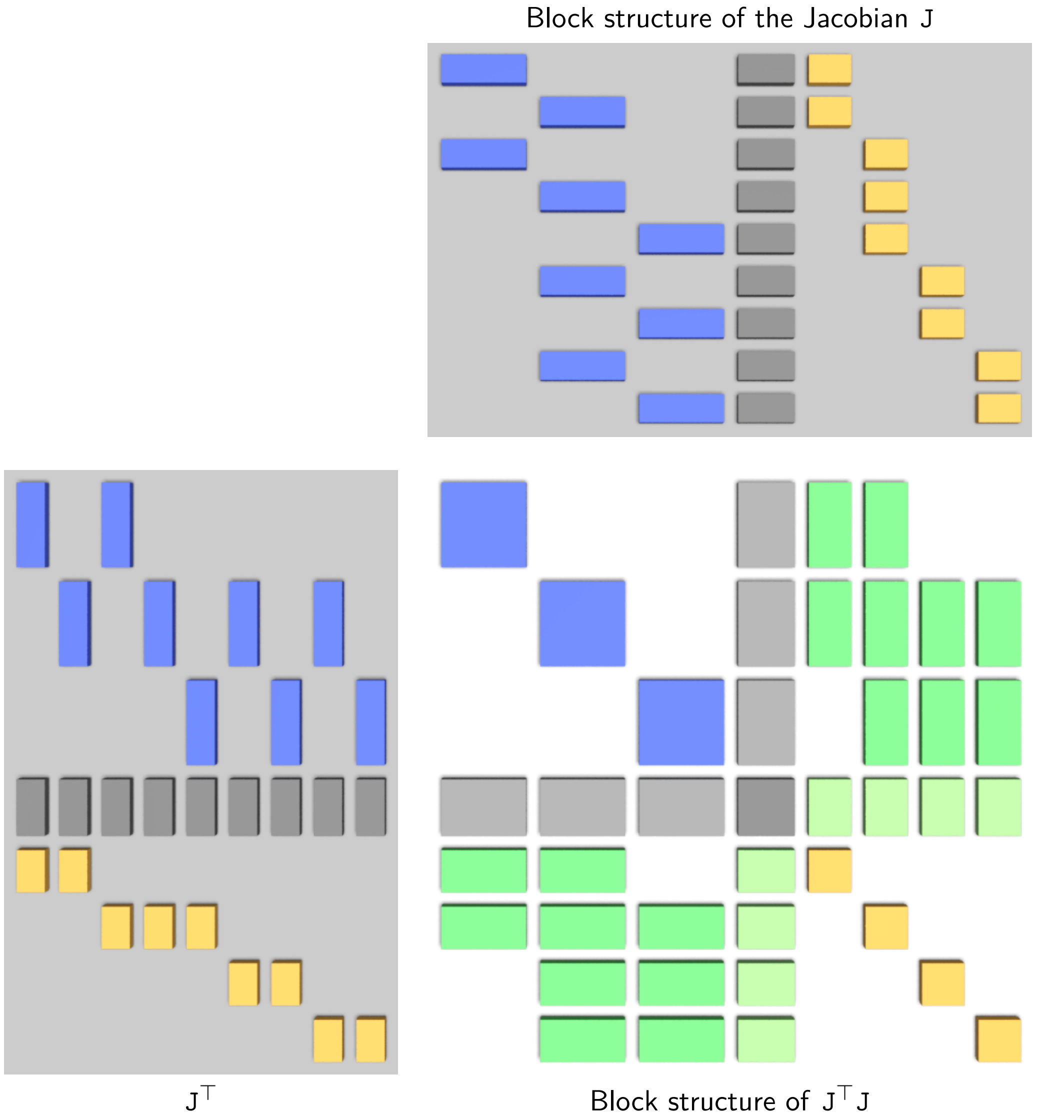

Block structure of the Jacobian $\mathtt{J}$
$\mathtt{J}^\top$
Block structure of $\mathtt{J}^\top\mathtt{J}$

Figure 2.7: Block structure of the Jacobian matrix $\mathtt{J}$ (top right) and of $\mathtt{J}^\top$ and $\mathtt{J}^\top\mathtt{J}$ (bottom left and right) for a toy example consisting of three images (with shared intrinsic parameters) and four reconstructed 3D object points. The first point is only visible in the first two images, the second point in all images, and the last two points only in the last two images. Blue blocks denote values corresponding to the extrinsic camera parameters, gray to the intrinsic camera parameters, and yellow to the 3D object point parameters. The blue and gray blocks together form the matrices $\mathtt{A}$ and $\mathtt{U} = \mathtt{A}^\top\mathtt{A}$, respectively. The green blocks of $\mathtt{J}^\top\mathtt{J}$ correspond to the matrix $\mathtt{W}$.

intrinsic camera parameters, and $\mathbf{b}$ the parameters of the 3D object points. Based on that, it is possible to partition the Jacobian

$$\mathtt{J} = \left[\frac{\partial \mathbf{h}(\mathbf{p})}{\partial \mathbf{p}}\right] \quad \text{into submatrices} \quad \mathtt{A} = \left[\frac{\partial \mathbf{h}(\mathbf{p})}{\partial \mathbf{a}}\right] \quad \text{and} \quad \mathtt{B} = \left[\frac{\partial \mathbf{h}(\mathbf{p})}{\partial \mathbf{b}}\right] \quad . \tag{2.51}$$

The block structure of the Jacobian is thus defined as $\mathtt{J} = [\mathtt{A}|\mathtt{B}]$, which gives rise to the following block structure for Equation (2.50):

$$\begin{bmatrix} \mathtt{A}^\top\mathtt{A} + \lambda\mathtt{I} & \mathtt{A}^\top\mathtt{B} \\ \mathtt{B}^\top\mathtt{A} & \mathtt{B}^\top\mathtt{B} + \lambda\mathtt{I} \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{a} \\ \boldsymbol{\delta}_\mathbf{b} \end{pmatrix} = \begin{pmatrix} -\mathtt{A}^\top\boldsymbol{\epsilon} \\ -\mathtt{B}^\top\boldsymbol{\epsilon} \end{pmatrix} \quad . \tag{2.52}$$

For convenience and simplicity, substitutions are made:

$$\begin{bmatrix} \mathtt{U}^* & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{a} \\ \boldsymbol{\delta}_\mathbf{b} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\epsilon}_\mathtt{A} \\ -\boldsymbol{\epsilon}_\mathtt{B} \end{pmatrix} \quad , \tag{2.53}$$

with the asterisk in $\mathtt{U}^*$ and $\mathtt{V}^*$ indicating the augmentation of the main diagonal.

Schur Complement To avoid solving the system of Equation (2.53) as a whole, it is partitioned into two smaller systems by the application of the *Schur complement.* To this end, Equation (2.53) is left-multiplied by the matrix

$$\begin{bmatrix} \mathtt{I} & -\mathtt{W}\mathtt{V}^{*-1} \\ \mathtt{0} & \mathtt{I} \end{bmatrix} \quad , \tag{2.54}$$

with $\mathtt{I}$ and $\mathtt{0}$ being the identity matrix and null matrix of appropriate size. The upper row of the reformulated system is independent of $\boldsymbol{\delta}_\mathbf{b}$:

$$\begin{bmatrix} \mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\top & \mathtt{0} \\ \mathtt{W}^\top & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{a} \\ \boldsymbol{\delta}_\mathbf{b} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\epsilon}_\mathtt{A} + \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_\mathtt{B} \\ -\boldsymbol{\epsilon}_\mathtt{B} \end{pmatrix} \quad . \tag{2.55}$$

Solution The first part of the solution, $\boldsymbol{\delta}_\mathbf{a}$, can be obtained by solving the system of linear equations

$$\left(\mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\top\right)\boldsymbol{\delta}_\mathbf{a} = -\boldsymbol{\epsilon}_\mathtt{A} + \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_\mathtt{B} \quad , \tag{2.56}$$

which is usually done by *Cholesky decomposition*, as the matrix obeys the requirement of being positive definite. The second part of the solution, $\boldsymbol{\delta}_\mathbf{b}$, may be calculated from

$$\boldsymbol{\delta}_\mathbf{b} = \mathtt{V}^{*-1}\left(-\boldsymbol{\epsilon}_\mathtt{B} + \mathtt{W}^\top\boldsymbol{\delta}_\mathbf{a}\right) \quad . \tag{2.57}$$

Both Equation (2.56) and Equation (2.57) contain the matrix inverse $\mathtt{V}^{*-1}$. While $\mathtt{V}^*$ is typically much larger in size than $\mathtt{U}^*$, it is a block diagonal matrix. Its inverse is composed of the inverses of all the individual 3 × 3 blocks, obtaining which has only negligible impact on the overall computation time, and may be parallelized on top of that.

Part I

Constrained Camera Motion Estimation

CHAPTER 3

Bundle Adjustment for Stereoscopic 3D

Stereoscopic 3D has recently made a reappearance in the movie industry, requiring the adaption of the traditional processing pipeline to stereoscopic input data. A stereoscopic camera model for bundle adjustment is developed in this chapter, which is applicable to a wide range of camera configurations and provides the efficiency of traditional methods and improved accuracy. This chapter is based on work by Kurz et al. [89].

3.1 Introduction and Outline

Computer generated special effects are ubiquitous in movies today. The extent to which *computer generated imagery (CGI)* is used ranges from small, almost insignificant objects to major parts of the movie, including actors and sets. The previous chapter has introduced SfM as a means to recover the parameters of the real camera and the sparse scene structure. These parameters are then used to create a virtual scene and place a virtual camera, allowing to composite the virtual objects with the real image sequences. Accurate and Reliable camera motion estimation is thus crucial in movie post-processing, as it is essential for the special effects to appear convincing.

Over the past couple of years, 3D films have made a reappearance, this time using modern *stereoscopic 3D (S3D)* technology. A consequence of that has been the creation of an unprecedented amount of high-resolution stereo image data. This new type of input data demands changes to the traditional processing pipelines, which are best equipped to deal with monocular input material.

Stereo image sequences have been used in computer vision over the past decades, but their predominant area of application has been robot and autonomous vehicle navigation and motion estimation. As a consequence, the employed stereo processing pipelines have to obey restrictive real-time requirements. In addition, the algorithms are only allowed to accumulate and process a limited amount of data.

In contrast to that, post-processing for movie productions is still done off-line. The amount of data involved precludes real-time processing, and since execution time is only a minor issue, computationally expensive algorithms may be used – algorithms, which are not yet updated to efficiently process stereo material.

This chapter describes an approach for reliable and accurate camera motion estimation for stereo sequences. An extended camera model for stereo cameras is presented. The model offers great flexibility in terms of its parameters and therefore can be employed for a variety of different cameras, ranging from entry-level consumer 3D camcorders using a 3D conversion lens with a static camera geometry to professional cameras used in movie productions. Furthermore, it is shown how the additional constraints introduced by the camera model can be incorporated into the sparse bundle adjustment framework of the previous chapter.

The approach is validated on a variety of data sets, from fully synthetic experiments to challenging real-world image sequences.

Outline The remainder of this chapter is organized as follows: Related work will be reviewed in the next section, followed by the introduction of the updated stereo scene model for SfM in Section 3.3. Section 3.4 introduces the new camera model for stereoscopic bundle adjustment, and the incorporation into bundle adjustment is described in Section 3.5. Results are shown in Section 3.6, followed by a discussion in Section 3.7.

3.2 Related Work

SfM Multi-camera systems in SfM usually assume a static and calibrated camera setup on a moving platform, such as the systems by Stewenius and Åström [139] or Kim et al. [82]. Another method is averaging the parameters of the independent reconstructions, demonstrated by Frahm et al. [46]. Di et al. [40] introduce the constraints arising from stereo geometry into bundle adjustment by simply adding soft constraints; the sparse structure of the problem is not addressed. Chandraker et al. [31] demonstrate an efficient stereo SfM framework using line features, but bundle adjustment is not used. Hirschmüller et al. [68] show a new system for correspondence generation and outlier elimination from stereoscopic image sequences, but the other stages of the reconstruction pipeline, in particular bundle adjustment, are not affected by these changes.

The research towards processing data of multiple independently moving cameras by Hasler et al. [64] or that towards the reconstruction of entire community photo collections by Goesele et al. [57] can be considered orthogonal to the work presented here.

Auto-calibration Zhang et al. [170] have explicitly modeled the problem of auto-calibration for an uncalibrated stereo rig with unknown motion for two pairs of stereo images. Brooks et al. [24] even consider varying vergence angles. The focus of these papers lies on providing a one-time calibration of these two stereo pairs instead of the optimization over a complete image sequence, however.

Stereo Navigation, Ego-motion Estimation, Visual Odometry Methods in robot or autonomous vehicle navigation and motion estimation that use stereo rigs, such as the systems described by Matthies et al. [99], Weng et al. [160] , Molton et al. [111], or Saeedi et al. [130], assume the rigs to be calibrated. Runtime constraints often require the problem of motion estimation to be reduced to estimating the parameters of an inter-frame motion model given two distinct sets of 3D points, and then feeding the results to a Kalman filter to achieve robustness. Olson et al. [117] use optimized feature selection and tracking, especially *multi-frame tracking*, to achieve robustness for tracking features over longer sequences.

Nistér et al. [115] and Sünderhauf et al.[142] assume calibrated stereo rigs for bundle adjustment in visual odometry, thus not optimizing the intrinsic camera parameters. A *reduced order bundle adjustment* is used by Dange et al. [39], where 3D object points are only parametrized by their depth in one image, thereby reducing the computational load of the system.

Mandelbaum et al. [97] present a correlation-based approach to ego-motion and scene structure estimation from stereo sequences, in which the transformation between left and right frames is assumed to be constant.

Uncalibrated Stereo Approaches for obtaining the epipolar geometry from uncalibrated stereo rigs have been presented by Zhang and Xu [169], Akhloufi et al. [2], Hartley et al. [63], Yin and Xie [165], and Ko et al. [85], among others, but these methods only consider a single pair of images without further optimization. Simond and Rives [137] determine the motion of an uncalibrated stereo rig under the assumption that a road plane is present and easy to identify in the images.

Uncalibrated, static stereo cameras are used in systems for visual servoing, such as the ones by Hodges and Richards [69], Shimizu and Sato [136], and Park and Chung [119], or for man-machine interaction, as shown by Cipolla et al. [35], but these systems do not perform explicit 3D reconstructions.

An approach for quasi-Euclidean epipolar rectification introduced by Fusiello and Irsara [50] has recently been adapted to work on uncalibrated stereo sequences by

Bleyer and Gelautz [16] and Cheng et al. [34], but no 3D reconstruction is performed.

Optical Flow, 3D Scene Flow Min et al. [106] and Huguet et al. [74] show systems for optical flow estimation from calibrated stereo setups, and how stereo constraints from setups consisting of an arbitrary number of cameras can be used is demonstrated by Zhang and Kambhamettu [168]. In these setups, the cameras are assumed to be calibrated. Vedula et al. [156] are able to recover the non-rigid scene motion, but again by using calibrated cameras. Optical flow estimation does not include maximum-likelihood estimation of the camera motion and scene structure over the whole image sequence.

Trinh and McAllester [154] adapt optical flow for ego-motion estimation, but their model only considers camera motion along the Z-axis.

Commercial Products Several commercial products feature tools for stereoscopic tracking and for stereoscopic camera solving (PFTrack™, SynthEyes™, or 3DEqualizer™, for example), but the corresponding algorithms have not been published.

3.3 Stereoscopic Scene Model

In contrast to the traditional formulation of Section 2.2, the input data for stereoscopic sequences consists of J *stereo frames* comprising two images each. At the same frame rate, a stereoscopic setup thus produces twice the amount of data per time unit. For convenience, the individual images of the stereo frame are denoted as $I_{j,\mathrm{L}}$ for the image of the left camera, and $I_{j,\mathrm{R}}$ for the image of the right camera. There are now also two separate camera matrices $\mathtt{P}_{j,\mathrm{L}}$ and $\mathtt{P}_{j,\mathrm{R}}$ for each stereo frame, and auto-calibration yields separate sets of intrinsic camera parameters $\mathtt{T}_{j,\mathrm{L}}$, $\mathtt{K}_{j,\mathrm{L}}$ and $\mathtt{T}_{j,\mathrm{R}}$, $\mathtt{K}_{j,\mathrm{R}}$. The set of 2D feature points is also distinctive for each camera, giving rise to points $\mathbf{x}_{j,k,\mathrm{L}}$ and $\mathbf{x}_{j,k,\mathrm{R}}$. The setup is illustrated in Figure 3.1.

Bundle Adjustment Cost Functions Introducing $x \in \{\mathrm{L}, \mathrm{R}\}$, the updated cost functions for bundle adjustment are

$$\operatorname*{arg\,min}_{\mathtt{P},\mathbf{X}} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{x} \mathrm{d}(\mathbf{x}_{j,k,x} \, , \, \mathtt{P}_{j,x}\mathbf{X}_k)^2 \tag{3.1}$$

for the projective case (as given by Equation (2.30)), and

$$\operatorname*{arg\,min}_{\mathtt{K},\mathtt{T},\mathbf{X}} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{x} \mathrm{d}\Big(\mathbf{x}_{j,k,x} \, , \, \mathtt{K}_{j,x}\mathtt{T}_{j,x}^{-1}\mathbf{X}_k\Big)^2 \tag{3.2}$$

for the metric case (as given by Equation (2.43)). Distortion is omitted from these formulations for the sake of simplicity.

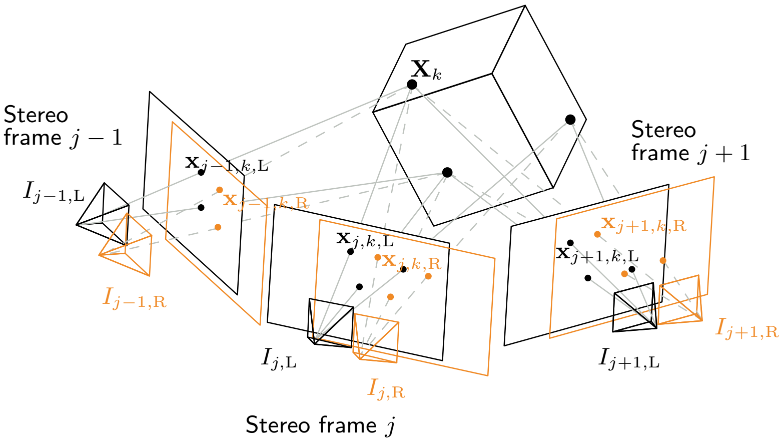

$\mathbf{X}_k$
Stereo frame $j-1$
$\mathbf{x}_{j-1,k,\mathrm{L}}$
$I_{j-1,\mathrm{L}}$
$\mathbf{x}_{j-1,k,\mathrm{R}}$
$I_{j-1,\mathrm{R}}$
Stereo frame $j+1$
$\mathbf{x}_{j+1,k,\mathrm{R}}$
$\mathbf{x}_{j+1,k,\mathrm{L}}$
$\mathbf{x}_{j,k,\mathrm{L}}$
$\mathbf{x}_{j,k,\mathrm{R}}$
$I_{j,\mathrm{L}}$
$I_{j,\mathrm{R}}$
$I_{j+1,\mathrm{R}}$
$I_{j+1,\mathrm{L}}$
Stereo frame j

Figure 3.1: Each stereo frame consists of a left camera image $I_{k,\,L}$ and a right camera image $I_{k,\,\mathrm{R}}$. In contrast to monocular SfM, there are now two sets of corresponding 2D feature points $\mathbf{x}_{j,k,\mathrm{L}}$ and $\mathbf{x}_{j,k,\mathrm{R}}$ for the set of 3D object points $\mathbf{X}_j$.

3.4 Stereoscopic Camera Model

This section describes the camera model for stereo bundle adjustment for the metric case. The projective formulation of SfM and bundle adjustment that has also been updated in the previous section, is of limited use for stereoscopic input data. The representation of the geometric constraints between the left and the right camera is not possible in the projective framework, because transformations in the local camera coordinate system including rotations and translations cannot be parametrized independently from the projective camera matrix. It is thus proposed to enforce the constraints introduced by the metric stereo camera model after an update from projective to metric space has been performed by auto-calibration, as described in Section 2.6.

Considering a standard stereo camera setup, the first observation is that the two cameras of the stereo system undergo only dependent motion – if the left camera translates to the right, the right camera will inherently have to follow that same translation. This dependency can be exploited to improve over the conventional bundle adjustment algorithm: Instead of treating the left and the right camera as separate entities, they are considered as elements of the same camera system. A change of parameters introduced by the left camera will therefore influence the whole system, and consequently the parameters of the right camera, and vice versa.

Furthermore, the total number of parameters representing the stereo camera system over the whole stereo sequence has to be reduced to benefit from the stereo camera model. For a consumer stereo setup, available for example as a 3D conversion lens

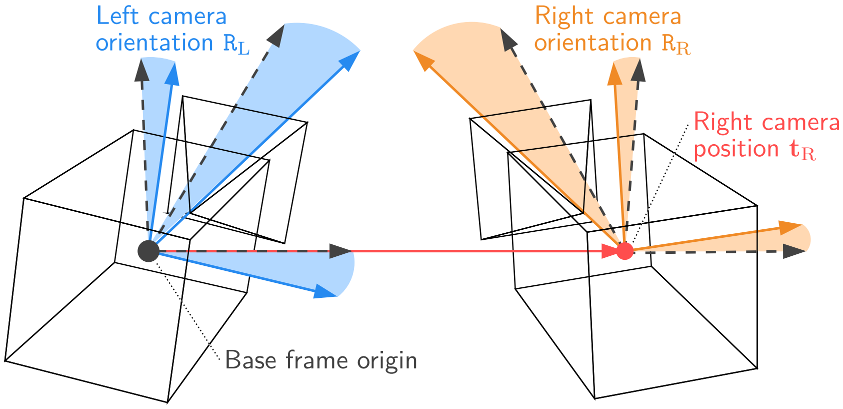


Figure 3.2: The novel camera model for bundle adjustment. The camera geometry of every stereo frame is given by a base frame (dashed lines), whose origin is aligned with the center of the left camera. The orientation $\mathtt{R}_\mathrm{L}$ of the left camera is encoded independent from the orientation of the base frame, allowing the position of the right camera to be specified by a single parameter vector $\mathbf{t}_\mathrm{R}$ (red arrow) for the whole sequence.

mounted onto a camcorder, the relative position offset of the two camera centers (representing the baseline between the cameras) and the relative rotation between the cameras can be assumed to be unknown but static, which significantly reduces the number of degrees of freedom of the system over the whole sequence.

Recalling Equation (2.3) and Equation (2.4), where the camera matrix is described by a calibration matrix $\mathtt{K}$ and a transformation matrix $\mathtt{T}$, this can be rewritten appropriately for the left and write camera to yield the stereo camera model:

$$\mathtt{P}_{j,\mathrm{L}} = \mathtt{K}_\mathrm{L} \left[\mathtt{R}_\mathrm{L}^\top \middle| \mathbf{0} \right] \mathtt{T}_j^{-1} , \tag{3.3}$$

$$\mathtt{P}_{j,\mathrm{R}} = \mathtt{K}_\mathrm{R} \left[\mathtt{R}_\mathrm{R}^\top \middle| -\mathtt{R}_\mathrm{R}^\top \mathbf{t}_\mathrm{R} \right] \mathtt{T}_j^{-1} , \tag{3.4}$$

where subscripts L and R denote parameters that are exclusive to the left and right camera respectively. The absence of the index j from the components of the actual stereo configuration in this case reflects that the stereo configuration is assumed to be moving but static. The setup is illustrated in Figure 3.2. This new stereo camera model leads to a bundle adjustment formulation different from the naïve one provided in Equation (3.2).

Unlike consumer stereo recording equipment, which usually precludes changes to the setup by the user, professional recording hardware allows changes to the setup, such as the variation of the point of convergence of the two cameras, during data acquisition. For such a setup, estimating a static frame of rotation between the cameras would not yield acceptable results.

Assuming the relative position offset of the two camera centers to be unknown but constant is a constraint that is always enforced, because the baseline between the cameras is usually not changed. As a matter of principle, there is some freedom in the choice of the stereo system base position. Here, it is chosen to coincide with the center of the left camera.

The rotation matrix of the left camera $\mathtt{R}_\mathrm{L}$ could be omitted for a static stereo setup. However, if the point of camera convergence changes in a dynamic setup, it is necessary to encode the orientation of the left camera separately from the orientation of the stereo system. This is due to the fact that a rotation of the left camera would otherwise inherently lead to a rotation of the coordinate frame in which the relative translation of the right camera takes place (see Fig. 3.2).

Depending on the actual acquisition system in operation, parameters can be chosen to be estimated for every frame, for a subset of frames, or for the whole sequence. Furthermore, the intrinsic camera parameters can of course be treated as shared between the two cameras, if this was the case at the time of recording.

3.5 Bundle Adjustment

The block structure that is exploited by the sparse Levenberg-Marquardt algorithm for bundle adjustment has already been discussed in Section 2.7.2. In this section, it is shown how the block structure can be modified to accommodate the stereo camera model.

First, it is demonstrated how the structure of the bundle adjustment problem can be modified for the estimation of joined intrinsic camera parameters, as this is needed as a basis for the joint estimation of parameters in the stereo camera model.

Joined Intrinsic Camera Parameters Joint estimation of parameters requires splitting the component $\mathbf{a}$ of the parameter vector $\mathbf{p}$ into subvectors $\mathbf{a}_0$ and $\mathbf{a}_1$. The structure of the Jacobian corresponding to this case is illustrated in Figure 3.3, upper left. Supposing that an image sequence was recorded by moving the camera around, but leaving all camera settings unchanged. It may thus be assumed that the intrinsic camera parameters do not change over the course of the sequence. In this case, the subvector $\mathbf{a}_0$ may represent the extrinsic camera parameters, with one set of parameters for every image in the sequence, and the subvector $\mathbf{a}_1$ the intrinsic camera parameters, with one set of parameters total. This may expressed by using the following block structure for $\mathtt{U}$:

$$\mathtt{U} = \begin{bmatrix} \mathtt{A}_0^\top \mathtt{A}_0 & \mathtt{A}_0^\top \mathtt{A}_1 \\ \mathtt{A}_1^\top \mathtt{A}_0 & \mathtt{A}_1^\top \mathtt{A}_1 \end{bmatrix} \quad . \tag{3.5}$$

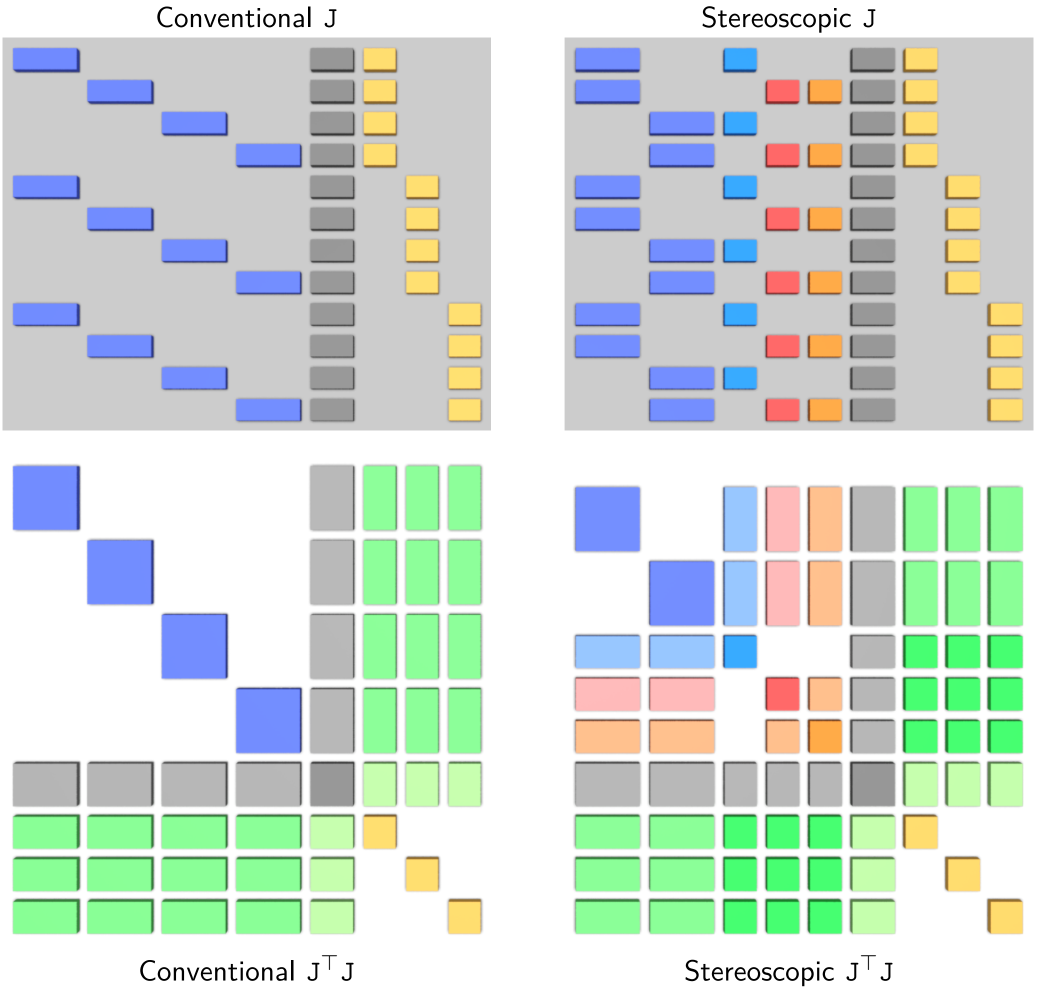


Figure 3.3: Block structure of the Jacobian matrix J (top) and the matrix $J^\top J$ (bottom) for Conventional bundle adjustment (left) and Stereoscopic bundle adjustment assuming a static stereo setup (right) for a toy example consisting of four images and three reconstructed 3D object points. The points are assumed to be visible in all images. The individual block matrices are set apart by different coloring: dark blue for the extrinsic camera parameters (Conventional) and base frame parameters (Stereoscopic), gray for the shared intrinsic camera parameters, light blue for the left camera orientation, salmon for the right camera position, orange for the right camera orientation, and yellow for the 3D object point parameters. Note the differences in the structure of the Jacobian between the two variants.

Model parameters		# of parameters	# of vector elements	designation
Base frame	$\mathbf{t}$, $\mathtt{R}$	6	J	$\mathbf{a}_0$
Left orientation	$\mathtt{R}_\mathrm{L}$	1-3	J, 1 (Joined)	$\mathbf{a}_2$
Right position	$\mathbf{t}_\mathrm{R}$	3	1	$\mathbf{a}_3$
Right orientation	$\mathtt{R}_\mathrm{R}$	1-3	J, 1 (Joined), 0 (Shared)	$\mathbf{a}_4$
Left intrinsics	$\mathtt{K}_\mathrm{L}$	5	J, 1 (Joined)	$\mathbf{a}_1$
Right intrinsics	$\mathtt{K}_\mathrm{R}$	5	J, 1 (Joined), 0 (Shared)	$\mathbf{a}_1$
3D object points	$\mathbf{X}$	3	K	$\mathbf{b}$

Table 3.1: Stereo model parameters with their typical parameter count, the number of elements in the associated vector, and the designation of the corresponding vector. Example: For a sequence of $J = 10$ images, $\mathbf{a}_0$ contains 10 elements with 6 parameters each, i.e., 60 entries in total. Joined indicates that the parameters are constant and are jointly estimated over the whole sequence. Shared indicates that the respective parameters of the right camera are estimated in combination with the corresponding parameters of the left camera, so that there are no separate entries for these parameters in the matrix $\mathtt{J}^\top\mathtt{J}$.

$\mathtt{A}_0^\top\mathtt{A}_0$ is a block-diagonal matrix, as was the case before, but the intrinsic camera parameters are collected into $\mathtt{A}_0^\top\mathtt{A}_1$ and $\mathtt{A}_1^\top\mathtt{A}_1$, a dense vector of blocks and a single block. This structure is further illustrated in Figure 3.3, lower left.

Stereo Camera Model For the stereo camera model, the partition of $\mathbf{a}$ is substantially modified:

$$\mathbf{a} = \left(\mathbf{a}_0^\top, \mathbf{a}_1^\top, \mathbf{a}_2^\top, \mathbf{a}_3^\top, \mathbf{a}_4^\top\right)^\top \quad . \tag{3.6}$$

The designation of the corresponding subvectors for all parameters of the camera model is summarized in Table 3.1, along with a listing of the number of parameters and the number of the respective vector entries. The structure of the Jacobian matrix for this case is illustrated in Figure 3.3, upper right. Most parameters can either be assumed to be variable for each frame or joined (i.e., estimated conjointly) over the whole sequence. The intrinsic parameters can also be shared for both cameras.

For the sake of simplicity, a static stereo setup with joined and shared intrinsic parameters will be assumed henceforth, resulting in two single rotation matrices $\mathtt{R}_L$ and $\mathtt{R}_R$ over the whole sequence, and a single calibration matrix $\mathtt{K}$. This would be the case in a stereo setup with a fixed convergence point, e.g., a camcorder with a 3D conversion lens.

In principle, it would also be possible to restrict $\mathtt{R}_\mathrm{L}$ and $\mathtt{R}_\mathrm{R}$ in a way that makes them depend on the vergence angle only. Dependent on the degrees of freedom for the convergence point, this results in 1 or 2 degrees of freedom for the rotation matrices $\mathtt{R}_\mathrm{L}$ and $\mathtt{R}_\mathrm{R}$ (as indicated in Table 3.1).

Following the partition of the vector $\mathbf{p}$ above, the component $\mathtt{A}$ of the Jacobian matrix $\mathtt{J}$ has the block structure $\mathtt{A} = [\,\mathtt{A}_0\,\mathtt{A}_1\,\mathtt{A}_2\,\mathtt{A}_3\,\mathtt{A}_4\,]$, where $\mathtt{A}_i = \partial\mathbf{h}(\mathbf{p})/\partial\mathbf{a}_i$. The resulting block structure for the $\mathtt{U}$ matrix is given by

$$\mathtt{U} = \begin{bmatrix} \mathtt{A}_0^\top\mathtt{A}_0 & \mathtt{A}_0^\top\mathtt{A}_1 & \mathtt{A}_0^\top\mathtt{A}_2 & \mathtt{A}_0^\top\mathtt{A}_3 & \mathtt{A}_0^\top\mathtt{A}_4 \\ \mathtt{A}_1^\top\mathtt{A}_0 & \mathtt{A}_1^\top\mathtt{A}_1 & \mathtt{A}_1^\top\mathtt{A}_2 & \mathtt{A}_1^\top\mathtt{A}_3 & \mathtt{A}_1^\top\mathtt{A}_4 \\ \mathtt{A}_2^\top\mathtt{A}_0 & \mathtt{A}_2^\top\mathtt{A}_1 & \mathtt{A}_2^\top\mathtt{A}_2 & \mathtt{0} & \mathtt{0} \\ \mathtt{A}_3^\top\mathtt{A}_0 & \mathtt{A}_3^\top\mathtt{A}_1 & \mathtt{0} & \mathtt{A}_3^\top\mathtt{A}_3 & \mathtt{A}_3^\top\mathtt{A}_4 \\ \mathtt{A}_4^\top\mathtt{A}_0 & \mathtt{A}_4^\top\mathtt{A}_1 & \mathtt{0} & \mathtt{A}_4^\top\mathtt{A}_3 & \mathtt{A}_4^\top\mathtt{A}_4 \end{bmatrix}, \tag{3.7}$$

with $\mathtt{0}$ being null matrices of the appropriate sizes introduced by the independence of the parametrization of the left camera from the parametrization of the right camera, excluding the base frame and intrinsic parameters. This structure is further illustrated in Figure 3.3, lower right. The size of the individual blocks may be derived from Table 3.1. The layout of block $\mathtt{A}_i^\top\mathtt{A}_j$ corresponds to that of the combination of $\mathbf{a}_i$ in vertical and $\mathbf{a}_j$ in horizontal direction.

When comparing the structure of the matrix $\mathtt{J}^\top\mathtt{J}$ taken from the stereo bundle adjustment and from a conventional bundle adjustment (see Figure 3.3, bottom row), it becomes evident that the only block affected by the changes is the top left block, corresponding to the matrix $\mathtt{U}$ (and, consequently, the matrix $\mathtt{W}$).

These changes do affect the sparsity of the equation system to solve, but for an increasing number of stereo frames only insignificantly so. Note that the Schur complement trick has yet to be performed for the depicted matrices – the sparse structure depends on point visibility, and if the majority of points is visible in most frames, the matrix in question may not be sparse at all. A significant effect is brought about by the reduction of overall block size, on the other hand, especially if the stereo setup may be assumed to be static, which leads to improved computational performance.

3.6 Results

In this section, the evaluation of the stereo bundle adjustment with purely synthetic data, rendered sequences and real-world sequences is presented.

For the synthetic and rendered experiments, the results are evaluated using three different algorithms for bundle adjustment: a conventional bundle adjustment (denoted as Unconstrained), a conventional bundle adjustment with joined focal length over the sequence (denoted as Joined), and the novel stereo bundle adjustment (denoted as Stereo). The focal length is treated as unknown but constant for Joined and Stereo, and it is therefore estimated conjointly for the whole input image sequence, whereas it may vary between frames for Unconstrained.

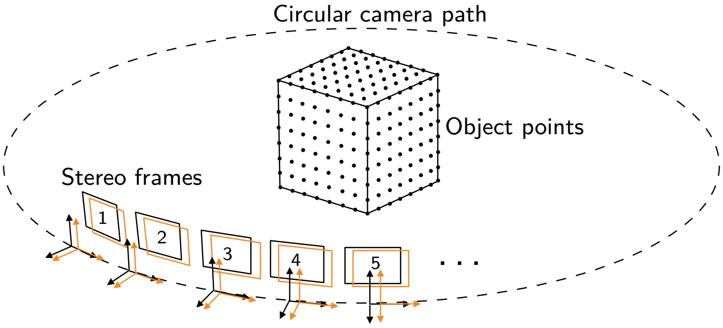


Figure 3.4: The setup used in the synthetic experiments for the generation of the ground-truth camera and 3D object point parameters. Stereo frames are generated while the camera performs a circular motion around a set of object points arranged on the surface of a cube.

Preprocessing For the rendered and real-world experiments, the stereoscopic image sequences are first processed by traditional SfM, where the images of the left and right camera are interleaved. The small displacements between the left and right camera image are well suited for KLT tracking in this fashion. After projective reconstruction, auto-calibration is performed, and initial parameters for Joined and Stereo are determined. Then the three different bundle adjustments are executed.

Synthetic Experiments The setup for the synthetic experiments is sketched in Fig. 3.4. It consists of a virtual stereo configuration composed of two cameras. The cameras execute a circular motion around a set of 296 3D object points arranged in a regular grid on the surface of a cube. The cube has an edge length of 100 mm, the radius of the camera path is 300 mm, and the opening angle of the cameras is 30 degrees.

A total of 40 stereo pairs is generated per trial, providing 80 images per sequence. All the ground-truth measurements for the 2D feature points contained in these images are calculated from the known ground-truth camera and 3D object point parameters. In a last step before the reconstruction process, Gaussian noise with a standard deviation σ_{syn} is applied to the measurements.

The value of σ_{syn} is varied, and for each value a total of 1000 trials is performed for Unconstrained, Joined and Stereo. A different random disturbance is introduced in the measurements each trial. For each reconstruction, a similarity transformation is estimated to register it to the ground-truth data, and then the average absolute position and orientation error is calculated. Stereo can be expected to yield better reconstruction accuracy for noisy data, as the additional constraints reduce the influence of the noise on the parameter estimation. This is evident in the results in Figure 3.5, left column. The *root-mean-square error (RMSE)* in the estimated focal length is significantly reduced for

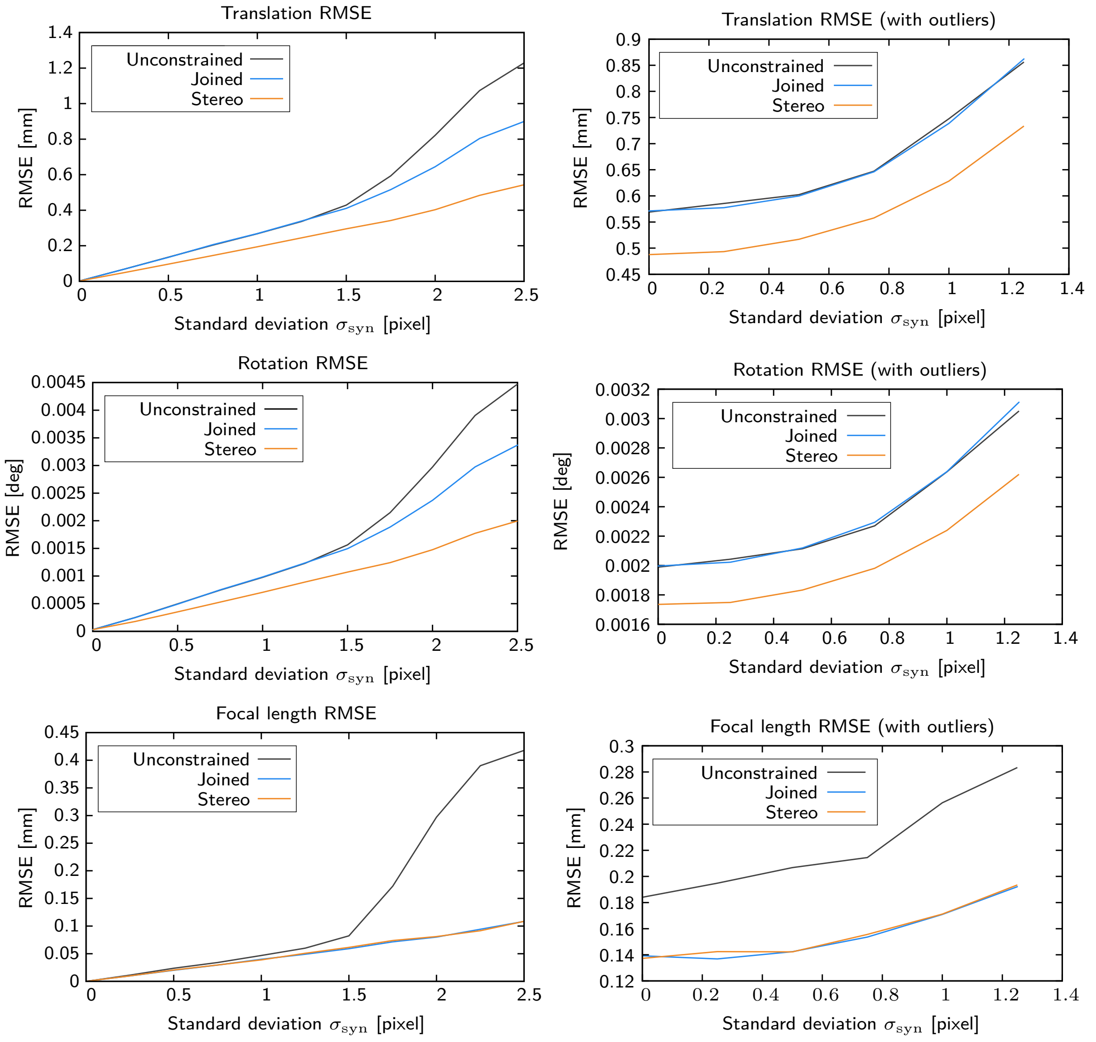


Figure 3.5: **Synthetic experiments:** Average translation, rotation, and focal length error observed for a given Gaussian error σ_{syn} of the 2D feature points over 1000 trials. For the results in the right column, 20 percent of the feature points were additionally disturbed by a large offset. The setup sketched in Figure 3.4 was used for the generation of the ground-truth parameters. The additional constraints of the camera model allow Stereo to outperform both other methods.

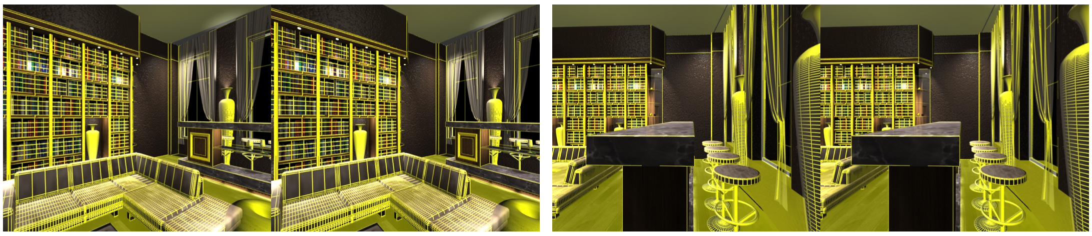


Figure 3.6: **Rendered experiments:** Two example stereo frames from the image sequence augmented with the wireframe model of the virtual scene placed using the estimated camera parameters from Stereo. The overlay fits the true scene geometry almost perfectly.

Joined and Stereo, and Stereo shows a significant improvement over the other methods in terms of translation and rotation RMSE in addition.

Furthermore, to simulate outliers, another test series was conducted. In this series, 20 percent of the measurements were disturbed by an offset of up to 12 pixel in addition to the Gaussian noise. Since not all outliers can be removed in the outlier elimination step (Section 2.4.2), the results obtained in this setting are different from the ones shown before. Despite that, Stereo should again perform better than the other methods in the presence of noise and outliers. This is indeed the case, as is shown in Figure 3.5, right column. Stereo outperforms both competitors again in terms of translation and rotation RMSE again, while it is on par with Joined in terms of focal length RMSE.

Rendered Experiments A virtual scene was rendered to obtain an image sequence with known ground-truth parameters. Figure 3.6 shows two sample stereo frames from this sequence (with a wireframe overlay using the camera parameters estimated by Stereo). The findings from the synthetic tests should carry over this experiment to a certain extent, and as can be seen from the RMSE values in Table 3.2, this is the case: Stereo provides the best results (the wireframe fits the true scene geometry almost perfectly in Figure 3.6), though the advantage over Unconstrained is less pronounced than in the synthetic experiments.

Real-world Experiments The evaluation of Stereo is concluded by the application of the algorithm to real-world image sequences.

The first image sequence was captured with a Panasonic® HDC-SDT750 consumer camcorder with a 3D conversion lens. It depicts some pieces of garden furniture. Figure 3.7, left, shows three sample stereo frames from this sequence. The estimated camera parameters have been used to place a cuboid on the surface of the table in the virtual scene, which has then been rendered on top of the image material. As can

RMSE	Unconstrained	Joined	Stereo
Translation	1.7274 mm	0.6459 mm	0.5964 mm
Rotation	0.0112 deg	0.0026 deg	0.0024 deg
Focal length	1.3609 mm	0.0975 mm	0.0600 mm
Average time	719 ms	860 ms	733 ms

Table 3.2: **Rendered experiments:** Average translation, rotation, and focal length error, and average time per iteration for Unconstrained, Joined, and Stereo. Stereo has the lowest error for all three measures. The difference in processing time between Joined and Stereo gets more pronounced as more frames are added, as Stereo has to estimate less parameters per additional frame.

be seen by this overlay geometry, the stereo bundle adjustment was able to obtain excellent results for the camera parameters.

The second real-world image sequence depicts a scene at a train station from *Grand Canyon Adventure 3D* (courtesy of *MacGillivray Freeman Films*). Professional equipment was used to record this sequence. As described before, the result of the reconstruction has been used to overlay the image sequence with virtual geometry to demonstrate the quality of the estimated parameters, which can be seen in Figure 3.7, right.

3.7 Discussion

The novel stereo camera model for use in bundle adjustment presented in this chapter has the generality to accommodate a wide range of the stereo cameras used today, and can be incorporated efficiently into the conventional sparse bundle adjustment algorithms. The conducted tests show that the use of the new stereoscopic camera model significantly increases the accuracy of the estimation in presence of noise and outliers. The reduction in the number of parameters used to describe the model enables significant reductions in the computation time required.

Limitations The formulation presented assumes fixed translations between the centers of the left and right camera. This assumption is required in order not to overparametrize the problem in respect to the conventional formulation. This could prove problematic for camera configurations where this assumption is violated.

Future Work For future work, it would be interesting to further investigate the implications of stereoscopic input data for the traditional processing pipeline. Though the creation of correspondences works satisfactory with the employed image interleaving

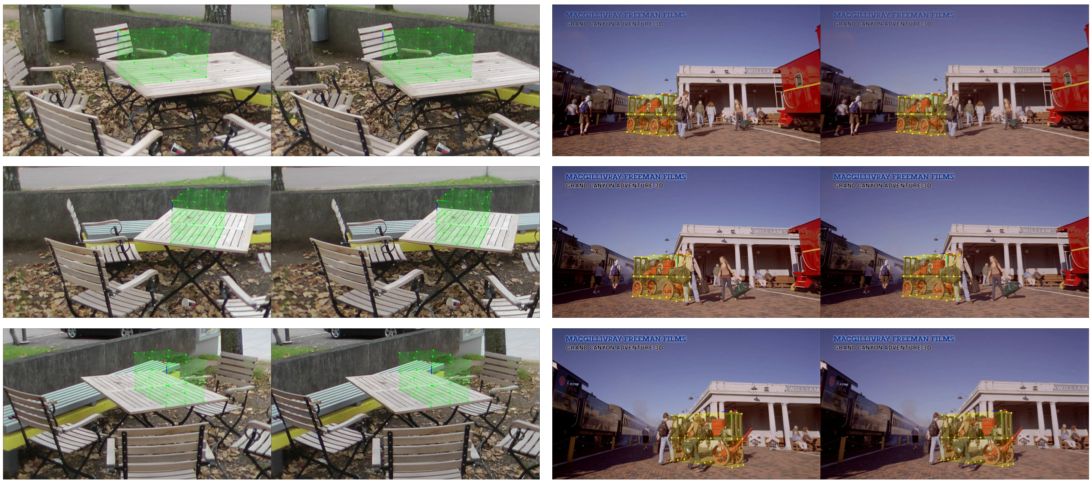


Figure 3.7: **Real-world experiments:** The frames on the left depict garden furniture and were recorded using a consumer HD camcorder with a 3D conversion lens. The frames on the right depict a scene at a train station and were recorded using professional equipment. Both sequences were augmented by a cuboid to demonstrate the quality of the estimated camera parameters.

scheme, a dedicated stereo detection and outlier elimination algorithm may further improve the results.

Future work might also include the investigation of the effects of different types of parametrizations on the quality of the results. To this end, extensive tests with different stereo cameras and different parametrizations have to be conducted. Alternative parametrizations could for example be based on a single vergence angle only.

CHAPTER 4

A Generalized Framework for Constrained Bundle Adjustment

This chapter introduces hierarchies of Euclidean transformations as a means for constrained bundle adjustment. The Euclidean transformations provide a framework able to handle many types of camera and scene constraints simultaneously in an intuitive and flexible way. It can be seen as a generalization of the stereoscopic camera model for bundle adjustment described in the previous chapter.

4.1 Introduction

In the previous chapter, a stereoscopic camera model for bundle adjustment has been introduced. This specialized model serves to reduce the number of parameters of the overall estimation process while still representing the real camera geometry, thus reducing overparametrization and providing enhanced reconstruction precision.

Overparametrization may not only be encountered in the description of the camera geometry, but also in the description of the scene. Reconstructed points may be collinear, coplanar, or share angular relations, for example. A result of overparameterization may be an unsatisfactory reconstruction in spite of a low reprojection error.

The constraints arising from the stereoscopic camera model of the previous chapter were introduced into bundle adjustment via additional rotational and translational components, or more generally: transformations. Using this approach as inspiration, this chapter will seek to improve on the stereo camera model to provide an elegant and intuitive framework for constraints in bundle adjustment based on hierarchies of Euclidean transformations. Hierarchies of Euclidean transformations can be used to represent dependencies between constraints, and allow efficient incorporation into existing bundle adjustment procedures.

The space of constraints addressed is coplanarity, collinearity, angular relations, distances, and parallelism, which can be conveniently expressed in terms of hierarchies of Euclidean transformations and therefore handled in a common mathematical framework.

Previous methods for constrained bundle adjustment, which will be reviewed in the next section, lack the ability to model constraints on the scene structure and on the camera geometry simultaneously, and are typically not able to describe all constraints in a consistent, homogenous way.

The novel approach for constrained bundle adjustment is flexible and applicable in many different scenarios, including stereo camera and moving object modeling.

As this approach makes explicit use of the properties of Euclidean space, for the remainder of this chapter it will be assumed that a perspective reconstruction and metric upgrade of the input data has been performed, as described in Chapter 2.

Outline This chapter continues with a review of related work in the next section, before methods for constrained bundle adjustment are reviewed in Section 4.3. Section 4.4 describes the new approach for constrained bundle adjustment based on hierarchies of Euclidean transformations, and Section 4.5 provides an example of how this approach might be used in order to construct a constrained parallelepiped. Section 4.6 presents application examples. This chapter is concluded by a discussion in Section 4.8.

4.2 Related Work

Lagrange Multipliers The method of Lagrange multipliers is commonly used to solve many constrained numerical optimization problems in mathematics. Triggs et al. [153] discuss the application of the Lagrange multiplier method to bundle adjustment in general. The matter is also described by McLauchlan et al. [102] in detail. They employ recursive partitioning in a variable state dimension filter formulation of SfM for efficiency. Meidow et al. [103] show how the method of Lagrange multipliers can be applied to fundamental matrix estimation and the constrained estimation of homogenous entities in general.

Weighting Schemes McGlone [101] and Hrabáček and van den Heuvel [71] present systems for bundle adjustment that introduce geometric constraints as additional pseudo-observations. Szeliski and Torr [143] include weighted coplanarity constraints in the object point optimization stage of an interleaved bundle adjustment scheme. The last approach in particular may lead to worse convergence behavior.

Reparametrization Förstner [45] has presented an approach involving minimal representations in projective space. Smith et al. [138] and Bartoli and Sturm [10] have shown how coplanarity constraints can be introduced into SfM, and Cornou et al. [38] present a system that can handle many types of constraints for user-selected points. An interactive system with constrained optimization of data containing orthonormal sets of lines and planes was presented by Robertson and Cipolla [125]. Fua [47] has presented a system for the optimization of parametrized head models using bundle adjustment. Geometric constraints are introduced into camera calibration by modeling parallelepipeds by Wilczkowiak et al. [161]. Bondyfalat and Bougnoux [20] investigate the application of Euclidean constraints during auto-calibration using a geometric reasoning system, resulting in high computational cost.

Moving Objects Multibody SfM, where the scene is composed of several independently moving, rigid objects, has been investigated by Fitzgibbon and Zisserman [44] and Ozden et al. [118], among others. Usually only image sequences from monocular cameras are considered, and a different set of extrinsic parameters is estimated for each moving object for each frame.

Different Camera Representations Holmes et al. [70] have presented a relative formulation for bundle adjustment, in which all subsequent frames are specified relative to the previous ones. The stereoscopic camera model for bundle adjustment presented in Chapter 3 may be attributed to this category. These approaches lack the ability to handle constraints on the cameras and the scene model simultaneously.

Transformation Hierarchies Transformation hierarchies have already proven to be an effective means of scene description in 3D modeling packages and 3D modeling from image applications, such as the work presented by Gibson et al. [56] and van den Hengel et al. [155].

4.3 Constrained Bundle Adjustment

Constraints in bundle adjustment are to date handled in three different ways: weighting schemes, the method of Lagrange multipliers, and reparametrizations. This section will briefly review the method of Lagrange multipliers and reparametrization, before the reparametrization approach in the context of Euclidean transformations is elaborated on in Section 4.4.

Weighting schemes, which try to enforce the constraints through the use of weighted residuals, are often considered to be slow and inexact (see Triggs [152]), and are therefore not elaborated on here.

4.3.1 The Method of Lagrange Multipliers

The Method of Lagrange multipliers is an important tool for constrained optimization. Equality constraints are directly included in the optimization formulation by forming a *Lagrange function* by introducing *Lagrange multipliers* λ. The Lagrange function is then optimized. Further details are given by Nocedal and Wright [116].

Assume that the observation has been made that certain 3D object points of a 3D reconstruction are coplanar in the real world. To enforce this constraint in the 3D reconstruction using the method of Lagrange multipliers, the mathematical formulation of the plane has to be included in the objective function. A plane can be expressed using its unit normal vector $(N_x, N_y, N_z)^\top$ and its distance to the origin d, which form a 4-dimensional vector $\mathbf{g} = (N_x, N_y, N_z, -d)^\top$. The distance of an arbitrary 3D point $\mathbf{X}$ expressed in homogenous coordinates from the plane is then given by the expression $\mathbf{g}^\top\mathbf{X}$, which evaluates to zero if the point is lying in the plane. To enforce this constraint during optimization, the method of Lagrange multipliers augments Equation (2.43) to create the Lagrange function

$$\underset{\mathtt{K},\mathtt{T},\mathbf{X},\mathbf{g},\lambda}{\arg\min} \quad \sum_{j=1}^{J}\sum_{k=1}^{K} \mathrm{d}\Big(\mathbf{x}_{j,k}\,,\, \mathtt{K}_k\mathtt{T}_k^{-1}\mathbf{X}_j\Big)^2 + \sum_{k\in\mathcal{K}} \lambda_k \cdot \mathbf{g}^\top\mathbf{X}_k \quad , \tag{4.1}$$

where $\mathcal{K}$ is the set of points that are coplanar, and λ are the Lagrange multipliers. Note that every point is required to have its own multiplier, i. e., for every constraint introduced an additional parameter has to be added to the optimization procedure. Distortion is omitted from the formulation throughout this chapter for the sake of simplicity.

Optimization of this updated cost function ensures that the refined estimation results respect the given constraint. A detailed description of the sparse structure of this optimization problem and an overview of how to minimize the corresponding cost function is given by McLauchlan et al. [102].

4.3.2 Reparametrization

Again, consider some of the reconstructed 3D object points to be coplanar. Given that a point in 3D space has three *degrees of freedom (DOF)*, it is immediately obvious that the scene model is overparametrized, as soon as only a few points are coplanar. As a plane has three DOF (two for an angular representation of the normal, and one for the distance from the origin), overparametrization occurs as soon as more than three points are constrained. Reparametrization seeks to reduce the inherent overparametrization by expressing the scene structure in a way that has a lower, more appropriate number of DOF. The cost function in this case could be expressed as

$$\underset{\mathtt{K},\mathtt{T},\mathbf{X},\bar{\mathbf{X}},\mathbf{g}}{\arg\min} \quad \sum_{j=1}^{J}\sum_{k\notin\mathcal{K}} \mathrm{d}\Big(\mathbf{x}_{j,k}\,,\,\mathtt{K}_j\mathtt{T}_j^{-1}\mathbf{X}_k\Big)^2 + \sum_{j=1}^{J}\sum_{k\in\mathcal{K}} \mathrm{d}\Big(\mathbf{x}_{j,k}\,,\,\mathtt{K}_j\mathtt{T}_j^{-1}\,\mathrm{f}\Big(\mathbf{g}\,,\,\bar{\mathbf{X}}_k\Big)\Big)^2 \quad , \tag{4.2}$$

where $\mathrm{f}(\mathbf{g}, \bar{\mathbf{X}}_k)$ is a function that appropriately constructs the 3D position $\mathbf{X}$ of a point from its position $\bar{\mathbf{X}}$ on the plane and the plane parameters $\mathbf{g}$. The position $\bar{\mathbf{X}}$ on the plane has only two DOF. Since the points are directly parametrized on the plane, the constraints are always exactly fulfilled. Note that although a plane in 3D may be uniquely specified by three parameters, a consistent convention for the orientation of the plane has to be adopted and taken into account for the function f.

4.4 Reparametrization using Euclidean Transformations

This section introduces the generalized reparametrization approach based on hierarchies of Euclidean transformations that allows the problem of constrained bundle adjustment to be tackled with ease and flexibility.

Euclidean Transformations Recalling Section 2.3.2, a Euclidean transformation may be represented by a 4×4 matrix $\mathtt{T}$ composed of a rotational component, the 3×3 rotation matrix $\mathtt{R}$, and a translational component, the 3-vector $\mathbf{t}$:

$$\mathtt{T} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad . \tag{4.3}$$

Left-multiplication to a point $\mathbf{X}$ applies rotation and translation in this order. In the process, the point is transformed from the local coordinate system to the superordinate coordinate system – which can either be the local coordinate system of another transformation, or the global coordinate system (world space).

Consider again the example of several coplanar points in the scene. In the previous section, a function f was required to convert the 2D plane coordinates of the point with the plane parameters to 3D space. Instead of specifying this function, the corresponding plane may also be described in terms of a Euclidean transformation $\mathtt{T}$. Each 3D object point $\mathbf{X}$ is replaced by its counterpart $\bar{\mathbf{X}} = (\bar{X}, \bar{Y}, 0, 1)^\top$, which has only two DOF. The relation $\mathbf{X} = \mathtt{T}\bar{\mathbf{X}}$ then transfers points from the local coordinate system, i.e., the plane, to the global coordinate system. The cost function can then be given as

$$\underset{\mathtt{P},\mathbf{X},\bar{\mathbf{X}},\mathtt{T}}{\arg\min} \quad \sum_{j=1}^{J}\sum_{k\notin\mathcal{K}} \mathrm{d}(\mathbf{x}_{j,k}, \mathtt{P}_j\mathbf{X}_k)^2 + \sum_{j=1}^{J}\sum_{k\in\mathcal{K}} \mathrm{d}\Big(\mathbf{x}_{j,k}, \mathtt{P}_j\mathtt{T}\bar{\mathbf{X}}_k\Big)^2 \quad . \tag{4.4}$$

Representation A plane in 3D space is completely defined by 3 parameters, 2 for the direction of the plane normal in polar coordinates and 1 for the distance to the origin. Representing a plane by a Euclidean transformation with 3 rotational and 3 translational degrees of freedom leads to a redundant parametrization. As a consequence of the redundancy, the Jacobian matrix of the system is rank deficient. For this application this is not a problem in practice: The Levenberg-Marquardt algorithm has a regularizing component and is therefore suited for the solution of such optimization problems. This topic is further discussed in Section 4.7.

Hierarchy Euclidean transformations can be applied in a sequential fashion, in order to establish – and optimize – a hierarchical description of the scene. An example for the creation of a transformation hierarchy can be found in Section 4.5. In fact, a hierarchy of Euclidean transformations has to be used whenever a desired constraint cannot be expressed through a single transformation. Collinearity and coplanarity usually do not require more than one transformation, while perpendicularity and parallelism do – as does the formulation of constraint interdependencies. The transformation hierarchy enforces that the structure of the constraint interdependency network is a tree – albeit one that branches out to two sides. The root of this tree is located in world space, and the leaves are represented through the 3D object points on one side, and the cameras on the other. This ensures that there is a unique path from the corresponding 3D object point to the associated camera for every 2D feature point, which is a requirement for this method to work.

Taking into account the camera matrix decomposition of Equation (2.4), the cost function of Equation (2.30) can be rewritten in a more general form as

$$\underset{\tilde{\mathtt{P}},\bar{\mathbf{X}}}{\arg\min} \quad \sum_{j=1}^{J}\sum_{k=1}^{K} \mathrm{d}\left(\mathbf{x}_{j,k}\,,\,\tilde{\mathtt{P}}_{j,k}\,\bar{\mathbf{X}}_k\right)^2 \quad , \quad \tilde{\mathtt{P}}_{j,k} = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,] \prod_{h\leftarrow\mathfrak{H}_j} \mathtt{T}_h^{-1} \prod_{i\leftarrow\mathfrak{I}_k} \mathtt{T}_i \quad , \qquad (4.5)$$

with $\mathfrak{I}_j$ and $\mathfrak{H}_k$ specifying sequences of Euclidean transformations $\mathtt{T}_i$ that transform the corresponding point from its local coordinate system to world space, and transformations $\mathtt{T}_h^{-1}$ from world space to the local coordinate system of the corresponding camera, respectively. The left arrow ← denotes the selection of the individual transformations in an ordered fashion.

Constraint Types Euclidean transformations can be used to emulate a wide variety of different constraints. As has already been shown, the full transformation can be used as a coplanarity constraint by simply reducing the DOF count per 3D object point by one and setting the appropriate coordinate to zero. Collinearity requires only two parameters for rotation and one parameter per 3D object point.

Orthogonality and double orthogonality can be handled similar to coplanarity by simply placing the corresponding points in the appropriate planes of the local coordinate

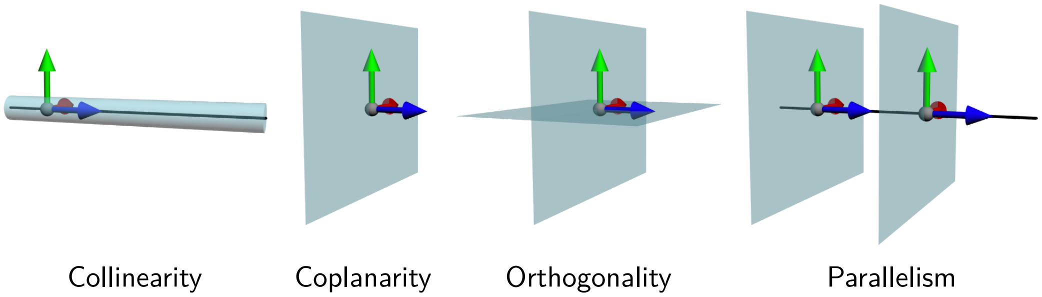


Figure 4.1: Examples for different constraints using Euclidean transformations. The origin and orientation of the coordinate systems associated with the respective transformation is shown, along with the space, in which the constrained points are allowed to move (cyan).

system. For example, two coplanar set of points that are orthogonal to one another could be modeled by moving one set of points to the XY-plane and the other to the XZ-plane of the local coordinate system. Double orthogonality would then consequently use the remaining YZ-plane as well.

Parallelism and arbitrary angular relationships have only one free parameter, but require a sequence of two transformations. For two sets of coplanar points that are parallel to each other, parallelism is modeled by first creating a coplanarity constraint for one set of points. Assuming that the constrained points are located in the XY-plane of the local coordinate system, a second transformation is created, which introduces an offset along the Z-axis as only DOF. The second set of points is then described in the local XY-plane of the second transformation, effectively permitting the two set of points to move closer together or farther apart, provided that the planes they describe stay parallel to each other (see Figure 4.1, far right).

For constraints which require a sequence of two transformations, such as parallelism, the actual constraint is defined in the local coordinate system of its parent.[1] A visualization of sample constraints can be found in Figure 4.1. Table 4.1 lists common constraints and the number of transformations required, including the respective number of DOF.

Moving Objects In a monocular SfM setting, moving objects are represented by estimating an additional set of extrinsic camera parameters per independently moving object. This setup may readily be extended to comprise multiple cameras, but modeling a moving object with one set of extrinsic parameters per camera per frame does not

[1] One could use only one transformation, but in that case the local coordinate system, relative to which the constraint is defined, would be the world coordinate system.

		Degrees of freedom	
Constraint type	**Transformations**	**Rotation**	**Translation**
Coplanarity	1	3	3
Collinearity	1	2	3
Orthogonality	1	3	3
Double orthogonality	1	3	3
Parallelism	2	3+0	3+1
Angular relation	2	3+1	3+0

Table 4.1: Common constraints that can be expressed by Euclidean transformations. For constraints that require two transformations, the number given for the degrees of freedom is the degrees of freedom for the first transformation plus the number of degrees of freedom for the second transformation.

take into consideration that the position and orientation should be the same in all camera images for a given point in time.

Using hierarchies of Euclidean transformations, the position and orientation of a moving object may be specified consistently by a single transformation for each point in time.

Initialization Currently, there is no completely general solution to automatically initialize the constrained bundle adjustment. The initial, unconstrained point cloud is instead transformed to one that respects the desired constraints by using a number of helper functions. On the one hand, these helper functions obtain the transformations associated with specific geometric primitives, such as lines or planes, given a set of 3D object points. On the other hand, they are also responsible for creating the constraint interdependencies, e. g., orthogonality or parallelism structures, in order to ensure that the desired number of DOF is respected. The order in which the helper functions are applied is problem-specific and depends on the desired structure of the transformation hierarchy.

Inaccuracies in the initialization are usually mitigated by the optimization process, so that the only requirement for the initialization is that it lies within reasonable bounds of the specified geometry.

4.4.1 Integration

In Chapter 3, partitions of the parameter vector $\mathbf{p}$ have been discussed for bundle adjustment with joined intrinsic parameters and for bundle adjustment with a stereo camera model. In the latter case, components $\mathbf{a}_2$, $\mathbf{a}_3$, and $\mathbf{a}_4$ have been introduced to accommodate the parameters of the camera model. It may be observed, however, that these parameters obey the same overall structure as the extrinsic camera parameters $\mathbf{a}_0$, albeit with two notable differences: First, the individual components do not need full Euclidean transformations for their description. And second, the parameters are not independent. To combine all these subvectors into $\mathbf{a}_0$, one has to be prepared to increase the size of the matrix $\mathtt{U}$, accept entries for off-diagonal blocks, and to eliminate rows and columns not participating in the optimization before Cholesky decomposition.

One may thus treat hierarchies of Euclidean transformations in bundle adjustment by using a parameter vector

$$\mathbf{p} = \left(\mathbf{a}_0^\top, \mathbf{a}_1^\top, \mathbf{b}^\top\right)^\top \quad , \tag{4.6}$$

with $\mathbf{a}_0$ being the parameters of all Euclidean transformations, $\mathbf{a}_1$ being all intrinsic camera parameters. The matrix $\mathtt{J}^\top\mathtt{J}$ may hence be given as

$$\mathtt{J}^\top\mathtt{J} = \begin{bmatrix} \mathtt{A}_0^\top \\ \mathtt{A}_1^\top \\ \mathtt{B}^\top \end{bmatrix} \begin{bmatrix} \mathtt{A}_0 & \mathtt{A}_1 & \mathtt{B} \end{bmatrix} = \begin{bmatrix} \mathtt{A}_0^\top\mathtt{A}_0 & \mathtt{A}_0^\top\mathtt{A}_1 & \mathtt{A}_0^\top\mathtt{B} \\ \mathtt{A}_1^\top\mathtt{A}_0 & \mathtt{A}_1^\top\mathtt{A}_1 & \mathtt{A}_1^\top\mathtt{B} \\ \mathtt{B}^\top\mathtt{A}_0 & \mathtt{B}^\top\mathtt{A}_1 & \mathtt{B}^\top\mathtt{B} \end{bmatrix} = \begin{bmatrix} \mathtt{U}_{00} & \mathtt{U}_{01} & \mathtt{W}_0 \\ \mathtt{U}_{01}^\top & \mathtt{U}_{11} & \mathtt{W}_1 \\ \mathtt{W}_0^\top & \mathtt{W}_1^\top & \mathtt{V} \end{bmatrix} \tag{4.7}$$

In the presence of distortion, one may introduce an additional element $\mathbf{a}_2$ comprising all distortion parameters into $\mathbf{p}$. This introduces additional blocks in the overall structure, but the overall process remains unaffected.

The requirements outlined above are a concession made to complexity and performance: If all Euclidean transformations are handled the same, regardless of their actual number of degrees of freedom, the description and implementation becomes easier, and the uniformity may be exploited to gain performance increases.

Block Structure For traditional bundle adjustment, the matrix $\mathtt{U}$, which corresponds to the matrix $\mathtt{U}_{00}$ here, was block diagonal (the blocks being of size 6×6 for the number of degrees of freedom of a Euclidean transformation). As mentioned above, this only holds if all Euclidean transformations are independent (implicating that they only describe extrinsic camera parameters and not some sort of constraint). If a more complex camera model or scene constraints are introduced, additional off-diagonal blocks will be created, depending on the structure of the problem. An illustration of this is shown in Figure 4.2.

The structure of the matrix $\mathtt{V}$ is largely unaffected. If 3D object points are constrained, however, the appropriate rows and columns have to be eliminated before inversion.

The block structure of all other matrices depends on the particular scene, ranging from small, dense matrices in case of joined intrinsic parameters for image sequences to large sparse matrices for high variety of intrinsic parameters in the input data.

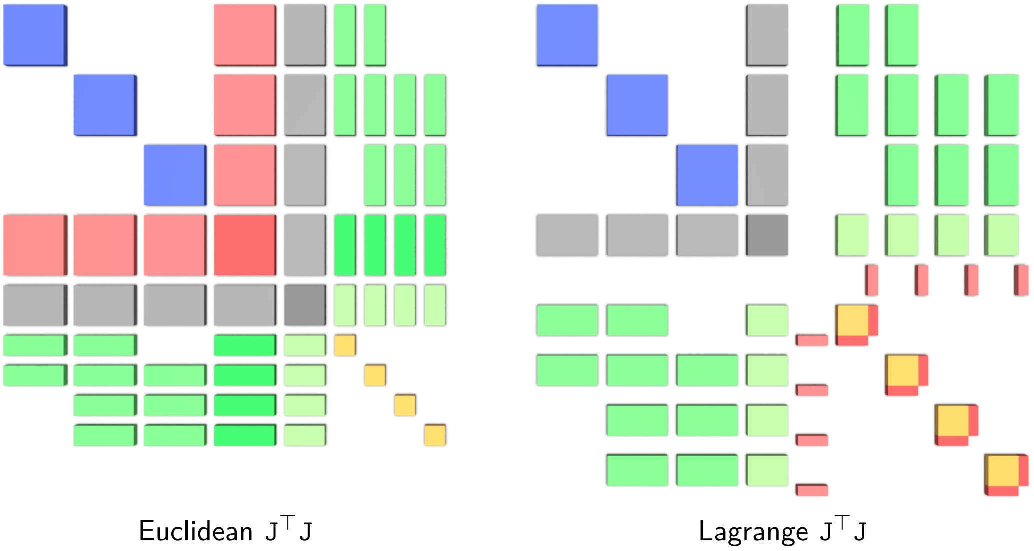


Figure 4.2: Structure of the matrix $J^\top J$ for Euclidean and Lagrange given a toy example with three cameras and four 3D object points that are constrained to be coplanar. The corresponding block structure for unconstrained optimization can be found in Figure 2.7. Blue blocks denote values corresponding to the extrinsic camera parameters, gray to the (shared) intrinsic camera parameters, and yellow to the 3D object point parameters. Red blocks arise from the respective type of constraints. For Euclidean, the 3D object points only have two parameters, whereas the problem size is actually increased for Lagrange.

Implementation To accommodate the updated camera matrices from Equation (4.5), which contain a sequence of camera and point transformations instead of only a single transformation for the unconstrained case, the size of the matrix U_{00} is simply increased. For each Euclidean transformation, an additional row and column of blocks is added. The matrices U_{01} and W_0 are treated in the same way.

Although there are essentially 6 parameters added for each Euclidean transformation, not all constraints require as many (see Table 4.1). However, to simplify the implementation, all Euclidean transformations are assumed to require the full range of parameters. To accommodate this situation, the partial derivatives of translation and rotation parameters that are not part of the constraint description are set to 0. Furthermore, the corresponding rows are eliminated after the construction of $U_{00} - W_0^\top V^{-1} W_0$, before Cholesky factorization. In addition to being easier to implement, this strategy has the benefit of allowing more opportunities for optimization to modern compilers, since all matrices in the individual blocks are of the same, known size.

Consequences Adding the constraints to $\mathtt{U}_{00}$ does not come without consequences. Since there are dependencies between a single constraint and multiple cameras, and between the constraints themselves, off-diagonal entries are created (see Figure 4.2). But the sparse structure of $\mathtt{U}_{00}$ breaks down in any case during the application of the Schur complement trick for the solution of the overall system. For applications relying on dense matrix factorization routines, the impact may be entirely negligible; when sparse factorization routines are used, the impact on the performance may depend on the actual structure of the problem examined. Regardless, this only holds if the number of constraints is small in comparison to the number of cameras. If the number of constraints is large, the increase in computation time may be significant. In such a case, it may be beneficial to reorder the constraints in a way that allows the block-wise inversion of $\mathtt{U}_{00}$, much like it is done with $\mathtt{V}$.

Derivative Calculation The calculation of the derivatives with respect to a single constraint (or rather, Euclidean transformation) can be implemented in a generic way. To this end, for a given sequence of transformations, all transformations preceding the current one are collapsed to a single 3 × 4 matrix, while the effect of all subsequent transformations is collapsed to a single point. The calculation is then done in an iterative fashion over all transformations associated with the current camera and 3D object point. An example for this is provided in the next section.

4.4.2 Example

Assume a point $\mathbf{X}$ to require the following sequence of transformations:

$$\mathbf{x} = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}_1^{-1}\,\mathtt{T}_2\,\mathtt{T}_3\,\mathbf{X} \quad . \tag{4.8}$$

In this case, $\mathtt{T}_1$ most likely represents the position and orientation of the camera, while $\mathtt{T}_2$ and $\mathtt{T}_3$ describe some sort of scene constraints. As an example, assume that the goal is to calculate the partial derivatives with respect to $\mathtt{T}_3$. For these partial derivatives, all other transformations and parameters can be treated as constant, but still have to be taken into account. As a consequence, the left hand side can be collapsed to a single 3 × 4 matrix $\tilde{\mathtt{P}}_3$:

$$\tilde{\mathtt{P}}_3 = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}_1^{-1}\,\mathtt{T}_2 \quad . \tag{4.9}$$

Substitution of this into Equation (4.8) yields

$$\mathbf{x} = \tilde{\mathtt{P}}_3\,\mathtt{T}_3\,\mathbf{X} \quad . \tag{4.10}$$

For the partial derivatives with respect to $\mathtt{T}_2$, the left hand side can again be collapsed to a single camera matrix $\tilde{\mathtt{P}}_2$:

$$\tilde{\mathtt{P}}_2 = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}_1^{-1} \quad . \tag{4.11}$$

By further observing that left-multiplication of $\mathbf{X}$ with a 4×4 matrix yields again a point, but in a different coordinate system,

$$\mathbf{X}_2 = \mathtt{T}_3 \mathbf{X} \tag{4.12}$$

is introduced. This yields

$$\mathbf{x} = \tilde{\mathtt{P}}_2\, \mathtt{T}_2\, \mathbf{X}_2 \quad . \tag{4.13}$$

Finally, $\tilde{\mathtt{P}}_1 = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]$ and $\mathbf{X}_1 = \mathtt{T}_2\, \mathtt{T}_3 \mathbf{X}$ may be evaluated to obtain

$$\mathbf{x} = \tilde{\mathtt{P}}_1\, \mathtt{T}_1^{-1}\, \mathbf{X}_1 \quad . \tag{4.14}$$

Generalization Note that equations (4.10), (4.13), and (4.14) all are similar, except for the use of the inverse transformation in the last equation. The partial derivatives with respect to an Euclidean transformation $\mathtt{T}$ may therefore simply be calculated by taking into account the equation

$$\mathbf{x} = \tilde{\mathtt{P}}\, \mathtt{T}\, \mathbf{X} \tag{4.15}$$

for arbitrary 3×4 matrices $\tilde{\mathtt{P}}$ and vectors $\mathbf{X}$, or, for inverse transformations $\mathtt{T}^{-1}$,

$$\mathbf{x} = \tilde{\mathtt{P}}\, \mathtt{T}^{-1}\, \mathbf{X} \quad . \tag{4.16}$$

4.5 Construction of a Parallelepiped

In this section, several possibilities for the construction of a hierarchy of Euclidean transformations to model a parallelepiped are discussed. Depending on the desired number of DOF, there a different approaches.

The cloud of 3D object points is assumed to be split into distinctive sets to represent the individual faces of the parallelepiped: $\mathbf{X}_a$ for the front face, $\mathbf{X}_b$ for the top face, and so forth.

No Shape Restrictions If the faces of the parallelepiped have arbitrary angular relations, the description through Euclidean transformations may be complex. In addition, it is not unique, as there are usually several ways to arrive at the same solution.

One could start by obtaining the transformation that specifies the frontal face of the parallelepiped, $\mathtt{T}_a$, from the set of 3D object points $\mathbf{X}_a$. The translational component is given by the mean of the point cloud. The rotational component of this transformation can be estimated by performing an SVD on the matrix composed of the mean adjusted points and using the right singular vectors as rotation axes. By applying $\mathtt{T}_a^{-1}$ to $\mathbf{X}_a$ and setting the appropriate coordinate to zero, the 3D object points in the local coordinate system $\bar{\mathbf{X}}_a = (\bar{X}, \bar{Y}, 0, 1)^\top$ are obtained. The corresponding 3D object points in world space are obtained through the relation $\mathtt{T}_a \bar{\mathbf{X}}_a$.

For the 3D object points of the top face, again a Euclidean transformation $\mathtt{T}_b$ can be obtained. Since the constraints to be imposed are not independent, the transformation is specified as $\mathtt{T}_b = \mathtt{T}_a^{-1}\mathtt{T}_b$ and arrive at $\mathtt{T}_a\mathtt{T}_b\bar{\mathbf{X}}_b$ for points in world space. Similarly, the 3D object points in world space of the left side face could be given with $\mathtt{T}_a\mathtt{T}_c\bar{\mathbf{X}}_c$. The transformations $\mathtt{T}_b$ and $\mathtt{T}_c$ given here do not have 6 DOF. Translations are covered by $\mathtt{T}_a$, which leaves only rotations with respect to the front face to be captured by these transformations. Consequently, a single rotational DOF is sufficient. Similarly, the side face on the right is parallel to the one on the left, and thus these points can be expressed by inserting a transformation $\mathtt{T}_d$ with a single, translational DOF into the hierarchy. The transformation sequence is then given by $\mathtt{T}_a\mathtt{T}_c\mathtt{T}_d$.

During the construction of the transformation hierarchy, the spatial arrangement of the transformations is of great importance. Starting with $\mathtt{T}_a$, which describes the frontal face of the cube, if the origin of the subspace specified through this transformation coincides with a top corner, the top and a side face can each be specified by a single subordinate transformation, i. e., $\mathtt{T}_b$, and $\mathtt{T}_c$ or $\mathtt{T}_d$, with the appropriate angle as only DOF. Euler angles as rotation parametrization are a convenient choice in this case, since individual components of the rotation can be constrained rather easily. The other face can either be included through offset transformations with a single, translational DOF, or through more complex relations, depending on the actual shape of the parallelepiped.

Perpendicular Faces – Cuboid A description as complex as the one from the previous section will most likely not be necessary in practice. If the faces of the parallelepiped are perpendicular (i. e., the shape of the object in question is a cuboid), already a single transformation can be used to describe up to three faces. The points of three faces can be given as $(0, \bar{Y}, \bar{Z}, 1)^\top$, $(\bar{X}, 0, \bar{Z}, 1)^\top$, and $(\bar{X}, \bar{Y}, 0, 1)^\top$ in the local coordinate system. A second transformation with three translational DOF can be used to describe the other three faces. In this case, there is no need for any rotational components in the second transformation.

Known Size If the size of the cuboid is known and does not need to be optimized, the model of the previous paragraph may be simplified even further. The second transformation, which was initially required to represent the unknown parameters, can be omitted. The points on the respective planes are given by $(S_x, \bar{Y}, \bar{Z}, 1)^\top$, $(\bar{X}, S_y, \bar{Z}, 1)^\top$, and $(\bar{X}, \bar{Y}, S_z, 1)^\top$, where the components S specify the size of the cuboid.

4.6 Application examples

This section contains three selected application examples and the corresponding results. A summary of these examples is given in Table 4.2. For each application example, a separate table summarizes the results, such as the RMSE after optimization, the

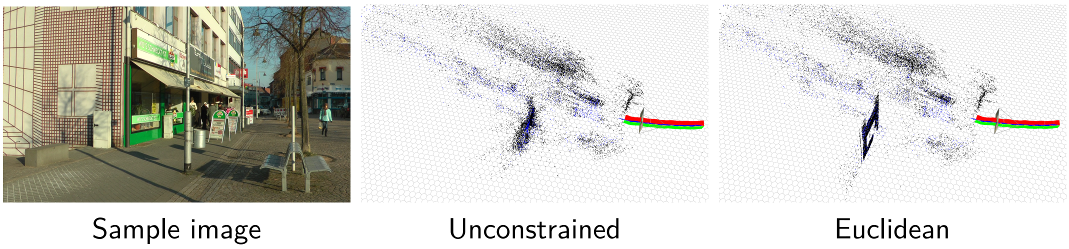
Example	Resolution	Images	Trs.	Cams	2D FP	3D OP	Constrained
Simple	1920 × 1080	705	706	1	1302526	15553	3731
Complex	720 × 576	140	150	1	52012	995	995
Stereo	960 × 540	400	202	2	172427	3356	0 (3356)

Table 4.2: Summary of the data of the image sequences used in the application examples. *Trs.* is the number of Euclidean transformations used in the constrained optimization procedure to model the camera geometry and and scene structure, *2D FP* is the total number of 2D feature points, *3D OP* is the total number of 3D object points, and *Constrained* is the number of 3D object points that were subjected to constraints. The *Stereo* application example applies constraints to the camera geometry that affect all points, but not to the scene description.

number of iterations until convergence, and the average duration per iteration for the Unconstrained optimization, and the corresponding constrained optimizations (Euclidean and Lagrange, if applicable). Shared camera intrinsics over the whole sequence were enforced. All timings are given for a single-threaded implementation running on an Intel Core 2 Quad CPU at 2.83 GHz.

Simple Scene Structure The first sequence consisted of 353 images of a storefront walk-by. A flat wall is prominently featured in the sequence. To evaluate the reconstruction accuracy, all images except the last one were appended to the sequence again in reverse order before processing. The extended sequence consisting of 705 images was then processed using the standard reconstruction pipeline; the images constituting the return path of the camera did not receive special treatment. For the constrained reconstructions, all 3D object points corresponding to 2D feature points detected on the flat wall had constraints placed on them to make them lie on the same plane. The constraint assignment was performed manually. Figure 4.3 shows a sample frame from the image sequence and details of the reconstruction for unconstrained and constrained optimization. It is clearly visible that the coplanarity constraint is respected by the constrained optimization. Constrained optimization was performed with the method of Lagrange multipliers (denoted as Lagrange) and with the new approach using Euclidean transformations (denoted as Euclidean). As the results of Lagrange and Euclidean are visually indistinguishable only the result for Euclidean is given in the figure.

The extension of the sequence allows the evaluation of the accuracy of the result, as the camera positions for the first half of the sequence should be the same as the positions observed in the identical return path. The evaluation of the error between matching camera position pairs between the first and second half of the sequence showed an improvement in the reconstruction accuracy when constraints were used,



Sample image

Unconstrained

Euclidean

Figure 4.3: **Simple scene structure:** A Sample image from the input sequence and detail images of the reconstruction for Unconstrained and Euclidean. The result for Lagrange is visually indistinguishable from Euclidean and therefore omitted.

Method	RMSE [pel]	Iterations	Avg. duration [s]	Position RMSE [%]
Unconstrained	1.20	88	54.99	0.20
Lagrange	1.26	90	87.12	0.14
Euclidean	1.26	90	56.78	0.14

Table 4.3: **Simple scene structure:** Summary of the reconstruction RMSE, the number of iterations, and the average time per iteration. The column *Position RMSE* additionally contains the average deviation of matching camera positions between the original camera path and the identical return path. The error is given relative to the length of the respective path, since the scaling factor of the scene is unknown. While the RMSE is slightly increased for the results of the constrained methods, the lower position RMSE indicates that the reconstructions are more accurate.

as can be seen in Table 4.3. As the scale of the scene was not known, the error was measured relative to the overall length of the reconstructed camera paths

Complex Scene Structure In the second sequence, several geometric shapes are arranged on graph paper, providing ground-truth data for the reconstructed scene. The shapes in these scene were constrained by breaking them down into three planes each. The front plane was used as a base transformation for the whole object, leaving the side and top planes as offset transformations. A third object required an additional plane to model points on a plane parallel to the front plane. Feature points not used in the modeling process were eliminated. This constraint configuration is denoted as Euclidean I. For the second optimization, denoted as Euclidean II, the transformations were restricted to orthogonal angular relations, keeping them closer to the true structure of the scene. The results for both experiments are shown in Table 4.4 and Figure 4.4.

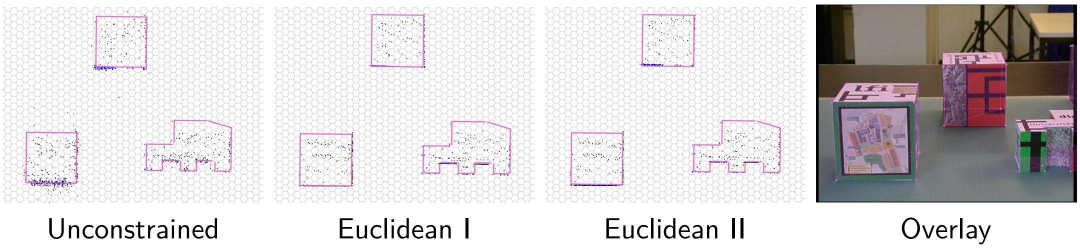


Figure 4.4: **Complex scene structure:** Orthographic detail images of the reconstruction with a ground-truth overlay in pink for Unconstrained and the constrained cases (Euclidean I and Euclidean II), and an image from the input sequence with an Overlay of the ground-truth model in pink, which has been placed with the estimated parameters of Euclidean II.

Method	RMSE [pel]	Iterations	Avg. duration [ms]
Unconstrained	0.60	48	870
Euclidean I	0.79	35	1110
Euclidean II	0.79	35	1112

Table 4.4: **Complex scene structure:** Summary of the reconstruction RMSE, the number of iterations, and the average time per iteration. Due to the additional constraints, the RMSE is slightly increased for Euclidean I and II.

Constrained optimization yields a reconstruction with reduced reconstruction error, as can be evaluated by comparison with the ground-truth overlay.

Stereo Camera Setup The specialized stereoscopic camera model of Chapter 3 can be expressed naturally in the generalized framework presented in this chapter. This provides an example were the camera setup is constrained, as opposed to constraints on the structure of the scene in the previous applications.

For a static stereo setup with negligible distortion, the stereo camera model (Equation (3.3) and Equation (3.4)) may be modeled as

$$\mathtt{P}_{i,\mathrm{L}} = \mathtt{K}_\mathrm{L}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\,\mathtt{T}_\mathrm{L}^{-1}\mathtt{T}_i^{-1} \tag{4.17}$$

$$\mathtt{P}_{i,\mathrm{R}} = \mathtt{K}_\mathrm{R}\,[\,\mathtt{I}\,|\mathbf{0}\,]\,\mathtt{T}_\mathrm{R}^{-1}\mathtt{T}_i^{-1} \quad , \tag{4.18}$$

where the transformation matrices $\mathtt{T}_\mathrm{L}$ and $\mathtt{T}_\mathrm{R}$ contain the additional parameters of the left and right camera with respect to the stereo base frame $\mathtt{T}_i$, as described in Section 3.4.

4.7 Limitations

Increased Matrix Size Conceptually, the proposed method leads to an increase in size of the block containing the transformation derivatives associated with the matrix `U` in the block structure. When compared to traditional SfM, the increase in computational cost for the introduction of a single transformation to model a constraint is equivalent to adding another set of extrinsic camera parameters (e. g., another image added to the sequence). For systems that involve many constraints, the additional cost in terms of execution time may become prohibitive. This is especially true for applications that use a dense matrix factorization approach; for sparse factorization approaches the impact on the performance may be significantly less severe.

If there is a certain structure to the underlying constraints, this structure may be exploited to completely negate the negative effects of the additional transformations on the computation time. When modeling a moving object, for example, there is no dependency between the different constraints, as they just signify temporal instances of the position and orientation of the object, which are completely unrelated for the purpose of optimization. For more complicated constraint interdependency structures, the inherent structures of the optimization problem may also be exploited, but the analysis may become vastly more complex.

Rank Deficiency/Gauge Freedom As mentioned before, the parametrization of certain constraints using Euclidean transformations incurs an overparametrization. For example, a plane in 3D space can be expressed by only 3 parameters, 2 for an angular representation of the normal and 1 for the distance to the origin. The full Euclidean transformation has 6 degrees of freedom – 3 for rotation and 3 for translation. As a consequence of this formulation, it is possible to modify the relative 2D position of constrained points in the plane represented by a Euclidean transformation and the translational parameters of the same transformation in a way that does not affect to actual 3D structure of the scene (e. g., all points are moved along the positive X-axis, but the origin of the local coordinate frame is translated the same distance along the negative X-axis). This is equivalent to gauge freedom, the possibility to arbitrarily choose an intrinsic coordinate frame.

The result of this overparametrization is a rank defect in the Jacobian matrix, which leads to singular normal equations. However, for the results presented in this chapter, no negative effect could be observed. This is due to the fact that the optimization procedure used for the solution of the normal equations (which lead to a singular matrix due to the rank deficiency in the Jacobian) is the Levenberg-Marquardt algorithm, which is intrinsically able to handle overparametrized problems. Levenberg-Marquardt is in fact still predominant in many areas of application of SfM, and thus these effects should not lead to any issues or preclude the usage of this framework in many cases.

The overparametrization could be reduced by identifying and constraining special points for the particular constraints, e. g., a point located in the origin of the local coordinate system (thus having no degrees of freedom) and a point specifying the orientation of the local X-axis (thus having one degree of freedom), which would eliminate the redundant degrees of freedom of the overall system. The further investigation of this topic is left for future work.

4.8 Discussion

A new framework for constrained bundle adjustment has been presented in this chapter. The framework, which is based on hierarchies of Euclidean transformations, provides a flexible and intuitive tool for scene modeling, which can be included into existing bundle adjustment procedures with a minimum of effort, compared to alternative approaches. Furthermore, the unique properties of the Euclidean transformations combined with the arrangement in a hierarchy allow for an elegant handling of constraint interdependencies.

Euclidean transformations can intrinsically handle many important constraints used to date – such as coplanarity, collinearity, angular relations, distances, and parallelism – in a homogeneous manner by constraining specific elements of their rotation and translation components. Special cases like orthogonality and double orthogonality can be represented by a single transformation. In addition, constraints on the scene structure are fully compatible with constraints on the camera geometry.

The proposed method has been compared to the method of Lagrange multipliers, and was found to be comparable, both in terms of accuracy and convergence behavior. In contrast to the method of Lagrange multipliers, however, the new framework effectively reduces the number of parameters in the system, instead of introducing new ones. Another advantage of the approach is uniformity. A 3D object point has at most three parameters, while the method of Lagrange multipliers adds a parameter for every constraint that is attached to a point. The Euclidean transformations themselves also have at most six parameters, in contrast to an unspecified number of parameters, which arises from the constraint itself and the Lagrange multipliers added for all other constraints that influence it. From a software development perspective, this implicates that Euclidean transformations can be integrated with less effort and require little maintenance overhead.

In terms of computation time, it has been shown that it is possible to achieve performance comparable to that of an unconstrained implementation in many cases. This is an improvement over the method of Lagrange multipliers, which usually affects performance.

The convergence behavior of the proposed approach is roughly similar to the corresponding unconstrained optimization, given that the initialization is of sufficient quality. In cases where this cannot be ensured, caution has to be exerted. The same is true for cases where the user-defined scene model does not correctly represent the true scene structure. There is a certain tolerance concerning the wrongful assignment of feature points to constraints, but even if the assignment is correct, outliers in the reconstruction can have negative effects on the result. In some cases, the outliers may cause the whole optimization procedure to break down.

In summary, the presented approach is very versatile and well-suited for the shown application examples.

Future Work The main avenue for future work will be an extensive investigation of the effect of the rank deficiency of the Jacobian matrix on the optimization procedure as well as the investigation of possible different parametrizations for Euclidean transformations to counter these deficiencies on a conceptual level. Different rotation parametrizations and their effects will also be subject of these evaluations.

Research into automatic parameter partitioning methods, similar to the procedures employed by McLauchlan et al. [102], will also be subject of future work, in order to counter the negative effects of a larger number of transformations on the overall computation time. It is also left to future work to investigate the effect of the different constraint interdependencies on the application of sparse solvers.

Other interesting topics for future research would be an in-depth review of the uncertainty propagation and convergence behavior of the hierarchies of constrained Euclidean transformations in bundle adjustment, the automatic or semi-automatic detection of camera and scene constraints, and an investigation of how more general constraints could be included efficiently in the proposed framework.

Part II

Constrained 3D Reconstruction

CHAPTER 5

Global Connectivity Constraints for 3D Line Segment Reconstruction

This chapter describes a novel approach for the probabilistic reconstruction of 3D line segments from images. The approach does not employ explicit line matching across views, but instead performs independent reconstructions while enforcing global topological constraints between neighboring 3D line segments, which are then merged. During the merging process of the partial reconstructions, outliers may be identified and eliminated easily. The proposed method, which is well-suited for the automatic 3D reconstruction of man-made environments, is more robust to image noise and partial occlusions than previous methods relying on explicit line matching across views. This chapter is based on work by Jain et al. [76].

5.1 Introduction and Outline

In many areas, the need for virtual 3D models is ever increasing. The applications are numerous, ranging from movie productions to games and other virtual environments. Creating these 3D models is a complicated and tedious process, however, and a high level of skill and expertise is required from the creator. A different approach is 3D scanning, but the equipment is usually expensive and cumbersome. Hence both approaches lack accessibility.

A viable alternative to manual modeling or 3D scanning is automatic 3D reconstruction from images. SfM, which has been introduced in Chapter 2, is traditionally used to obtain an initial reconstruction of the scene, which may then be processed further with different methods in order to reconstruct 3D models from images. There have been several methods proposed in the literature that estimate detailed 3D models based on the sparse point cloud produced by SfM, such as the work by Gibson et al. [56] or Vogiatzis et al. [157], for example.

3D Line Segment Reconstruction One instance of such an approach relying on prior information obtained through SfM is the reconstruction of 3D line segments from images, which is considered in this chapter. Straight line segments are ubiquitous in man-made environments, indoors and outdoors, and also common in man-made objects. Reconstructions based on 3D line segments are thus well-suited for this kind of setting, including the reconstruction of building exteriors and the creation of urban 3D models. The reconstructed lines may then again serve as a basis for planar reconstruction, as described by Scholze et al.[133], for example. Planar areas are established by first creating a reconstruction of 3D line segments, and then sweeping 3D space to find the best fitting plane.

Traditional formulations for 3D line reconstruction, which will be reviewed in the next section, do not take into account the global topology of the line segments, and hence operate only locally. The matching process for line segments across images is thus complicated by spurious or missing detections due to image noise or partial occlusions. These shortcomings are addressed by the approach presented in this chapter.

Summary of the Approach After the detection of line segments in the input images, the unknown depth parameters of the 3D line segment end points are expressed as random variables. The discrete probability distribution on the different states (i. e., depth values) of the line segment end points are determined using a sweeping-based approach. Line connectivity information between neighboring line segments is then obtained based on the depth values of their respective end points: if the end points in question share the same depth, the line segments are assumed to be connected. This leads to a joint probability distribution of the depth values of the end points of all 3D line segments, in which they are conditioned with the line connectivity information. The joint distribution can be factorized as a graphical model, and *loopy belief propagation* can be used on the corresponding factor graph to obtain the depth values for all segment end points that maximize the joint probability. These depth values yield the globally optimal reconstruction of the 3D line segments.

This process is repeated for all input images, each time yielding a subset of all 3D line segments that comprise the scene. The resulting partial reconstructions are merged, which permits to perform outlier elimination based on the redundancy across the different images.

Contributions In summary, this chapter introduces a probabilistic estimation algorithm for 3D line segments that takes the global topology of line connections into account. The additional constraints for the 3D reconstruction make the algorithm perform better than local approaches, which will be demonstrated in the evaluation. The 3D line segment estimation algorithm uses a sweeping-based approach that does not require explicit line correspondences across views for line segment localization. Solving the correspondence problem for line segment matching, which may be complicated due to image noise and partial occlusions, is thus not required. Finally, an algorithm for line grouping and merging is presented, which allows the combination of the partial reconstructions obtained for the different input images into a global reconstruction. Outlier elimination can easily be performed during this process.

Outline This chapter is organized as follows. After the review of related work in the next section, the mathematical description of the 3D line segment reconstruction problem is given in Section 5.3. An overview over the new probabilistic formulation of the problem can be found in Section 5.4, followed by a structured and detailed description of the individual steps. Section 5.5 covers the evaluation of the algorithm. The chapter is concluded by a description of the limitations of the approach in Section 5.6 and a discussion in Section 5.7.

5.2 Related Work

Research concerning the reconstruction of straight 3D line segments from images can broadly be divided into two categories: epipolar matching and line-based SfM. For epipolar matching, the search space for the matching problem is reduced by exploiting the known epipolar geometry between images. Line-based SfM on the other hand is geared towards the estimation of camera and 3D line segment parameters from known (user-specified) line correspondences.

Epipolar Matching The epipolar beam – the line-equivalent to the epipolar geometry for 2D feature points discussed in Section 2.4.2 – is extensively used to perform line matching across images. Although the epipolar geometry is given, line matching across views is still a difficult task due to the weaker geometric constraints in comparison to point matching.

Baillard et al. [9] use the epipolar beam to establish correspondences between lines in different views by evaluating the normalized cross-correlation scores of line patches. Reconstruction of the 3D line segments is then straightforward by performing the intersection of the half-planes defined by the lines of sight through the end points of the 2D line segments. Moons et al.[113] also use the epipolar geometry to restrict the line matching process to small regions. As their method is geared towards aerial footage, they are able to take flight path information into account for the 3D reconstruction of the line segments. They report difficulties for longer lines that must be matched to more than one shorter line segment in different views. The method by Woo et al.[162] is based on stereo matching, which provides a disparity map for two images. Through the known disparity, the search space for matching candidates is greatly reduced. Heuel and Förstner [67] describe a system that employs geometric reasoning to match line segments through the use of geometric constraints in a probabilistic framework. The method, which takes uncertainty due to measurement noise into account, again starts from constraints arising from the epipolar geometry.

Line-based SfM Similar to Chapter 2, SfM based on line features relies on the projection of the 3D line segments into the images, where it allows the evaluation of a distance-based cost function with the corresponding 2D line segments. The line correspondences these approaches require to work are to date established manually.

The approach by Taylor et al.[144] uses a hybrid optimization to achieve a globally optimal reconstruction of the 3D line segments and camera parameters for given 2D line segment correspondences. Similar SfM formulations have also been presented by Bartoli and Sturm [11] and Schindler et al.[132], for example. The latter method additionally takes vanishing point information into account, in order to reduce the number of parameters in the optimization procedure. Finally, Martinec et al.[98] use a linear factorization-based approach for the reconstruction.

5.3 Problem Statement

The scene model for the constrained reconstruction of 3D line segments is similar to the one already presented in Section 2.2. The input data again consists of J images I_j, with $j = 1, \ldots, J$, of a scene. The goal is to estimate the configuration of the 3D line segments corresponding to the 2D line segments detected in these images.

Scene Model The set of all K 3D line segments $\mathbf{L}_k$ comprised in the description of the real world scene is denoted as $\mathcal{L}$:

$$\mathcal{L} = \{\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_K\} \quad . \tag{5.1}$$

Each individual 3D line segment $\mathbf{L}_k$ is described by its start and end points $\mathbf{X}_k^s$ and $\mathbf{X}_k^e$. These points are equivalent to the 3D object points $\mathbf{X}$ introduced in Section 2.3:

$$\mathbf{X} \in \mathbb{P}^3 \quad , \quad \mathbf{X} = (X, Y, Z, 1)^\top \quad . \tag{5.2}$$

The set of 3D line segments visible in image I_j is denoted by $\mathcal{L}_j \subseteq \mathcal{L}$. Corresponding to the set $\mathcal{L}_j$, there is a set of 2D line segments $\mathcal{E}_j$:

$$\mathcal{E}_j = \{\mathbf{l}_{j,1}, \mathbf{l}_{j,2}, \ldots, \mathbf{l}_{j,K}\} \quad , \tag{5.3}$$

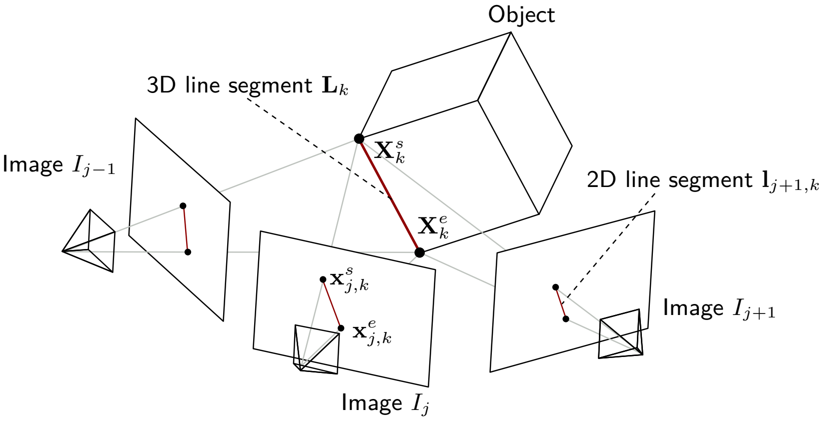


Figure 5.1: 3D line segments $\mathbf{L}_k$ are represented by their start point $\mathbf{X}_k^s$ and end point $\mathbf{X}_k^e$. Their projection into image I_j yields a 2D line segment $\mathbf{l}_{j,k}$ with start point $\mathbf{x}_{j,k}^s$ and end point $\mathbf{x}_{j,k}^e$.

consisting of 2D line segments $\mathbf{l}_{j,k}$. Each individual 2D line segment $\mathbf{l}_{j,k}$ is described by its start and end points $\mathbf{x}_{j,k}^s$ and $\mathbf{x}_{j,k}^e$. These points are equivalent to the 2D feature points $\mathbf{x} \in \mathbb{P}^2$ introduced in Section 2.3. The scene model is illustrated in Figure 5.1.

Projective Mapping If provided with a camera matrix $\mathtt{P}_j$ for each image I_j, the mapping

$$\mathcal{L}_j \mapsto \mathcal{E}_j \quad : \quad \mathbf{l}_{j,k} = \mathtt{P}_j(\mathbf{L}_k) \quad \forall \quad \mathbf{L}_k \in \mathcal{L}_j \quad , \quad \mathbf{l}_{j,k} \in \mathcal{E}_j \tag{5.4}$$

from the 3D line segments to the corresponding 2D line segments in the images is given by the back-projection of the start and end points of the 3D line segments,

$$\mathbf{x}_{j,k}^s \simeq \mathtt{P}_j \mathbf{X}_k^s \quad \text{and} \quad \mathbf{x}_{j,k}^e \simeq \mathtt{P}_j \mathbf{X}_k^e \quad , \tag{5.5}$$

in accordance with Equation (2.3). Note that the effect of distortion is considered to be negligible in this chapter, with distortion thus being omitted from the formulation for simplicity. The back-projected 2D points $\mathbf{x}_{j,k}^s$ and $\mathbf{x}_{j,k}^e$ in Equation (5.5) are not expected to be visible in all views due to occlusions or back-projection outside the image area; they are considered to be virtual points.

Objective The reconstruction approach described in this chapter aims to estimate the set of 3D line segments

$$\mathcal{L} = \bigcup_{j=1}^{J} \mathcal{L}_j \tag{5.6}$$

by leveraging the information represented by sets of detected 2D line segments $\mathcal{E}_j$ and the corresponding camera matrices $\mathtt{P}_j$.

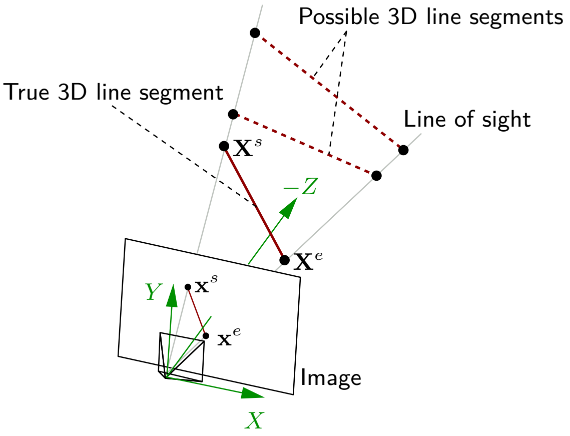


Figure 5.2: The lines of sight through the start and end points $\{\mathbf{x}_j^s, \mathbf{x}_j^e\}$ of the detected 2D line segment restrict the possible positions for the start and end points $\{\mathbf{X}_j^s, \mathbf{X}_j^e\}$ of the corresponding 3D line segment.

Preprocessing The formulation of the problem given above requires a set of detected 2D line segments $\mathcal{E}_j$ and a camera matrix $\mathtt{P}_j$ for every image I_j to be determined in a preprocessing step. The sets of 2D line segments $\mathcal{E}_j$ are established by using a straight line detector based on image gradients obtained with the edge detector proposed by Canny [30]. The camera matrices $\mathtt{P}_j$ corresponding to the images I_j can be obtained by traditional SfM, as described in Chapter 2.

5.4 Probabilistic Reconstruction of 3D Line Segments

The start and end points $\mathbf{x}^s$ and $\mathbf{x}^e$ of a single detected 2D line segment $\mathbf{l}$ restrict the spatial position of the corresponding 3D line segment $\mathbf{L}$: the start and end points of $\mathbf{L}$, $\mathbf{X}^s$ and $\mathbf{X}^e$, have to lie on the line of sight through the corresponding 2D point. This is illustrated in Figure 5.2. It is thus possible to parametrize the 3D points $\mathbf{X}^s$ and $\mathbf{X}^e$, which have only one degree of freedom, by their Z-coordinates, Z^s and Z^e. For a given Z-coordinate, the corresponding X- and Y-coordinates in the local camera coordinate system can be calculated with the help of the 2D positions $\mathbf{x}^s$ or $\mathbf{x}^e$ in the image plane.

Probability Distribution By interpreting Z^s and Z^e as discrete random variables, a probability distribution $p(\mathbf{L})$ over the space of possible orientations for a 3D line segment $\mathbf{L}$ can be defined:

$$p(\mathbf{L}) = p(Z^s, Z^e) \quad . \tag{5.7}$$

Section 5.4.1 describes how this probability distribution may be obtained. The optimal position for the 3D line segment is given by

$$\underset{Z^s, Z^e}{\arg\max} \quad p(Z^s, Z^e) \quad . \tag{5.8}$$

Connectivity The previous paragraphs have modeled the probability distribution of the position of a single 3D line segment. This model will now be extended to cover multiple lines and their connections. Let $\mathcal{A}$ be the set containing the connections between all 3D line segments $\mathbf{L}_k$. This set then describes the global 3D line segment topology of the scene. Connections between 3D line segments $\mathbf{L}_p = \{\mathbf{X}_p^s, \mathbf{X}_p^e\}$ and $\mathbf{L}_q = \{\mathbf{X}_q^s, \mathbf{X}_q^e\}$ are expressed by an equivalence relation for the connected points. This equivalence relation is indicated by $\mathcal{A}$ via

$$\mathcal{A} = \begin{cases} \alpha_{p,q}^{a,b} = 1 & \text{if} \quad \mathbf{X}_p^a = \mathbf{X}_q^b \quad \text{with} \quad a, b \in \{s, e\} \\ \alpha_{p,q}^{a,b} = 0 & \text{else} \end{cases} \quad . \tag{5.9}$$

The size of $\mathcal{A}$ is given by $4A(A-1)$, with A being the total number of 3D line segments in the scene, as start points can be connected to end points and vice versa. The set $\mathcal{A}_j$ is the set of line segment connections corresponding to the topology of $\mathcal{L}_j$ for image I_j. How the initial sets $\mathcal{A}_j$ can be obtained is described in Section 5.4.2.

Joint Probability Distribution Given a set of 3D line segments $\mathcal{L}_j$ and the corresponding set of connections $\mathcal{A}_j$ for an image I_j, the joint probability distribution is given by $p(\mathcal{L}_j|\mathcal{A}_j)$. The globally best position of all 3D line segments $\mathcal{L}_j$ is given by the states at which the random variables Z_k^s and Z_k^e lead to a maximum in the joint probability distribution while taking the global line connectivity information into account:

$$\underset{\mathbf{L}_k}{\arg\max} \quad p(\mathcal{L}_j|\mathcal{A}_j) \quad . \tag{5.10}$$

Solution Finding a solution to the optimization problem of Equation (5.10) directly is NP-hard[1], but the initial line connectivity information $\mathcal{A}_j$ can be used to factorize the joint probability distribution $p(\mathcal{L}_j|\mathcal{A}_j)$ using a graphical model. The max-product for the distribution on the graph can then be found by *loopy belief propagation*, thus optimizing Equation (5.10). After the best positions for the 3D line segments have been found based on the initial connectivity information, the line connectivity may be refined, followed again by the estimation of the best line positions. Line connectivity refinement and line position estimation is alternated until convergence. This process will be elaborated on in Section 5.4.3.

[1] Exact and approximate probabilistic inference and related problems have been shown to be NP-hard on general graphs. More information can be found in the survey by Guo and Hsu [58], for example.

Extension To extend the above procedure to multiple images I_j, an individual reconstruction of the visible subset of 3D line segments $\mathcal{L}_j \subseteq \mathcal{L}$ is first performed for each image. The grouping and merging strategy for the creation of the conjoined set $\mathcal{L}$ given by Equation (5.6) will be discussed in Section 5.4.4. This section will also describe how lines from more than one image may be used to remove outliers.

5.4.1 Line Sweeping

This subsection covers the definition and calculation of the probability distribution $p(\mathbf{L})$ used in Equation (5.8), based on work by Collins [37].

The definition of the probability distribution $p(\mathbf{L})$ is based on the assumption that the cumulative gradient overlap of the back-projection of a 3D line segment $\mathbf{L}$ in an image is proportional to its probability of taking a certain position in 3D space. To calculate the probability $p(\mathbf{L})$, all possible positions of a 3D line segment $\mathbf{L}$ are evaluated by varying the Z-coordinates Z^s and Z^e of the start and end points of the line segment. The coordinates are given in the local camera coordinate system of the image I_j where the corresponding 2D line segment was detected. Each possible 3D line segment is back-projected into the other images $I_{j'}$, with $j' \neq j$, using Equation (5.5). Restrictions on the spatial proximity and camera orientation are enforced to prevent occlusions: if the camera positions for I_j and $I_{j'}$ are too far apart, or if the angle difference between the corresponding principal axes is above 45 degrees, the image $I_{j'}$ is excluded from the probability calculation.

Cumulative Gradient Overlap Evaluation The back-projection of the 3D line segment into image $I_{j'}$ yields 2D start and end points $\mathbf{x}^s_{j'}$ and $\mathbf{x}^e_{j'}$. The enclosed 2D line segment is divided into $M_\mathbf{l}$ equidistant points $\mathbf{y}_m$, $m = 1, \ldots, M_\mathbf{l}$. For every point $\mathbf{y}_m$ created, a set $\mathcal{Y}_m$ of $M_\mathbf{y}$ measurement points perpendicular to the 2D line segment is evaluated on both sides. This configuration is illustrated in Figure 5.3.

Taking all these measurements into account, the probability of the 3D line segment $\mathbf{L}\,(Z^s, Z^e)$ can be evaluated as

$$p(\mathbf{L}) \propto \sum_{j'} \sum_{m=1}^{M_\mathbf{l}} \sum_{\bar{\mathbf{y}} \in \mathcal{Y}_m} \left\| \nabla I_{j'}\left(\bar{\mathbf{y}}\right) \right\| e^{\frac{-\gamma \|\bar{\mathbf{y}} - \mathbf{y}_m\|^2}{\|M_\mathbf{y}\|^2}} \quad , \tag{5.11}$$

where $\nabla I_{j'}\left(\bar{\mathbf{y}}\right)$ is the gradient image corresponding to image $I_{j'}$ evaluated at position $\bar{\mathbf{y}}$, and γ is a scaling factor. The probability values are evaluated for all depth values Z^s and Z^e in order to get the whole distribution $p(\mathbf{L})$. In practice, the Z-values are restricted to lie in a problem-specific range based on the initial 3D reconstruction, in order to reduce the computation time.

The probability distribution, which is calculated for all lines in $\mathcal{L}_j$ for all relevant base image I_j, is sufficient to determine the optimal 3D line segments with Equation (5.8).

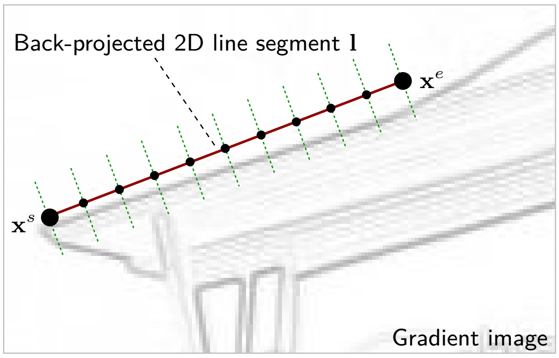


Figure 5.3: Evaluation of the cumulative gradient overlap. The small green dots in direction perpendicular to the back-projected line segment $\mathbf{l}$ represent measurement points where the gradient image is evaluated in order to calculate the overall score.

5.4.2 Connectivity Initialization

This subsection describes how an initial set $\mathcal{A}_j$ of 3D line connections for the factorization of the joint probability in Equation (5.10) can be obtained.

To determine $\mathcal{A}_j$, pairwise connections between the 2D line segments are evaluated. This evaluation is only performed for pairs of line segments that have a start or end point of one segment located within a circular region of a start or end point of the other segment, as illustrated in Figure 5.4, a).

Evaluation The evaluation of a pair of 2D line segments $\{\mathbf{L}_p, \mathbf{L}_q\}$ consists of the calculation of the unconnected cost $U_{p,q}$ and the connected cost $C_{p,q}$. The unconnected cost

$$U_{p,q} = \left(\operatorname*{arg\,min}_{\mathbf{L}_p,\mathbf{L}_q} \quad -\log\left(p(\mathbf{L}_p) \cdot p(\mathbf{L}_q)\right) \right) \tag{5.12}$$

assumes that the line segments $\mathbf{L}_p$ and $\mathbf{L}_q$ are statistically independent, and the probability distribution is determined separately for each segment using Equation (5.11). For the connected cost

$$C_{p,q} = \left(\operatorname*{arg\,min}_{\bar{\mathbf{L}}_p,\bar{\mathbf{L}}_q} \quad -\log\left(p\left(\bar{\mathbf{L}}_p\right) \cdot p\left(\bar{\mathbf{L}}_q\right)\right) \right) - B \tag{5.13}$$

the notation $\bar{\mathbf{L}}$ indicates that the probability distribution of the two 3D line segments is evaluated while the Z-coordinates of the connected start or end points are identical. The scalar B is a user-defined bonus term that encourages line connections. This term is required because the unconnected cost would always be at most as high as

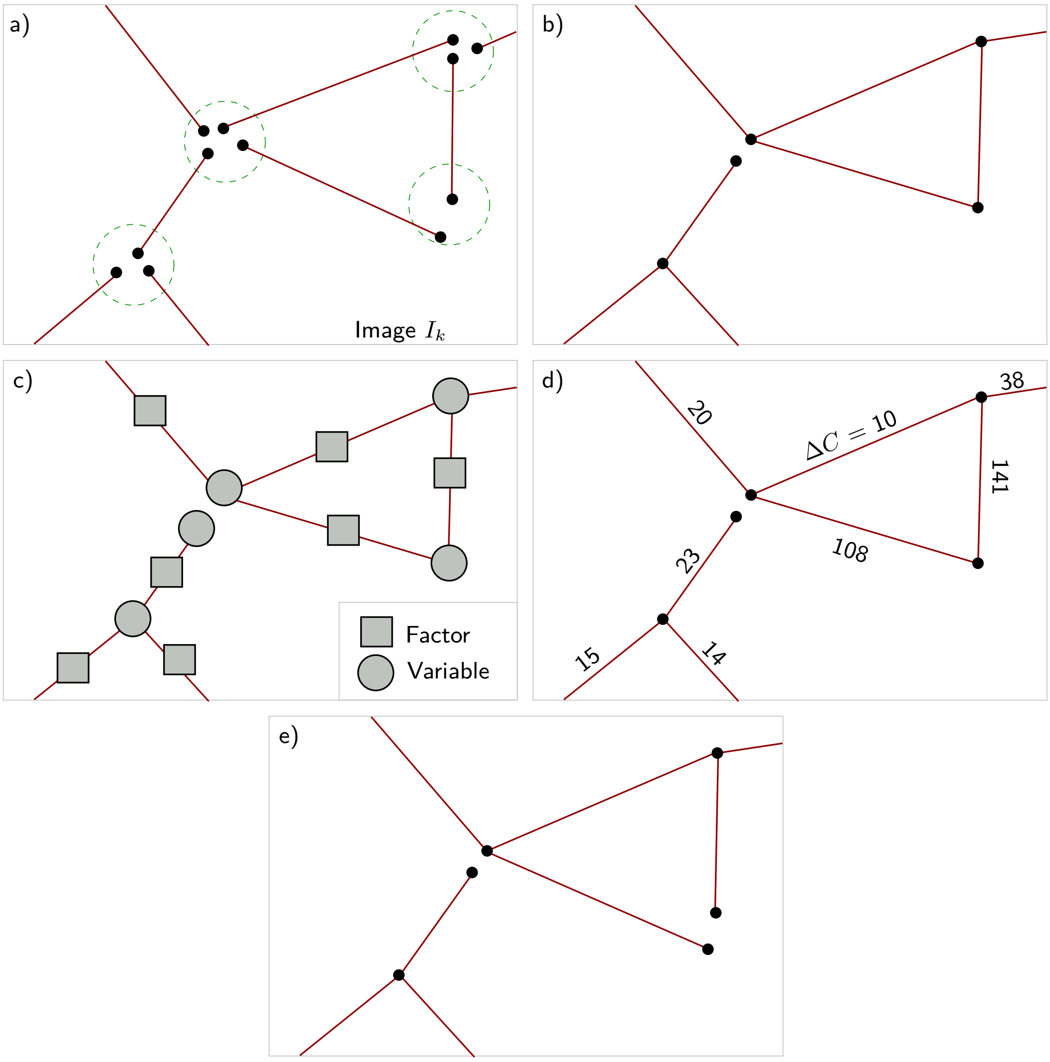


Figure 5.4: Optimization process overview: a) Line segment start and end points are used to create connection candidates based on their proximity. b) Pairwise evaluation of the connected cost $C_{p,q}$ and the unconnected cost $U_{p,q}$ yields the initial line segment connections. c) Factor graph for loopy belief propagation obtained from the initial line connectivity information. d) Calculation of the additional cost of the global connections per line segment. e) If the largest additional cost is larger than a threshold value, the corresponding connection is erased. Steps c) to e) are repeated until convergence.

the connected cost if B was omitted. If $C_{p,q} < Up, q$, the line segments are connected by setting $\alpha_{p,q} = 1$ in the set $\mathcal{A}_j$. If not, the initial value $\alpha_{p,q} = 0$ is not modified. In Figure 5.4, b), a possible connected topology is shown for the example given in Figure 5.4, a).

5.4.3 Belief Propagation and Connectivity Update

The initial line connectivity information $\mathcal{A}_j$ can be used to create a factor graph for loopy belief propagation (an illustration can be found in Figure 5.4, c). The variables of the factor graph are the unknown Z-coordinates of the 3D line segment start and end points. Each factor vertex is connected to two of those variables to create a connection. Taking this into account, the joint probability from Equation (5.10) may be expressed as

$$p(\mathcal{L}_j|\mathcal{A}_j) = \prod_k p\Big(\bar{\mathbf{L}}_k\Big) \quad , \tag{5.14}$$

with $\bar{\mathbf{L}}_k$ being again the 3D line segments with connected start and end points (where appropriate). This means that if start or end points of the line segments are connected, they have to be represented by the same random variable. The probability distributions $p(\mathbf{L})$ are calculated as described in Equation (5.11). The optimization problem that has to be solved is thus

$$\arg\max_{\mathbf{L}_k} \quad \prod_k p\Big(\bar{\mathbf{L}}_k\Big) \quad . \tag{5.15}$$

The optimal positions for the 3D line segments given the prescribed global connectivity can be obtained using loopy belief propagation on the factor graph.

Belief Propagation and Factor Graphs Belief Propagation described by Pearl [122] is a message-passing algorithm for the efficient propagation of the impact of new evidence through Bayesian networks (Pearl [123]). It may be considered as an instance of the sum-product algorithm working on a factor graph, as summarized by Kschischang et al. [88]. They also discuss relevant modifications necessary for the application of the algorithm to Equation (5.15), the problem at hand: The sum-product algorithm, which is initially geared towards calculating the *a posteriori* probabilities of the nodes, can be reformulated into a max-product algorithm to yield the configuration with the largest *a posteriori* probability: summarily speaking, summation during belief update calculation is replaced by the *max* operator. In addition, they also describe methods for coping with loops in the factor graph, thus permitting the calculation of an approximate solution by *loopy* belief propagation.

Both the seminal book by Pearl [123] and the paper by Kschischang et al. [88] contain more detailed descriptions and further information about these topics.

Connectivity Update After belief propagation, the additional cost ΔC_k introduced by the line connectivity is evaluated for every 3D line segment $\mathbf{L}_k$ that has any connections to other segments:

$$\Delta C_k = -\log\left(p\left(\hat{\mathbf{L}}_k\right)\right) - \left(\underset{\mathbf{L}_k}{\arg\min} \quad -\log\left(p(\mathbf{L}_k)\right)\right) \quad , \tag{5.16}$$

where $\hat{\mathbf{L}}_k$ are the optimal global 3D line segments obtained by belief propagation. The subtrahend in the above subtraction is the cost for the hypothetical case that the 3D line segment is unconnected, as given in Equation (5.8). If the highest additional cost ΔC_k is above a user-defined threshold τ, the corresponding connection is erased. Belief propagation is then performed again with the updated factor graph, and the whole procedure is repeated until all ΔC_k are smaller than τ. The positions of the 3D line segments $\mathcal{L}_j$ and the connectivity $\mathcal{A}_j$ are thus optimized conjointly.

5.4.4 Line Grouping and Outlier Elimination

The above process is repeated independently for all images I_j, resulting in independent sets $\mathcal{L}_j$ of 3D line segments. This subsection describes a procedure to group corresponding 3D line segments across images, based on their spatial proximity. Groups of 3D line segments are then merged and replaced by a single 3D line segment, as required by Equation (5.6).

Line Segment Grouping A cylindrical region is defined around each 3D line segment, with the normal of the cross section of the cylinder being orientated in the direction of the line segment. The cylinder is extended in height by 10 percent at the top and bottom past the line segment start and end points, along the direction of the 3D line segment. If both the start and end point of a 3D line segment from another base image are contained in this cylinder, it is grouped together with the one that serves as basis for the cylinder. See Figure 5.5 for an illustration of the grouping procedure.

Line Segment Merging Once all groups have been established, each group is merged into a single 3D line segment. To this end, a new 3D line segment is generated along the principal component direction of the grouped line segments. The principal component direction is obtained as eigenvector corresponding to the largest eigenvalue of the scatter matrix of all start and end points of the grouped line segments. The length of the newly created segment is defined by the maximum extent of the projection of all group points onto the principal component direction, thereby creating a new start and end point at the maximal and minimal value.

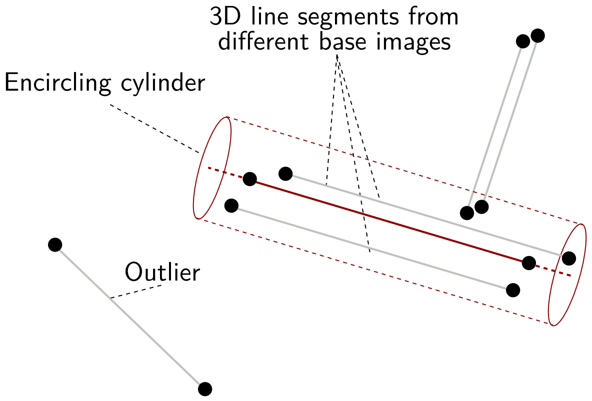


Figure 5.5: For grouping, encircling cylinders are created for every 3D line segment. If 3D line segments from other images are fully contained in a cylinder, the respective segments are grouped together. All 3D line segments in a group are later merged and replaced by a single 3D line segment. 3D line segments not contained in any group are considered to be outliers.

Point Recalculation The established connections between the 3D line segments may be disturbed by the grouping process. To re-enforce the connections, updated start and end point positions $\hat{\mathbf{X}}_k^p$ and $\hat{\mathbf{X}}_k^q$ are estimated by optimizing the cost function

$$\underset{\hat{\mathbf{X}}_k^s, \hat{\mathbf{X}}_k^e}{\arg\min} \quad \sum_{\mathcal{A}} \alpha_{p,q}^{a,b} \left(\left\| \hat{\mathbf{X}}_p^a - \mathbf{X}_q^b \right\|^2 + \left\| \hat{\mathbf{X}}_p^a - \mathbf{X}_p^a \right\|^2 \right) \quad , \tag{5.17}$$

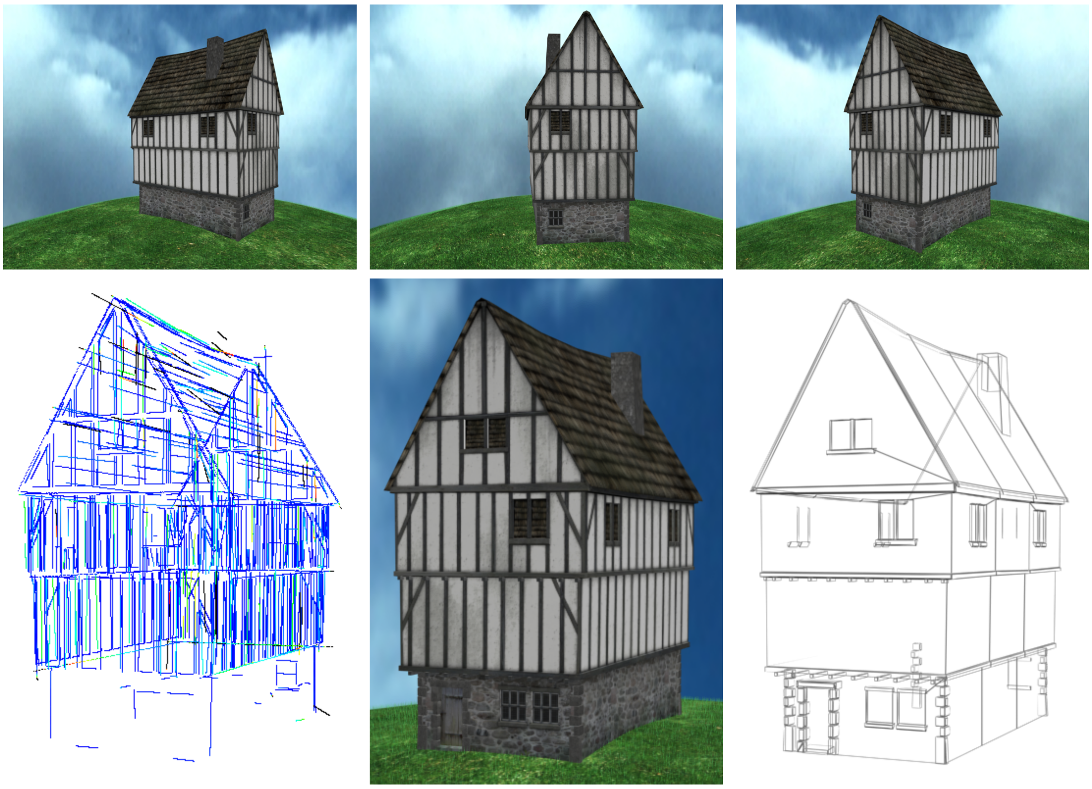
where $\alpha_{p,q}^{a,b} = 1$ capture all connections remaining in the set $\mathcal{A}$.

Outliers All 3D line segments that are not contained in any group are considered to be outliers. These outliers are removed from the final reconstruction.

5.5 Results

In this section, the approach for probabilistic 3D line segment reconstruction with global connectivity constraints presented in this chapter is evaluated. There are 3 data sets presented: a synthetic data set, a data set captured in a lab environment including a laser scan for ground-truth evaluation, and a data set captured outside with a consumer DSLR camera. The reconstruction results for the unconstrained 3D reconstruction are denoted as Unconstrained, while the results for 3D reconstruction with global line connectivity are denoted as Constrained.

In the generation of these results, the software library implemented by Mooij [112] has been used to perform loopy belief propagation.



Constrained Textured Wireframe

Figure 5.6: **Timber-frame house** (rendered scene): Sample images from the data set (top row), results for Constrained, and the corresponding view rendered in Textured and Wireframe mode (bottom row). The line reconstruction is color-coded: blue indicates a low error, red a high error, and black an error larger than 0.5 m with respect to the ground-truth model.

Synthetic Experiments For the synthetic experiments, an image sequence consisting of 240 images with a resolution of 1280 × 960 pixels was generated from a virtual 3D model of a timber-frame house. Sample images from this sequence can be found in Figure 5.6, top row. Table 5.1 shows a comparison of the RMSE values of Unconstrained and Constrained with respect to the ground-truth 3D model. The table lists RMSE values for different cut-off thresholds. Line segments that exhibit error values above this threshold are considered as outliers and the values are not included in the RMSE calculation. Constrained consistently leads to lower RMSE values and thus yields a result of higher accuracy. The result for Constrained is also shown in Figure 5.6, bottom left, where the error values are color-coded. The large majority of the line segments have a very small reconstruction error.

RMSE [m]			
Unconstrained	Constrained	**Threshold [m]**	**Improvement [%]**
0.3361	0.1970	none	41.1
0.2019	0.1810	3.5	10.3
0.1918	0.1736	2.5	9.4
0.1470	0.1262	1.5	14.1
0.0964	0.0807	0.5	16.2

Table 5.1: **Timber-frame house** (rendered scene): RMSE values for Unconstrained and Constrained with respect to a ground-truth 3D scan. The RMSE is given for different error cut-off thresholds. Constrained yields significantly better error values in all cases.

Outlier elimination, as described in Section 5.4.4, is performed for both Unconstrained and Constrained before RMSE calculation, in order to make Unconstrained more competitive. A comparison between the results before and after line grouping and outlier elimination is shown in Figure 5.7.

Lab Experiment: Toy Blocks The second data set was recorded with an HDV camcorder in a lab environment and consists of 84 images at a resolution of 1440×1080 pixels. It depicts red and yellow toy blocks on a planar black-and-white checkerboard. The edge length of the checkerboard squares is 50mm. To create data for ground-truth evaluation, a reconstruction of the scene was also performed with a commercial laser scanner. SfM was used to create the camera matrices, and the laser scan data was aligned using feature points provided by the checkerboard. Sample images from the input data as well as reconstruction results for Unconstrained, Constrained, and the laser scan are shown in Figure 5.8. The RMSE values are listed in Table 5.2. Constrained again significantly outperforms Unconstrained in terms of RMSE and provides excellent reconstruction results, which can also be assessed visually in Figure 5.8. For the lowest error cut-off threshold (5 mm), the improvement is only 3.3 %, but this particular comparison may already be affected by the measurement error of the laser scanner.

Real-world Experiment: Houses The third 3D line reconstruction was performed for a set of 20 photos of semi-detached houses taken with a consumer DSLR camera in an outdoor low-light situation. Due to this, the images exhibit a high degree of pixel noise. Sample images along with the resulting 3D line segment reconstruction with global connectivity constraints are depicted in Figure 5.9. As can be seen, the reconstruction result is excellent and includes many fine details, such as the tiles of the rightmost house.

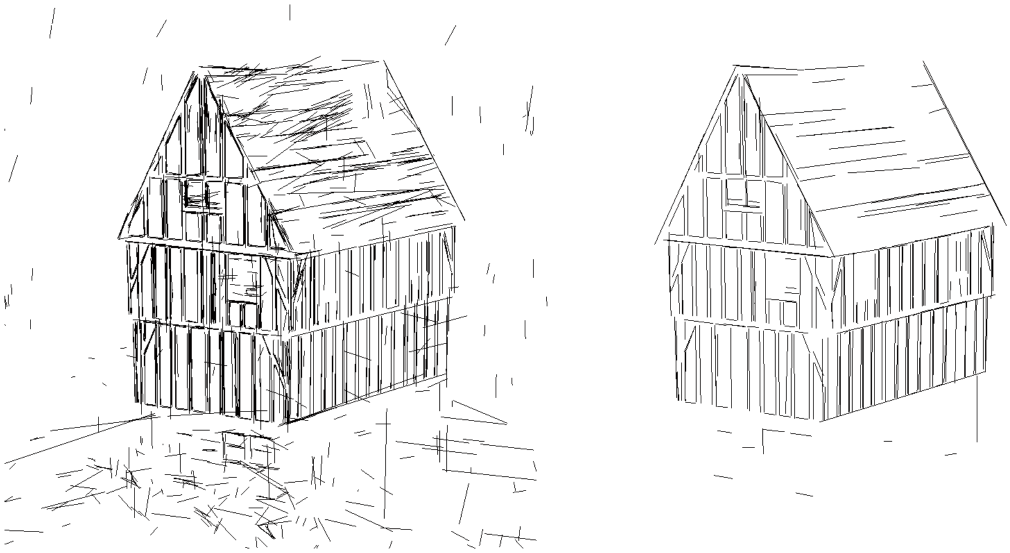
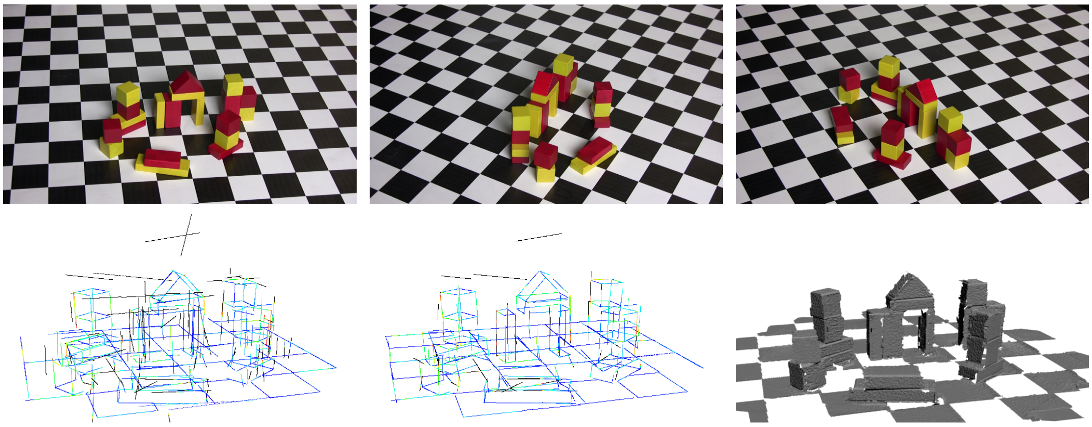


Figure 5.7: **Timber-frame house** (rendered scene): Comparison of a 3D line reconstruction before (left) and after line grouping and outlier elimination (right). Lines corresponding to the back-facing part of the scene are excluded from both visualizations to promote clarity.



Unconstrained

Constrained

Laser scan

Figure 5.8: **Toy Blocks** (lab scene): Sample images from the data set (top row), and results for Unconstrained, Constrained, and the Laser scan used for ground-truth evaluation (bottom row). The line reconstructions are color-coded: blue indicates a low error, red a high error, and black an error larger than 5 mm with respect to the laser scan. Constrained has a very low error overall.

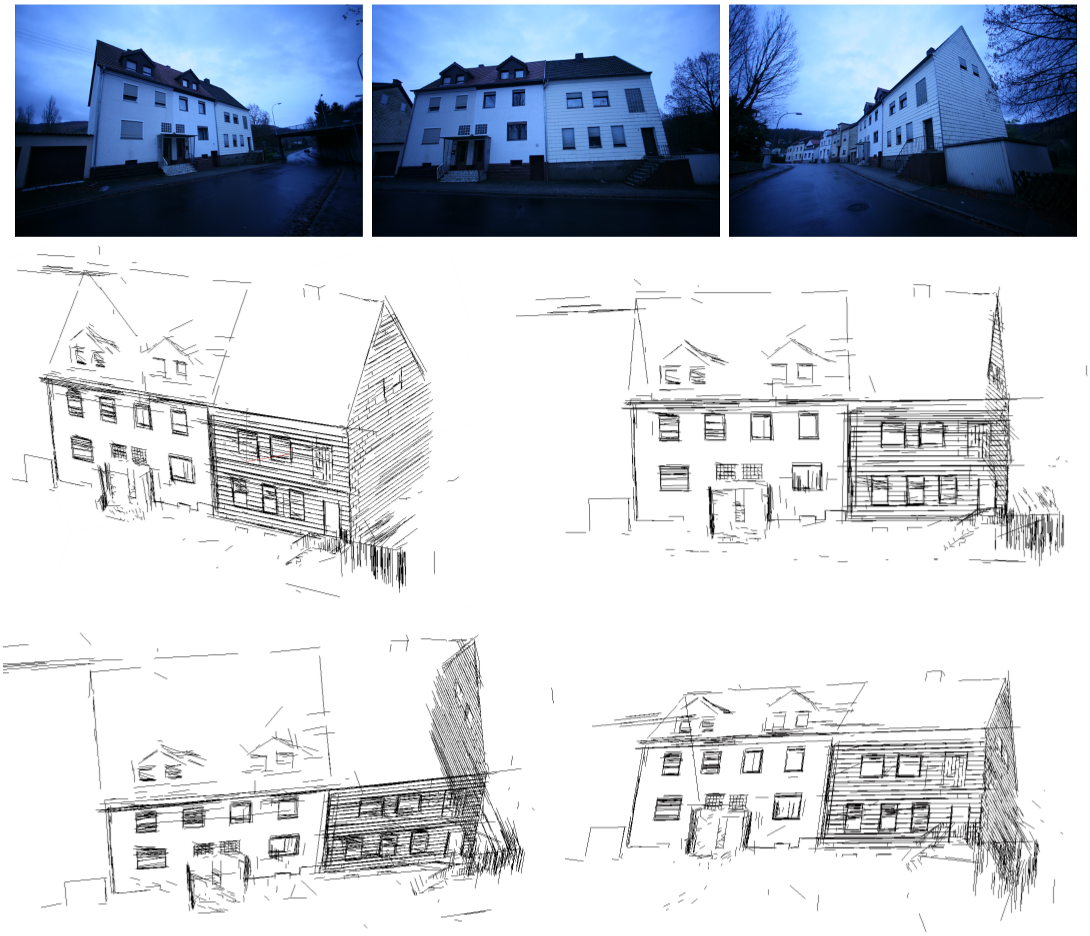


Figure 5.9: **Houses** (real-world scene): Sample images from the data set (top row) and the resulting 3D line segment reconstruction with global connectivity constraints (middle and bottom row). The reconstruction includes many fine details, such as the tiles on the rightmost house.

RMSE [mm]			
Unconstrained	Constrained	**Threshold** [mm]	**Improvement** [%]
10.66	9.65	none	9.4
8.87	6.29	75.0	29.1
7.41	4.30	50.0	42.0
5.48	3.83	25.0	30.1
2.36	2.28	5.0	3.3

Table 5.2: **Toy Blocks** (lab scene): RMSE values for Unconstrained and Constrained with respect to the ground-truth 3D scan. The RMSE is given for different error cut-off thresholds. Constrained yields better error values in all cases. For the lowest threshold value, the result may already be influenced by the measurement error of the 3D scanner.

5.6 Limitations

Computation Time The sweeping-based approach of Section 5.4.1 is computationally more expensive than explicit 2D line matching, as all possible Z-coordinates have to be evaluated. For the examples presented in this chapter, the highest computation time was in the order of 8 to 10 hours. For an even higher number of images, time in excess of that given may be required for computation.

Sweeping Range Another limitation of the sweeping-based approach is that the range in which the Z-coordinates are evaluated is limited in order to reduce the computation time. This implies that 3D line segments that are really located outside this range (typically farther away from the camera than anticipated) cannot be reconstructed correctly.

3D Line Segment Merging During the grouping and merging step of the individual sets of 3D line segments, distinct 3D line segments that lie in close proximity to one another may erroneously be merged together although they should be kept separate.

5.7 Discussion

In this chapter, a novel approach for probabilistic 3D line segment reconstruction from image sequences has been presented. Straight 2D line segments are extracted for all input images, followed by the estimation of an initial reconstruction per image. Based on this initial reconstruction, connections between neighboring 3D line segments are established automatically, which sets this method apart from existing approaches.

Optimization by loopy belief propagation and updates of the connectivity information are then iterated until convergence. The final reconstruction is obtained by grouping and merging the 3D line segments from the individual views, which is also used to perform outlier elimination.

The additional geometric connectivity constraints significantly improve the reconstruction, which has been demonstrated by the evaluation with ground-truth data. The RMSE can by reduced by approximately 20 percent.

Explicit 2D line matching is not needed due to the use of a sweeping-based approach. It is thus possible to reconstruct lines in situations where matching-based approaches would fail, e. g., because the corresponding line segment is not detected in neighboring views due to noise or partial occlusions.

An automatic approach for the merging of partial reconstructions has also been presented in this chapter. 3D line segments are grouped together based on their spatial proximity and then merged and replaced by a single segment. Line segments that are not contained in any group are considered as outliers and removed from the reconstruction. As shown in the results, almost all outliers are eliminated by this approach.

Future Work In the future, additional geometric constraints are a topic for investigation. Such constraints could be the perpendicularity of 3D line segments, for example, which is often present in man-made environments. The reconstruction could also serve as a starting point for closed surface reconstruction algorithms. Another interesting topic would be the inclusion of the camera position and orientation into the probabilistic formulation of the reconstruction problem.

CHAPTER 6

Symmetry-aware Template Deformation and Fitting

This chapter describes a new method for dense 3D reconstruction from low-quality scanning data. The reconstruction of a specific object is performed with the aid of a coarse template model, which is deformed by an *iterative closest points (ICP)* framework with thin-plate-spline regularization. During deformation, the characteristic high-level features of the template are preserved for different variants of the same type of object in order to obtain more plausible reconstructions than previous variants of ICP. The high-level invariants in consideration are the partial symmetry structures of the template models under Euclidean transformations. This chapter is based on work by Kurz et al. [90].

6.1 Introduction and Outline

Nowadays, the creation of virtual content is among the most important aspects in many areas, be it for movie productions, computer games, or other applications, such as virtual museums. Although numerous techniques exist that permit the representation and rendering of increasingly complex scenes, the challenge lies in the fact that the creation of the input data for these techniques – 3D models and similar – is still a tedious task. Further, the creation of high-quality content is a form of art and requires appropriate levels of skill and expertise. These issues with modern content creation have already been identified as a motivation for the development of new algorithms and techniques at several points in this thesis.

Concerning the creation of 3D models of real objects, three approaches can be distinguished in principle: the manual creation of the 3D models, the use of 3D scanning hardware, and the use of pre-existing models.

Manual Modeling The traditional and still most common approach to the creation of virtual objects is modeling from scratch. There are many software packages available to aid in this task, and a lot of research is focused on making the editing process more efficient (see the survey by Mitra et al. [110]). This is achieved by analyzing the object during modeling and detecting properties consistent with a certain *structure model*. Ideally, this structure model captures the common properties of a larger class of similar shapes, and thus facilitates interactive editing by automatically maintaining these properties.

3D Scanning The alternative to manual modeling is the use of 3D scanning hardware or some other sort of visual 3D reconstruction approach. However, this is only a viable approach if the sought virtual model is very similar to a physical object available – and the actual real object is suited for scanning. Another drawback of this method is that it is subject to the issues commonly associated with all optical acquisition methods: noise, occlusions, and structured outliers – there may also be problems caused by the inaccurate registration of partial scans. This is especially true for inexpensive consumer equipment, such as the Microsoft Kinect. As a consequence, there may be a significant amount of manual modeling required before the scanned model can be used for its intended purpose. Aside from cleaning up the reconstruction, many applications require a certain mesh quality and thus necessitate a complete manual re-tessellation.

Pre-existing Models Ready-to-use 3D models are widely available on the internet. There exist numerous websites that will host user-created models, and some services, such as Trimble® 3D Warehouse (formerly Google 3D Warehouse™), even provide facilities for the users to directly create content. Shape libraries containing a selection of professional-grade 3D models, such as *The Archive* by Digimation®, are also available for purchase. It is reasonable to assume that a model at least similar to the object to be scanned can be found, although it may be necessary to further edit the model, depending on the requirements of the application. If the model is of poor quality or not suitably close to the target object, the amount of work required for the editing process may come close to or even exceed that of modeling from scratch.

Shape libraries typically contain many different models per object class, but it is unlikely that they contain the exact same model the user is trying to obtain. Especially for non-trivial classes, such as furniture and household items, the shape space is high dimensional and has many degrees of freedom – exhaustive coverage would require more instances per object class than even a very comprehensive shape library could reasonably provide. This is a consequence of differences in shape *geometry*, though. Finding a model with the exact same geometry is very unlikely. Finding a model with similar *structure*, on the other hand, is much more probable. For many classes of objects, man-made shapes in particular, the object geometry exhibits certain high-level structural invariants pertaining to related functionality. As mentioned above, this has already been exploited in the context of manual editing.

Summary of the Approach The basic observation is thus that there usually exist broad similarities in structure across many instances of objects from a particular class. Based on this observation, a new method for constrained deformation is introduced in this chapter. The method leverages additional structure information to counter common problems encountered during 3D scanning. To this end, a pre-existing, user-selected template model is analyzed and then fitted to scanned data with minimal user interaction. The fitting process is governed by the detected structure priors, which allows the new method to fill acquisition holes and suppress noise and outliers. In addition, the user does not have to rely on the unstructured point cloud or mesh produced by the scanning equipment: the template fitting approach enables the use of a handcrafted, high-quality 3D meshes without manual re-tessellation.

Overview On a conceptual level, the new method represents an instance of structure-aware shape deformation. The structural invariants it is based on are the partial extrinsic Euclidean symmetries of the template shape. These include continuous and discrete symmetries and are automatically detected by leveraging previous work. Once the symmetry analysis has been performed, the template model is deformed iteratively to fit the scanned data using a smooth free-form deformation approach. The *symmetry structure* is maintained during this process, which ensures that the structural relations between parts of the geometry (represented through rigid transformations) are still intact after deformation. To this end, a standard variational thin-plate-spline regularizer is combined with an additional quadratic energy that encourages symmetry preservation. This quadratic energy formulation is co-rotated according to the latent rigid transformation variables.

Contributions The proposed method makes two important conceptually novel contributions: First, it is only based on the very basic assumption that the algebraic structure of the partial Euclidean symmetries of a 3D shape should be preserved. In the formulation, this is implemented by deformation governed by pairwise symmetry transformations. Second, the technique uses symmetry-aware deformation for template fitting to noisy scanner data. This permits high-quality 3D meshes to be created from low-quality 3D scans.

Evaluation For evaluation, a Microsoft Kinect in combination with the Kinect Fusion framework presented by Izadi et al. [75] is used to acquire 3D scans of different objects. The scan quality varies from low-quality partial scans to high-quality full scans. After the acquisition, template models are fitted to the scanned data. The model complexity of the template ranges from very simple to complex. For comparison, the fitting is performed with the proposed method as well as with previous base-line methods, such as deformable ICP and other structure-aware deformation models. In summary, the reconstruction results are more plausible, in particular for partial scans in the presence of noise and missing data. The method thus constitutes a valuable tool for the incorporation of knowledge about the template model's structure into the corresponding 3D scan.

Outline This chapter is organized as follows. After a review of related work in the next section, Section 6.3 gives an overview over the new method. The deformation model is then described in Section 6.4, before a discussion of symmetry and the symmetry detection pipeline in Section 6.5 concludes the introduction of the theoretical foundations. Design choices and implementation strategies are elaborated on in Section 6.6, followed by the presentation of results in Section 6.7. The parameters used are given in Section 6.8. After a review of the limitations of the new approach in Section 6.9, the chapter is concluded by a discussion in Section 6.10.

6.2 Related Work

In this section, previous work concerning structure models and algorithms for preserving these structures under shape alteration is reviewed. The focus is thereby placed on deformation models, database-driven approaches, and template fitting.

Deformation Models Computer graphics has recently seen several methods relying on explicitly constructed basis functions with suitable smoothness properties in order to compute plausible deformations for given shapes, e. g.,*mean value coordinates* by Ju et al. [79], *harmonic coordinates* by Joshi et al. [78], *green coordinates* by Lipman et al. [94], or *variational harmonic maps* by Ben-Chen et al. [13]. Smooth shape deformation may also be accomplished by variational elasticity formulations, such as proposed by Terzopoulos et al. [145] more than two decades ago (a survey of recent advances in that area is given by Botsch and Sorkine [22]).

The particular deformation model used in this chapter is a standard variational thin-plate-spline model, based on the *smoothness error* given by Allen et al. [5] and the *bending energy* formulation in terms of second order partial derivatives by Brown and Rusinkiewicz [25]. This formulation, which aims at general smooth deformations, is subsequently extended to preserve algebraic symmetry structures during deformation.

Structure-aware Deformation A structure-aware shape deformation approach similar in spirit to the seminal *seam carving* approach for image resizing and retargeting by Avidan and Shamir [8] has been presented by Kraevoy et al. [87]. Slippage and curvature analysis of the geometric object to be resized is performed to obtain a *vulnerability map* with respect to stretch in the direction of the three coordinate axis. Consequently, the non-homogeneous resizing operation is then restricted to be axis-aligned. This method has recently been generalized by Bokeloh et al. [18, 19] to incorporate translational symmetry invariants for pattern-aware resizing. Topological changes – insertion and removal of repeating elements – are handled, but the method only supports translational resizing. The use of only translational symmetries is complemented by the approach described in this chapter, as general Euclidean symmetries including rotations and reflections are supported. On the other hand, topological changes are not possible – the novel approach is restricted to the domain of continuous, homeomorphic deformations. However, this allows the formulation in terms of a simple co-rotated least-squares problem; the pattern-aware resizing requires complex discrete optimization.

The image resizing method proposed by Huang et al. [73] and the *iWires* system for mesh editing by Gal et al. [52] also try to preserve a more general set of structural invariants. This is achieved by an analysis that yields global properties like symmetries, parallelism, or vanishing points (for the image resizing approach). Although the approach presented in this chapter is strongly inspired by this previous work, it is solely based on symmetry assumptions. As a consequence, the resulting simple, variational framework can be used for template registration, which is not possible with iWires. While symmetry is covered more completely, other geometric relations, such as parallelism or specific angular relationships, are only represented implicitly for symmetric shapes by the symmetry constraints derived during shape analysis.

Concerning smart deformation tools, Chen and Meng [33] have recently presented an approach for *anisotropic resizing*, which is based on the application of synthesized geometric textures extracted from the original shape. In addition, Xu et al. [164] have introduced a method that performs joint detection on the input model to create a *joint-aware deformation* framework.

Symmetry-based mesh editing techniques rely on symmetry relations for the creation of modification rules: Wang et al. [159] seek to establish the *symmetry hierarchy* of a model; modification of the hierarchy or parameters therein then yields an edited version of the original model. A similar approach based on symmetry analysis is chosen by Zheng et al. [171], who establish a set of component-wise controllers that allow the modification of the original shape.

While all of the previous methods for structure-aware deformation cited above have been applied to user-guided shape deformation, the regularization of deformable shape matching is a novel contribution of the method presented in this chapter.

Editing by Part-based Assembly The analysis of a given example model is used to deduce a set of rules for the part-based assembly of modified or resized models without deformation in the methods described by Merrell [104] and Bokeloh et al. [17]. The assembly may be carried out fully-automatic or with user guidance.

The system introduced by Funkhouser et al. [48] allows a user to compose models by system-assisted cutting and recombining parts from models in a database. Jain et al. [77] proposed a database-driven model decomposition approach that allows blending of several shapes by part-based recombination. Chaudhuri et al. [32] and Kalogerakis et al. [80] have presented a probabilistic model for component-based shape synthesis that allows user editing and database amplification – the further population of the database with models synthesized from models it already contains. Their approach analyzes the relationships between shape components and takes geometric, semantic, and functional relationships between parts into account.

Part-based model synthesis is very different from the continuous deformation approach utilized in this chapter, which currently leaves part-based reconstruction out of scope.

Structure-aware Template Fitting Deformed template meshes may be used to remove artifacts or holes from range scans or similar input data if the template can be fitted appropriately. Kraevoy and Sheffer [86] describe an approach that calculates a mapping of the incomplete input scan to a given template model and then performs mesh completion. A similar, ICP-based system is described by Pauly et al. [120]: deformed variants of models obtained from a database are leveraged for user-assisted scan completion. The methods of Allen et al. [5] and Brown and Rusinkiewicz [25], which have already been discussed above, also have a similar concept; further instances of this approach – the combination of an ICP-formulation with a suitable deformation model – have been presented by Hähnel et al. [60], Wand et al. [158], and Amberg et al. [6].

The template fitting approach for human faces to photographs by Blanz et al. [15], as well as the methods for template fitting to human body scans by Anguelov et al. [7] and Hasler et al. [65] consider only a single class of objects. As a consequence, they are able to exploit model-specific parametric shape spaces, which are low dimensional, to address the overparametrization issues and shortcomings of the smoothness constraints to preserve global structures often exhibited by general deformable ICP approaches.

Xu et al. [163] show how the approach developed by Zheng et al. [171] can be used to fit template models to photographs. The system described by Shen et al. [134] allows low-quality scans to be quickly converted into high-quality 3D models by part assembly. In contrast to the approach presented in this chapter, which allows non-rigid deformation, the model is constructed of individual rigid parts.

Symmetry-based Reconstruction Thrun and Wegbreit [148] have proposed a symmetry-based approach for scan completion for range scans. Partial scans are analyzed to determine the symmetry structures observed in the scanned object, which then allows surfaces occluded during the scan to be synthesized using the available data based on the symmetry information. The approach is fully automatic, but it does not provide facilities for structure-aware editing as described above due to the lack of a template.

Scan Processing An approach to acquire indoor environments from single-view scans using primitive-based 3D models from a separate learning stage has been demonstrated by Kim et al. [83]. In follow-up work [84] they introduce a shape descriptor for user guidance in interactive scanning. These approaches are orthogonal to the algorithm presented in this chapter and could serve in a pre-processing stage, in order to extract suitable input from large-scale data and to select template models automatically.

Surface Reconstruction *GlobFit* by Li et al. [93] augments a local RANSAC-based primitive detection approach for surface reconstruction by enforcing global relations between the primitives. The method presented in this chapter is not limited to models consisting of basic primitives.

6.3 Overview

Input Data The method described in this chapter requires the user to provide a *template model*, denoted as $\mathcal{S} \subset \mathbb{R}^3$. For simplicity, this template model is assumed to take the form of a triangle mesh (of arbitrary topology). However, the generalization of the algorithm to other input representations is straightforward. The user is further required to provide a *target shape* $\mathcal{D} \subset \mathbb{R}^3$, which may commonly be obtained by using a 3D scanner, an SfM-based technique for dense 3D reconstruction, or similar. The target shape is assumed to take the form of an unstructured cloud of 3D points:

$$\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\} \quad , \tag{6.1}$$

with the data points $\mathbf{d}$ being vectors in $\mathbb{R}^3$. Note that the mathematical formulations in this chapter are no longer based on projective space.

Objective The approach described in this chapter aims to estimate an optimal deformation of the 3D surface $\mathcal{S}$ subject to a number of external deformation constraints. In this context, *optimal* means that the high-level structural properties of $\mathcal{S}$ shall be kept intact during deformation. The external deformation constraints may either be manually defined or derived from a set of target data points $\mathcal{D}$. To model the high-level structure, the discrete and continuous symmetries of the template model $\mathcal{S}$ are used.

Symmetry In the context of this thesis, symmetry is defined with respect to a *group of admissible transformations* $\mathcal{G}$. This group $\mathcal{G}$ consists of homeomorphisms – i. e., bijective, in both ways continuous mappings – $\mathfrak{T} : \mathbb{R}^3 \to \mathbb{R}^3$. In this chapter, the group of admissible transformations is restricted to the group of Euclidean transformations: $\mathcal{G} = \mathrm{E}(3)$. It thus comprises translations, rotations, and reflections.

Constrained Deformation Using the provided definition of symmetry, the objective as stated above may be refined. The goal is to deform a template model $\mathcal{S}$ to obtain an output model $\mathfrak{f}(\mathcal{S})$ while keeping the *algebraic symmetry structure* of the template model $\mathcal{S}$ intact. This does not prevent $\mathfrak{f}(\mathcal{S})$ from accommodating very different geometries. It does, however, require the symmetries $\mathfrak{T}_\mathrm{o}$ of $\mathcal{S}$ and their mutual relations to be preserved. For example, if two sub-meshes $\mathcal{P}, \mathcal{Q} \subseteq \mathcal{S}$ are symmetric, so should be the respective sub-meshes of the output model, $\mathfrak{f}(\mathcal{P})$ and $\mathfrak{f}(\mathcal{Q})$. Again, this does not mean that sub-meshes of the output model are bound to the original geometry. They may have very different geometry, and even the transformation $\mathfrak{T}_\mathfrak{f}$ relating them is allowed to differ from $\mathfrak{T}_\mathrm{o}$. An example for relations between symmetries are symmetric parts aligned on a regular grid. Both these types of constraints, direct correspondences and relations, are illustrated in Figure 6.1. To summarize, the term *algebraic symmetry structure* indicates that only the fact that geometry is related by a Euclidean transformation is preserved, but not the concrete mapping itself.

6.4 Deformation Model

The basis of the new method is a smooth free-form deformation model described in the following sections. The description of the representation (Section 6.4.1), the external deformation constraints (Section 6.4.2), and the thin-plate-spline deformation model (Section 6.4.3) is provided for the sake of completeness; the model is not novel and could be substituted with most variational deformation models in the literature. The new additional symmetry constraints will then be described in Section 6.4.4.

6.4.1 Representation

The surface $\mathcal{S}$ is embedded into a bounding volume $\mathcal{V} \subset \mathbb{R}^3$, $\mathcal{S} \subset \mathcal{V}$ to compute the deformation independent from the representation of $\mathcal{S}$. This way, arbitrary types of input geometry and topology can be easily be handled in a homogeneous way. The surface is deformed by a deformation field $\mathfrak{f} : \mathcal{V} \to \mathbb{R}^3$ acting on the bounding volume.

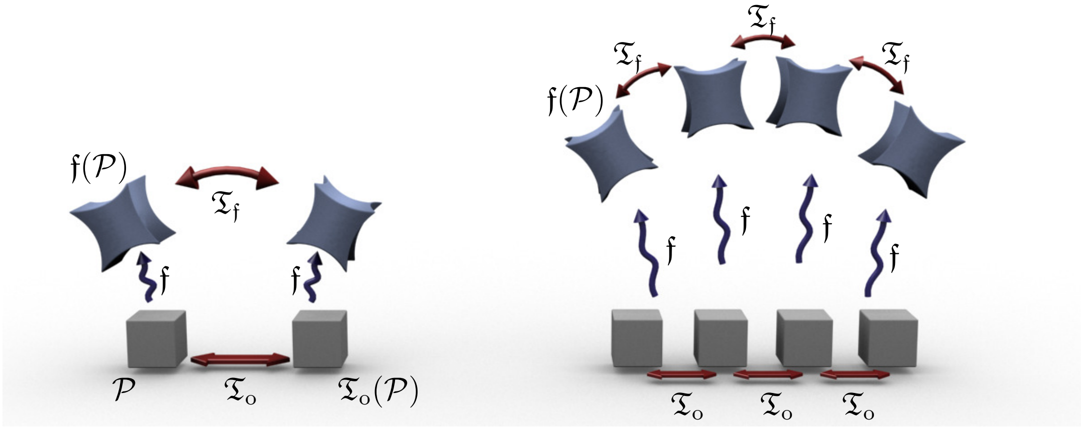


Figure 6.1: Two types of symmetry constraints. The basic constraint (left) assures that sub-meshes deemed symmetric are identical up to a transformation $\mathfrak{T}_\mathfrak{f}$. The relation constraint (right) allows the modeling of regular patterns by sharing a latent transformation $\mathfrak{T}_\mathfrak{f}$.

The deformation field is represented by a number of nodes $\mathbf{u}_1, \ldots, \mathbf{u}_k \subset \mathbb{R}^3$ with associated radial basis functions b centered around them:

$$\mathfrak{f}(\mathbf{z}) = \sum_{i=1}^{K} \tilde{\mathbf{u}}_i \, b(\|\mathbf{z} - \mathbf{u}_i\|) \quad , \tag{6.2}$$

with $\tilde{\mathbf{u}}_i \in \mathbb{R}^3$ being the displaced positions of the nodes $\mathbf{u}_i$. This approach of using a subspace method for discretization is based on the work of Huang et al. [72] and Sumner et al. [140].

Radial Basis Functions Uniform cubic tensor-product B-splines are used as radial basis functions. They provide second order smoothness with minimal support. Experiments with radial basis functions created from Wendland functions have also been performed. The results were visually identical, but the computational cost was considerably higher in comparison to the B-spline basis functions due to the required increase of overlap.

Discretization A regular grid with user-specified spacing ϵ_{grid} is used to discretize the template $\mathcal{S}$. First, nodes $\mathbf{u}_i$ are placed at the corners of all grid cells occupied by $\mathcal{S}$. Additional grid points are then added around the initial points so that every surface point is overlapped by four B-spline functions in X-, Y-, and Z-direction. This is necessary to obtain a valid B-spline basis and guarantees that the basis functions and their derivatives are well defined on $\mathcal{S}$.

Variational Formulation Following a standard variational approach, the deformation field $\mathfrak{f}$ is estimated by setting up an energy function $E(\mathfrak{f})$:

$$E = \omega_c E_c + E_d + \omega_r E_r + \omega_s E_s \quad . \tag{6.3}$$

An optimal $\mathfrak{f}$ minimizes this function. Each term in this formulation models a different aspect: external deformation constraints are described by E_c and E_d (handle constraints and ICP-like constraints, respectively), the thin-plate-spline regularizer that encourages smoothness by E_r, and similarity preservation of symmetric parts and similarity of transformations in regular structures by E_s. Each term is weighted by a parameter (ω_c, ω_r, and ω_s) to control its influence relative to the ICP-like constraints E_d. In addition to the unknown displaced node positions $\tilde{\mathbf{u}}_i$, minimization will later be performed over latent variables[1] that model the transformations.

6.4.2 External Deformation Constraints

Handle Constraints The first energy term E_c accounts for manual user constraints, which are created by using the standard *handle* model described by Bendels et al. [14] and Botsch and Kobbelt [21]. This model imposes a series of *position constraints* $\mathcal{C}_i = (\mathbf{s}_i, \mathbf{c}_i)$ by specifying a one-to-one mapping between an initial point $\mathbf{s}$ on $\mathcal{S}$ and a target point $\mathbf{c}$:

$$E_c(\mathfrak{f}, \mathcal{C}) = \sum_{\mathcal{C}_i \in \mathcal{C}} \|\mathfrak{f}(\mathbf{s}_i) - \mathbf{c}_i\|^2 \quad . \tag{6.4}$$

ICP-like Constraints The data term E_d of Equation (6.3) ensures that the deformation field $\mathfrak{f}$ is formed in a way that makes $\mathcal{S}$ match the target surface $\mathcal{D}$. This is achieved by formulating a series of ICP-like constraints between $\mathcal{S}$ and $\mathcal{D}$, as described by Hähnel et al. [60] and Wand et al. [158]:

$$E_d(\mathfrak{f}, \mathcal{D}) = \sum_{\mathbf{d}_i \in \mathcal{D}} w_i \, \langle \mathfrak{f}(\mathbf{s}_j) - \mathbf{d}_i, \, \mathbf{n}_i \rangle^2 \quad , \tag{6.5}$$

with w being a *weighting factor* that penalizes outliers, $\mathbf{s}$ a sample point on $\mathcal{S}$, and $\mathbf{n}$ the normals corresponding to the points $\mathbf{d}$. The *closest point index* j is selected in a way that makes $\mathbf{s}_j$ the point in $\mathcal{S}$ closest to $\mathbf{d}_i$.

[1] Latent means that the value of the variable is implicitly derived from the context.

6.4.3 Thin-plate-spline Deformation Model

The regularizer term E_r of Equation (6.3) governs the structure of the deformation field where it is underconstrained. To this end, a standard formulation of a thin-plate-spline deformation model is used (see Allen et al. [5] and Brown and Rusinkiewicz [25]), which discourages bending in $\mathcal{S}$:

$$E_r(\mathfrak{f}) = \int_{\mathcal{V}} \|\mathcal{H}_{\mathfrak{f}}(\mathbf{z})\|_{\mathrm{F}}^2 \, \mathrm{d}\mathbf{z} \quad , \tag{6.6}$$

with $\mathcal{H}_{\mathfrak{f}}(\mathbf{z})$ the *Hessian matrix* of the deformation field $\mathfrak{f}$ at position $\mathbf{z}$, and $\|\cdot\|_{\mathrm{F}}$ the Frobenius norm. This energy term encourages smoothness by penalizing second order derivatives.

6.4.4 Preserving Symmetries

The constraint that preserves the shape of symmetric parts of the template $\mathcal{S}$ models that two pieces $\mathcal{P}, \mathcal{Q} \subseteq \mathcal{S}$ are symmetric according to a transformation from $\mathcal{G}$:

$$E_s(\mathfrak{f}, \mathfrak{T}_{\mathfrak{f}} \mid \mathcal{P} \sim \mathfrak{T}_{\mathrm{o}}(\mathcal{P})\,, \mathfrak{T}_{\mathrm{o}}) = \int_{\mathcal{P}} \|\mathfrak{T}_{\mathfrak{f}}(\mathfrak{f}(\mathbf{z})) - \mathfrak{f}(\mathfrak{T}_{\mathrm{o}}(\mathbf{z}))\|^2 \, \mathrm{d}\mathbf{z} \quad . \tag{6.7}$$

In this energy term, only the original transformation $\mathfrak{T}_{\mathrm{o}}$ that maps a fixed piece $\mathcal{P} \subseteq \mathcal{S}$ to $\mathfrak{T}_{\mathrm{o}}(\mathcal{P}) \subseteq \mathcal{S}$ in the original model is known. The deformation $\mathfrak{f}$ and the transformation $\mathfrak{T}_{\mathfrak{f}}$ that maps $\mathfrak{f}(\mathcal{P})$ to $\mathfrak{f}(\mathfrak{T}_{\mathrm{o}}(\mathcal{P}))$ in the deformed model are unknown. The transformation $\mathfrak{T}_{\mathfrak{f}}$ is a *latent* variable – it is only implicitly computed in order to optimize for best symmetry preservation and minimal deformation.

An illustration of this setup is given by the commutative diagram in Figure 6.1, left. The deformation function $\mathfrak{f}$ must commute with any symmetry transformation $\mathfrak{T}$ in the symmetric regions of $\mathcal{S}$ in order to preserve the original symmetries $\mathfrak{T}_{\mathrm{o}}$: the prescribed symmetry has to be provided by the deformation field $\mathfrak{f}$.

For the preservation of the algebraic symmetry structure alone, the original symmetry transformation $\mathfrak{T}_{\mathrm{o}}$ may be replaced by a new, yet-to-be-determined corresponding transformation $\mathfrak{T}_{\mathfrak{f}}$ when it is moved out of the argument of $\mathfrak{f}(\cdot)$. For the example given in Figure 6.1, left, paths $\mathfrak{T}_{\mathfrak{f}} \circ \mathfrak{f}$ and $\mathfrak{f} \circ \mathfrak{T}_{\mathrm{o}}$ must thus lead to identical results. Equation (6.7) penalizes violations of the commutative behavior in a least-squares sense by a transfinite integral constraint over the symmetric domain $\mathcal{P} \subseteq \mathcal{S}$.

If the deformation was restricted to preserve the original symmetry transformations $\mathfrak{T}_{\mathrm{o}}$, i. e., it would have to maintain all original symmetry properties, such as absolute rotation axes and reflection planes in space, relative displacements and rotations, its ability to accommodate the external deformation constraints would be severely limited.

Application The symmetry constraints are used to handle both simple pairwise symmetries and complex patterns. For simple pairwise symmetries, Equation (6.7) is applied as-is to enforce a similar shape. For complex patterns, shared transformation variables $\mathfrak{T}_n$ are used. This models a group of transformations being generated by a small set of generators (see Figure 6.1, right). For example, for N shapes $\mathcal{P}_n$, $n = 1, \ldots, N$ originally aligned on a regular grid, the same transformation $\mathfrak{T}_\mathfrak{f}$ would be used to constrain $\mathfrak{f}(\mathcal{P}_n)$, $n = 1, \ldots, N-1$ to $\mathfrak{f}(\mathcal{P}_{n+1})$. A detailed description of the detection of symmetry groups is given in Section 6.5.1.

Solving the System Equation (6.7) is a quadratic energy under the assumptions that all transformations are known. This permits the linear expressions for the gradient with respect to the displaced node positions $\tilde{\mathbf{u}}_i$ simply to be added to the previously obtained linear system – the linear system is co-rotated with the latent transformations. The solution to the overall system is obtained by applying the *conjugate gradient* method (see Nocedal and Wright [116]).

An iterative approach is used if the transformations $\mathfrak{T}_\mathfrak{f}$ are unknown. The linear system is first solved after performing the initialization as $\mathfrak{T}_\mathfrak{f} = \mathfrak{T}_\mathrm{o}$ and shape matching between $\mathfrak{f}(\mathcal{P})$ and $\mathfrak{f}(\mathcal{Q})$ with the initial deformed template then yields a new estimate for the transformation $\mathfrak{T}_\mathrm{o}$. With the correspondences thus being known through $\mathfrak{f}$, a least-squares optimal affine map may be fitted to them and back-projected to E(3) by a polar decomposition of the linear part. The same principle may be applied for multiple components $\mathcal{P}_i$ corresponding under the same, unknown transformation $\mathfrak{T}_\mathfrak{f}$ by computing the least-squares fit to the difference vectors between all pairs $\{i, i+1\}$ instead of a single pair.

6.5 Symmetry Detection

The symmetry constraint energy terms E_s given in Equation (6.7) are the numerical tool for the expression of both pairwise symmetries between symmetric regions of an object and of relations between individual constraints. The latter is accomplished by using shared latent transformation variables among the E_s terms. In this section, a pipeline for the analysis of the symmetry structure of the template model $\mathcal{S}$ is discussed, which is needed in order to use this tool by generating appropriate symmetry constraints.

6.5.1 Symmetry Structuring

The relevant concepts for the analysis of the symmetry structure, which is formulated in terms of symmetry groups (see Mitra et al.[109]), are introduced in this section.

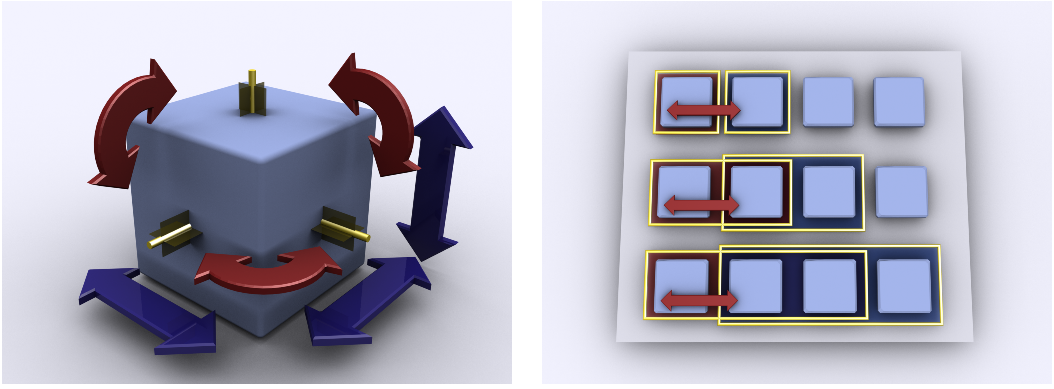


Figure 6.2: Left: A cube as an example of a symmetry group. It is symmetric under rotations by 90 degree (indicated by the red arrows) and mirroring across the main axes (indicated by the blue arrows). The 48 possible configurations form the full octahedral symmetry group (see Miller [105]). Right: Symmetric subsets of a grid of symmetric pieces are mapped to each other by multiple transformations.

Symmetry The set of symmetries extracted from the template model $\mathcal{S}$ with respect to a *symmetry transformation* $\mathfrak{T} \in \mathcal{G}$ is denoted as

$$\xi\left(\mathfrak{T}\right) = \{\mathbf{z} \in \mathcal{S} \mid \mathfrak{T}(\mathbf{z}) \in \mathcal{S}\} \quad . \tag{6.8}$$

The symmetric parts are identified by intersecting the object with a transformed version of itself: $\xi(\mathfrak{T}) = \mathcal{S} \cap \mathfrak{T}\left(\mathcal{S}\right)$. Only results with a large enough area of $\xi(\mathfrak{T})$ are considered in order to avoid spurious matches.

Symmetry Groups For a fixed piece of geometry $\mathcal{P} \subseteq \mathcal{S}$ and a set of transformations $\mathcal{T} \subseteq \mathcal{G}$ that map $\mathcal{P}$ back to $\mathcal{S}$, the geometry created by the application of these transformations to $\mathcal{P}$ is denoted as

$$\mathcal{P}_{\mathcal{T}} = \bigcup_{\mathfrak{T} \in \mathcal{T}} \mathfrak{T}(\mathcal{P}) \quad . \tag{6.9}$$

The 3D object $\mathcal{P}_{\mathcal{T}}$ forms a *symmetry group* if $\mathcal{T}$ is closed under multiplication (i. e., any product of elements of $\mathcal{T}$ is again element of $\mathcal{T}$). This means that $\mathcal{P}_{\mathcal{T}}$ is globally symmetric under the group action of any $\mathfrak{T} \in \mathcal{T}$. The cube shown in Figure 6.2, left, is symmetric under 90 degree rotations and mirroring along all axes, and thus constitutes an example for such a symmetry group.

Full symmetry groups are rarely observed, and a finite translational symmetry group does not even exist (see Figure 6.2, right, for a depiction of excerpts from a symmetry group). To detect the symmetry groups nevertheless, the approach described by

Bokeloh et al. [18] is used: if at least three repetitions of a particular transformation are found, the corresponding observed symmetry group $\mathcal{T}$ is interpreted as a subset of a larger, non-observed proper symmetry group $\mathcal{T}'$.

Euclidean Symmetry Groups Euclidean symmetry is well studied. For the Euclidean group E(3) there is even a full classification of all subgroups available (see Hahn [59]). Broadly, discrete and continuous groups can be distinguished. The discrete groups comprise countable sets of transformations generated by up to three rotations and/or translations with additional involutions (reflections or rotations by 180 degrees). The generators of the continuous groups may include instantaneous motions (see Gelfand and Guibas [54]).

Computation of all pairwise transformations within $\mathcal{S}$ implicitly yields the symmetry groups, as each element of the group contributes to each of the transformations. The explicit computation of all symmetry groups is useful nevertheless, as this additional information may be exploited to only enforce symmetry under the action of the generators of the particular symmetry group using Equation (6.7). The generators are sufficient to preserve the whole symmetry group because of the area overlaps: all additional pairwise transformations are constrained implicitly. To be more precise, the same transformation variable implicitly represents all transformations that generate the same group. This has already been discussed in Section 6.3 and Figure 6.1, right. In the figure, only the mapping of the left three elements to the right three elements is constrained under a single pair of transformations $\mathfrak{T}_o, \mathfrak{T}_f$ for the source and target domain. Furthermore, this avoids weighting issues for the least-squares constraints resulting from the symmetry transformations. If complex group structures were not factored into their minimal set of generators, the large amount of resulting pairwise constraints would introduce a bias towards those constraints in contrast to simple, pairwise symmetries. For example, a large grid or a rotational symmetry of high order would receive a disproportionately higher weight than a single reflective symmetry. In addition, the omission of the additional pairwise constraints during processing also reduces the computation time required for the creation of the system matrix considerably.

6.5.2 Symmetry Detection

The method previously described by Bokeloh et al. [18], which permits the detection of discrete translational and reflective symmetries, constitutes the basis for the symmetry detection algorithm used in this framework. It is extended to also output rotational symmetry groups, which does not make the detection algorithm more challenging on a conceptual level. The detection process is based on edge matching in the triangle mesh, which yields potential generators. These generators are then merged into one-parameter groups, and subsequently into more complex structures.

For the detection of continuous symmetries, the slippage-analysis approach by Gelfand and Guibas [54] is used. The translational slippage properties – collinearity of the mesh edges and coplanarity of the mesh faces – are computed directly on the triangle mesh by comparing normals. The computation of the rotational slippage properties is omitted, as it is unreliable and susceptible to difficult thresholding problems. Detection of the rotational symmetries is thus left to the discrete method. This approach has two important limitations. First, the non-flat edges of the triangulation have to be consistent with the rotational symmetry. This requirement precludes the processing of 3D scanning data as templates by this approach. Second, only cylindrical symmetries will be detected for spheres, consistent meshing assumed.

Symmetry detection is not a contribution of this thesis. There are many different strategies available, such as the methods described by Mitra et al. [108], Gal and Cohen-Or [51], Pauly et al. [121], or Bokeloh et al. [19], which could be applied to this task as well. The system created by Tevs et al.[146] was used to detect the symmetry structure of the template models for the results presented in this chapter.

Post-processing The symmetry detection results are not absolutely reliable, and thus certain measures have to be taken in order to avoid spurious matches. The first measure is to delete groups which form subsets of other detected groups in mostly overlapping mesh regions. The second measure is to discard regions of symmetry with an area below 0.025 area units for a scene scaled to a unit bounding box.

A further issue are ghosting artifacts – areas where discrete symmetries bleed into continuous ones. These are prevented by discarding candidate areas for discrete symmetries that are not enclosed in sharp boundaries. The boundaries are computed by region growing starting from the discrete feature the detection was triggered by. To give an example, a pair of chairs on a ground plane would result in the chairs being detected as symmetric, without including the plane. However, the plane would still be detected as a continuous symmetry. Sample results for symmetry detection for several data sets can be found in Figure 6.3.

6.6 Implementation

The creation of a symmetry-aware deformable ICP algorithm requires the combination of the variational model of Section 6.4 with symmetry information as described in the previous section. This section describes relevant implementation details.

ICP-like Constraints Iterative deformable ICP requires a current estimate of the deformed template to be maintained. A rough manual alignment (scaling included) is used to initialize the process. To create the ICP-like constraints, the closest surface point in the current deformed template shape is computed for each data point. Then, a

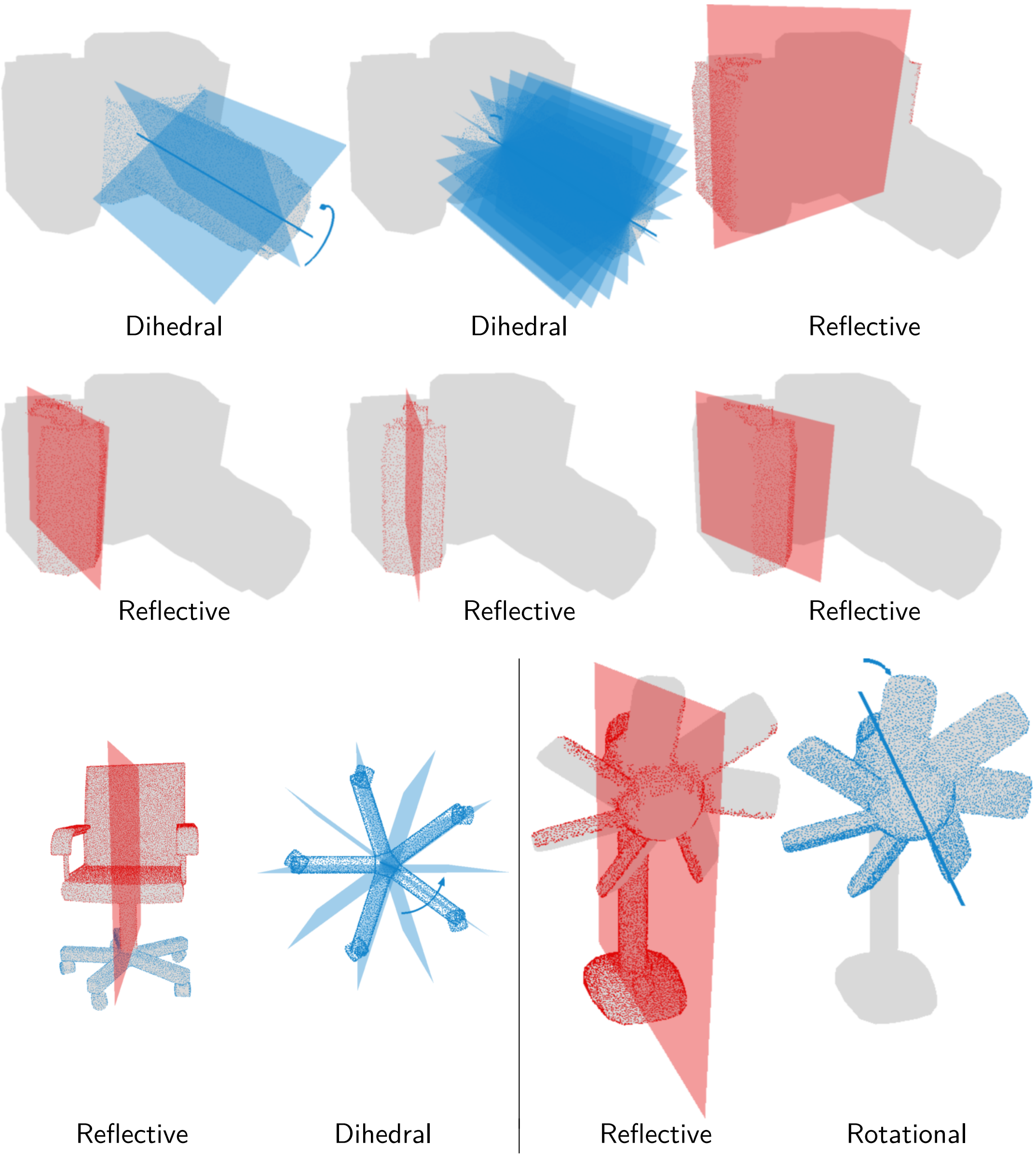


Figure 6.3: Symmetry detection results for the DSLR camera (first and second row), office chair (third row, left), and small fan (third row, right) template meshes. Only the points comprised by the respective symmetry are rendered over the gray model backdrop. The reflection planes of the reflective symmetries are depicted in red. The rotational symmetries are depicted by the rotation axis in blue and a small arrow indicating the rotation. For dihedral symmetries, the reflection planes are also shown in blue.

least-squares point-to-plane attraction constraint is created using Equation (6.5). The 20 nearest neighbors of a surface point are used to estimate the corresponding plane with a PCA-based fit. These constraints originating from the data points of the target shape are denoted as *forward* constraints. Optionally, *backward* constraints may be included in the estimation. For the backward constraints, the role of template model and target shape for the generation of the ICP-like constraints is reversed. Backward constraints should only be used if the target shape is almost complete. If significant portions of the target shape are missing, the algorithm may try to fit the whole template model to the partial structure, which produces highly undesired results. In such cases, backward constraints should therefore not be used.

Pruning for Robustness To make the algorithm more stable, implausible correspondences are pruned during the generation of the ICP-like constraints: First, constraints whose distance $\|\mathfrak{f}(\mathbf{s}) - \mathbf{d}\|$ is above the τ_{d}th percentile of all distances are removed. This parameter has to be supplied by the user in accordance with the amount of outliers expected to be contained in the data. Second, a threshold τ_{nf} is used to determine whether a constraint is located in the near field to avoid oscillation. Constraints located in the near field (i. e., with a distance below τ_{nf}) are not removed, even if their distance lies above the τ_{d}th percentile. The implementation of the ICP-like constraints is kept rather basic, as it is not the focus of this work. Rusinkiewicz and Levoy [129] have presented more sophisticated correspondence filtering methods.

Basis Functions There are two types of basis functions used during the generation of the results. Initially, the iterations use linear basis functions, as this greatly reduces the computation time. To improve the quality of the final result, a final update with smooth B-spline basis functions is performed. In contrast to the linear basis functions, which require $2^3 = 8$ matrix entries per constraint, the B-spline basis functions have a support of four intervals, and thus require $4^3 = 64$ matrix entries per constraint. This significantly increases the computational cost, as may be verified in Table 6.2. Experiments with Wendland functions have also been performed, but their use has been discontinued: they do not provide any improvements in quality over the B-spline basis functions, but are again substantially more costly to compute.

Constraint Sampling The resolution of the discretization grid is typically much lower than the average sampling space of the embedded surfaces. Because of that, the point-based calculation and application of the constraints leads to heavy oversampling. To reduce the overall computation time, all constraints are sampled using a sampling factor $\epsilon_{\mathrm{sampling}}$ chosen below the Nyquist-limit of the discretization grid. Choosing the sampling factor to be $\epsilon_{\mathrm{sampling}} = 0.25\epsilon_{\mathrm{grid}}$ has worked well in practice.

Coplanarity Constraints (Continuous Symmetry) Based on the sets of adjacent planar triangle faces in the template model $\mathcal{S}$, the corresponding sets of points $\mathbf{s}_i \in \mathcal{S}$ are constrained to lie on a plane. A plane is fitted to each set of points and the points are then projected to this surface along the normal direction. To enforce the coplanarity constraints, a sum-of-squares energy similar to Equation (6.5) is formulated for each point and its projection and included in the estimation.

Collinearity Constraints (Continuous Symmetry) The feature lines, which are lists of continuous collinear edges in the template model, are subdivided into directed segments. This is achieved by extracting consecutive constraint points at grid cell intersections. The direction vectors of the individual segments are then enforced to be identical during the computation of the optimal solution.

6.7 Results

To obtain example target surfaces $\mathcal{D}$, scans of everyday objects were created with a Microsoft Kinect and Kinect Fusion. These scans are denoted as Target in the result figures. Parts not related to the original object, like the floor or the background, were removed from the scanned data. This was done to simplify subsequent processing, as the ICP implementation is not in the focus of this work.

For each scanned object a similar 3D model was searched in the shape libraries *The Archive* by Digimation (DSLR camera, TV set, bowl) and in Google 3D Warehouse (all other models except where noted otherwise). The symmetry detection algorithm of Section 6.5 was applied to each model after it had been scaled to unit length. A manual coarse rigid alignment with scaling was then performed as next step. This initial alignment is the starting point for the application of the new algorithm and several other variants of the ICP algorithm, which are used for comparison.

The individual variants of the ICP algorithm applied are Rigid ICP, Affine ICP, Deformable ICP (with thin-plate spline regularization, Equation (6.6)), and the new deformable ICP with additional Symmetry constraints. In order for the results to be comparable, the same implementation was used in each case: Deformable is obtained by switching off the symmetry constraints, and Affine in addition uses a very large weight ω_r such that only affine mappings are obtained. The processing is performed by running 100 ICP iterations with linear basis functions to guarantee convergence, and then an additional single iteration with the smooth basis functions in the case of Deformable and Symmetry. A state-of-the art ICP implementation as described by Mitra et al. [107] was used for Rigid.

Symmetry-aware Fitting The results from the new approach and the base-line methods are illustrated for comparison in figures 6.4–6.18. The figures show that the symmetry constraints are particularly helpful to infer the shape of a man-made object in regions of missing data.

Figures 6.4, 6.5, and 6.6 show examples for the inter- and extrapolation in regions of missing and corrupted data using the detected symmetry information. In Figure 6.4, rows 1 and 2, for example, the scanned data is missing the complete backside of the cooking pot. Unlike Deformable, the results for Symmetry do not show distortions in these regions. The same is visible in Figure 6.4, row 3, which shows a low-quality scan of a pan made of reflective metal, which proved hard to acquire. Implausible deformations are prevented by maintaining the symmetry structure extracted from the template shape.

The result of Symmetry for the cone-shaped cup in Figure 6.6 is distorted by erroneous ICP correspondences, which is a limitation of this approach. In this case, the symmetry structure of the handle, which could have prevented the implausible deformation, could not be determined correctly. The only symmetry detected for the handle was a reflection corresponding to the cross section.

The results for the camera data set are shown in Figure 6.7. For Symmetry, the rotational symmetry of the lens is preserved by rotational symmetry constraints while the continuous symmetry constraints keep the case of the camera in a rectangular shape. This is a considerable improvement over Deformable which exhibits numerous distortions and local deformations. The parameters have to be chosen carefully, though: The weights for the continuous symmetry constraints had to be increased by a factor of 10 for this example. A comparison to the result obtained using the default parameters is included in the figure. Generally, the need to choose parameters is a limitation of the least-squares formulation of the problem.

Traditional Deformable ICP Deformable, which is not guided by the additional symmetry constraints in regions of missing data or poor, corrupted scanning results, is more susceptible to distortions and local deformations. This can be observed in Figure 6.4, rows 1 and 2, for example. The approach may be able to compensate small irregularities (see Figure 6.5, rows 2 and 4). However, the preservation of the symmetry structure leads to more plausible results in addition to providing a close match for the input data. This is illustrated in Figure 6.8, where overlays of the results of the different ICP variants over the scanned data are shown.

Rigid ICP The scaled template is also registered with the data using standard rigid ICP as a sanity check. If the shape of the template is very similar to that of the scanned object, even with this method reasonable results can be produced (e. g., Figure 6.4, rows 1 and 2, and Figure 6.5, row 4). However, if the available template is too different, a good alignment can often not be achieved (e. g., Figure 6.9). A complex example is shown in Figure 6.7: The template is quite different from the input data, but still a

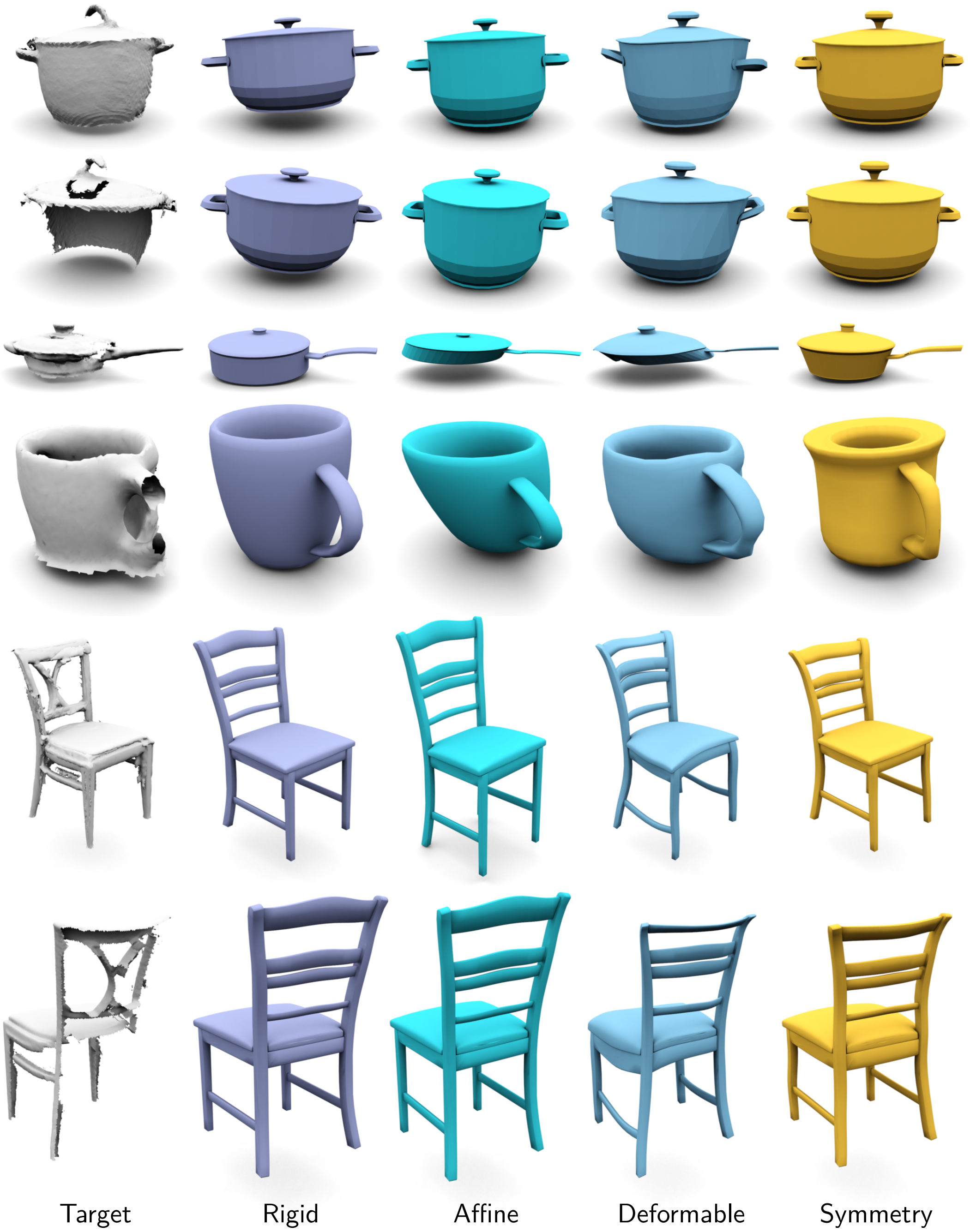


Figure 6.4: Results of different ICP variants. From top to bottom: Cooking pot (single-view, front); the same pot (single-view, back); frying pan (single-view); rounded cup (single-view); chair (full scan); the same chair (single-view).

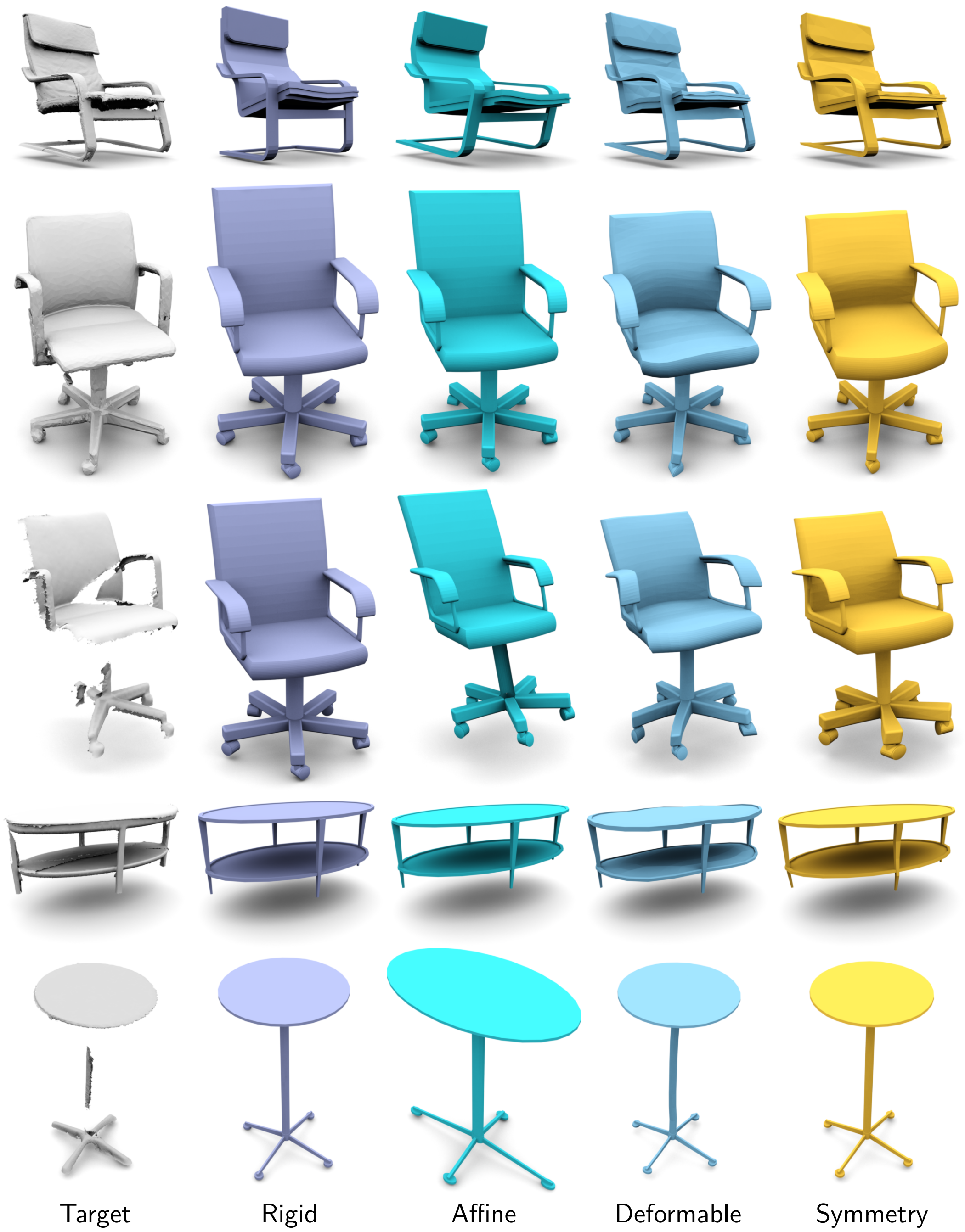


Figure 6.5: Results of different ICP variants. From top to bottom: Armchair (full scan); office chair (full scan); the same office chair (single-view); oval table (full scan); bar table (single-view).

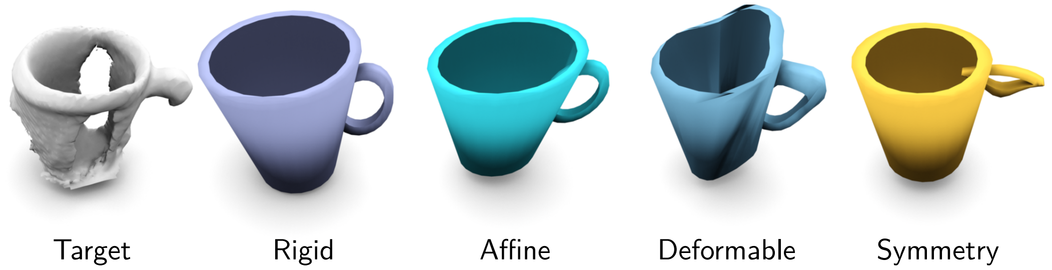


Figure 6.6: Single-view scan of a cone-shaped cup. Symmetry is much closer to Target. Only the handle of the cup shows some distortions. This is because the symmetry structure could not be detected properly.

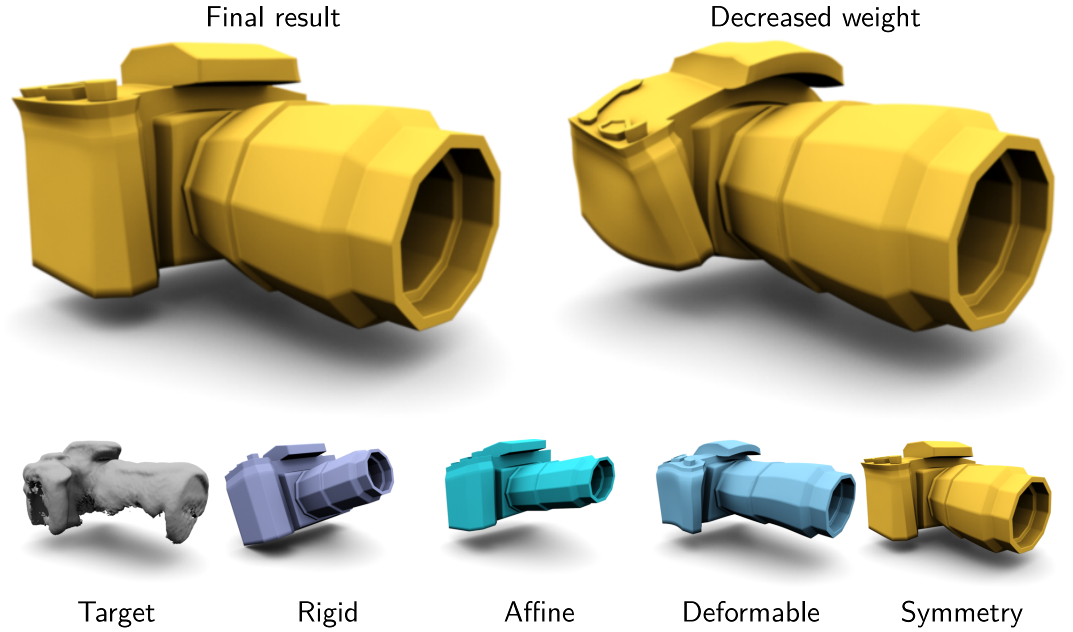


Figure 6.7: Scan of a DSLR camera. There are significant differences in proportion between the actual object and the template model. The rotational symmetry of the lens is preserved by Symmetry. The final result was obtained by increasing the weight of the continuous symmetry constraints. The top right shows the intermediate step with decreased weights.

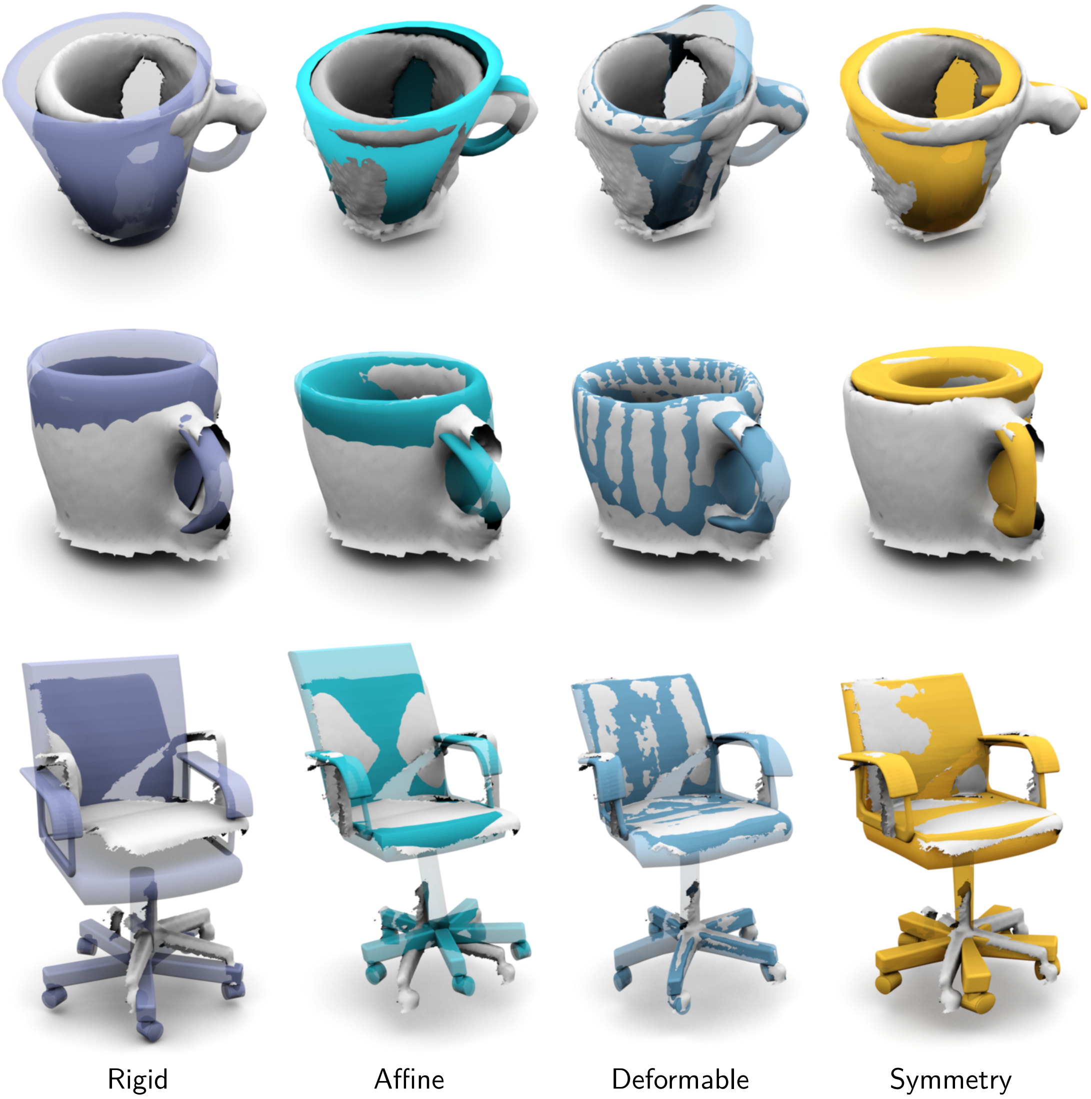


Figure 6.8: Overlays of the deformed models and the corresponding target shapes. From top to bottom: cone-shaped cup; rounded cup, office chair (single-view). In complex cases where no tight template is available, only the deformable ICP variants can match the data closely. Symmetry is able to preserve the structure better, as has already been shown in previous figures.

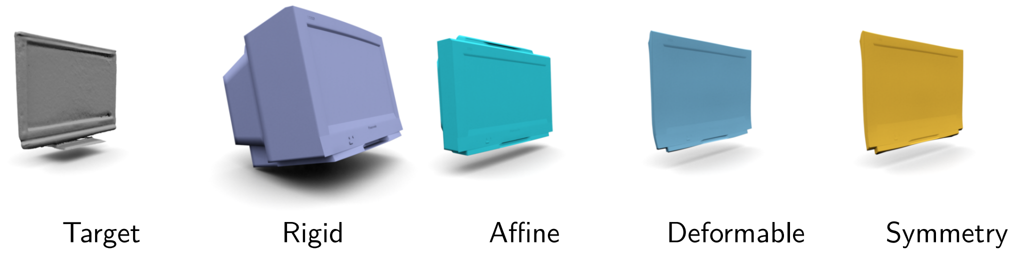


Figure 6.9: Scan of an LCD monitor; the template is a CRT TV. For this example, the coplanarity constraints had to be disabled, as the ICP algorithm was otherwise prevented from gradually establishing correspondences in the area that was not initially touched by the target surface, which prevented most of the deformation.

plausible model is reconstructed. In this case, the original, rigidly aligned model is not a good fit for the target data, especially in terms of lens diameter and length.

Additional Base-line Tests For further comparison, results are generated for ICP with affine mappings.[2] As can be seen in Figure 6.4, rows 1 and 2, for example, this works well for certain combinations of scan and template. In general, though, affine mappings lack the flexibility of general deformations, and the scan can thus often not be matched well. On the other hand, the risk of encountering artifacts increases nevertheless: Figure 6.4, row 3, Figure 6.5, row 2, or Figure 6.11, bottom, for example, show results where the shearing in particular has led to heavy distortions.

Previous Structure-aware Deformation Previous work by Bokeloh et al. [18] is also consulted for comparison, although this method originally does not perform scan registration. The deformation model may be recreated in the new framework by only using continuous translational constraints, i. e., collinearity and coplanarity. Discrete regular grids (with at least three instances in each direction) could also have been handled by their method, but there are no discrete grids in any of the examples presented in this chapter. The comparison in Figure 6.10 shows that using only the continuous translational symmetry constraints yields inferior results. It is well visible that the reduced model only maintains straight lines and planes, whereas the full model proposed in this chapter also preserves the rotational symmetries. The new model also captures the relations between straight lines and planes by establishing global reflective and rotational relations, which is visible in the body of the camera and the blades of the fan.

[2] An affine mapping permits rotation, translation, shear, and scaling.

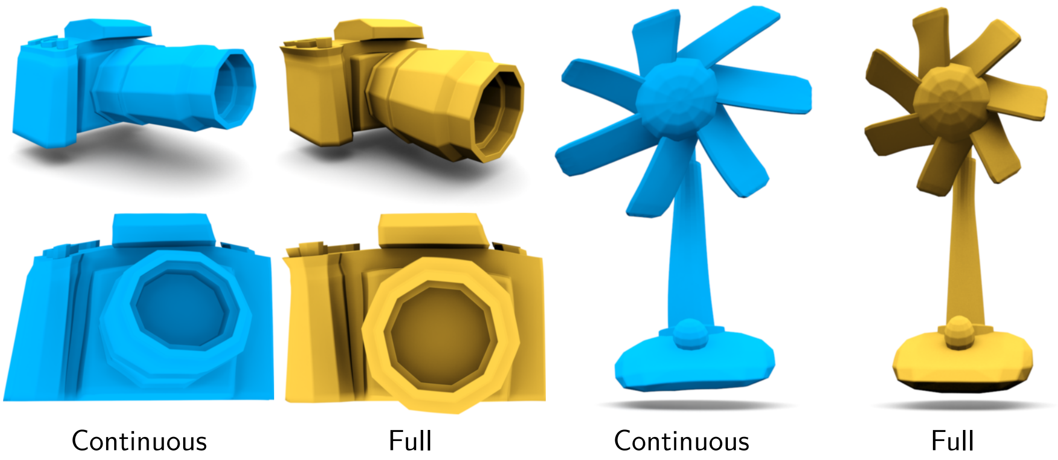


Figure 6.10: Comparison between the method proposed by Bokeloh et al. [18, 19], denoted as Continuous, which uses only continuous symmetry constraints, and the deformable ICP with all symmetry constraints, denoted as Full. The full set of symmetry constraints governs the deformation on a more global scale and provides better results. The continuous symmetry constraints enforce planar surfaces and straight edges only in localized parts without higher-level consistency.

In the follow-up paper [19], the same structure model is extended by hard constraints in order to reduce residual bending. This precludes the use of the rotational symmetries supported by the framework presented in this chapter, as the linearity of the group actions is crucial for this to work.

Discrete Symmetries Only The continuous symmetry constraints are useful in many applications, but there are cases where these additional constraints are not desired. In Figure 6.11, for example, results are presented for the hourglass and bowl data sets, both depicting objects that exhibit rotational symmetries only. For the generation of the results, the continuous symmetries have been disabled. The continuous symmetry constraints try to prevent the deformation field from adapting straight edges and planar surfaces to small local deformations in the scanned data. For the hourglass, the cylinder that was used as a template model is just an approximation of a real cylinder. The side consists of many planar surfaces that create coplanarity constraints, and the edges that connect them create collinearity constraints. If enforced, these constraints prevent the cylinder from adapting the typical hourglass shape. The situation with the bowl is similar, although there are less planar surfaces. The cylinder used as template mesh for

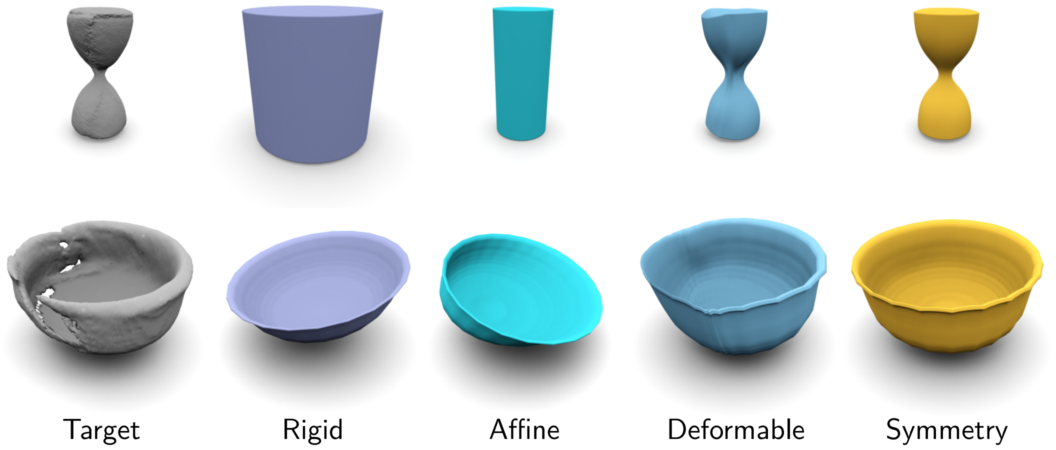


Figure 6.11: Scan of an hourglass (top) and of a bowl (bottom). Due to the nature of the scanned objects (curved surfaces are predominant in the scans), continuous symmetry constraints were not enforced. For the hourglass, for example, the continuous symmetries detected in the mantle of the cylinder would prevent the deformation to the hourglass shape. Symmetry provides excellent results; Deformable exhibits distortions.

the hourglass was modeled in a professional 3D modeling suite. For the bowl, backward ICP constraints from the template shape to the scanned data were used in addition to the normal forward ICP constraints. While this is often detrimental, particularly if the scan has lots of missing data, in this case it prevented the whole scanned data from being fitted by only a part of the template.

Quantitative Evaluation A quantitative evaluation of the chair data sets (Figure 6.4, rows 5 and 6) with ground-truth data is shown in Figure 6.12. The result and the high-quality ground-truth reference scan can be seen in Figure 6.13. In addition, Table 6.1 provides RMSE values. The RMSE values are calculated as

$$e_{\mathrm{RMSE}}(\mathcal{S},\mathcal{D}) = \sqrt{\frac{1}{\sum_i \chi_i} \sum_i \chi_i \left\|\mathbf{s}_j - \mathbf{d}_i\right\|^2} \quad , \tag{6.10}$$

with $\mathbf{d}_i \in \mathcal{D}$ and $\mathbf{s}_j \in \mathcal{S}$. As in Equation (6.5), the *closest point index* j is selected in a way that makes $\mathbf{s}_j$ the point in $\mathcal{S}$ closest to $\mathbf{d}_i$. The weighting factors χ_i are chosen as the combined area of all triangles comprising the respective vertex $\mathbf{d}_i$. The values given in the table were obtained by averaging the RMSE from the deformed template mesh to the reference scan and from the reference scan to the deformed template mesh. The deformable ICP variants fit the data more closely. Symmetry exhibits slightly higher error values than Deformable, especially at the borders of the mesh. This is the

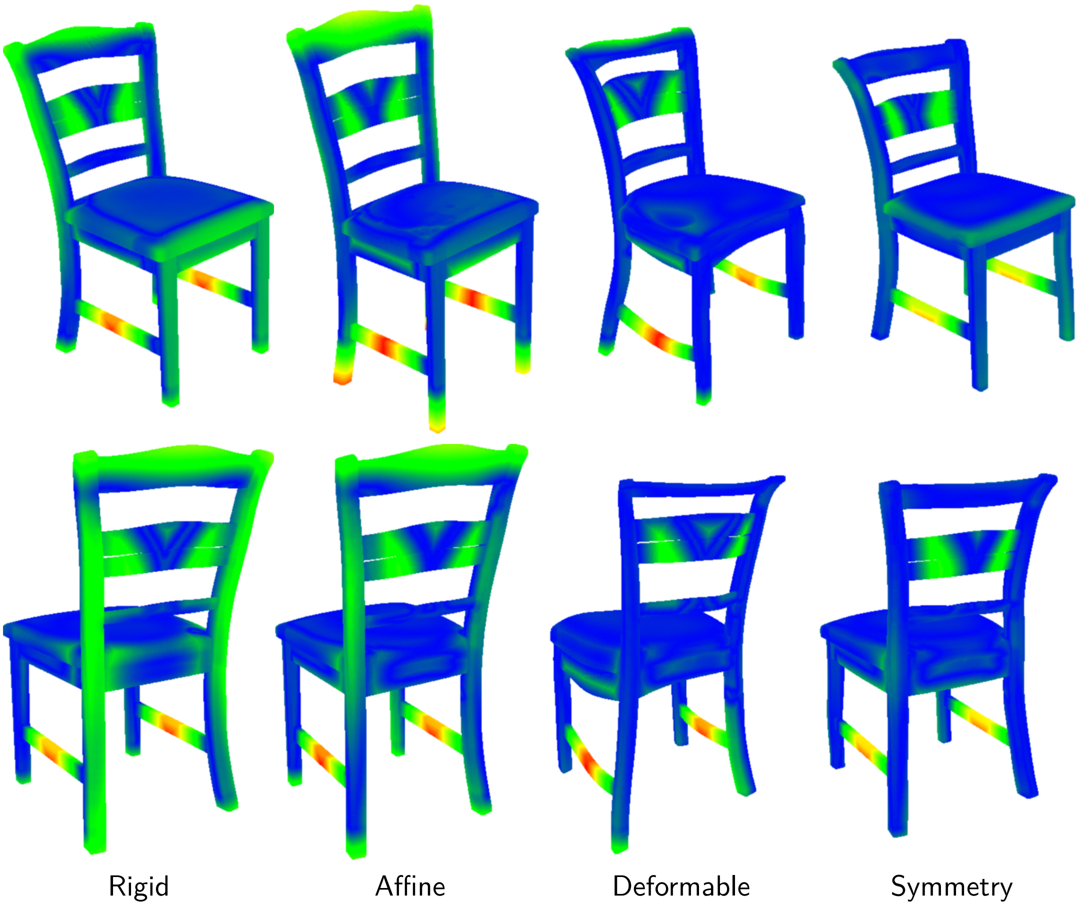


Figure 6.12: Error visualization of the chair data sets (Figure 6.4, rows 5 and 6). The upper and lower row use different scans, as shown in Figure 6.13 (Full and Single). For visualization, the error values are normalized per data set (i. e., row) and range from the lowest error observed (blue) to the highest error observed (red). Symmetry exhibits higher error values due to the additional constraints. For the single-view scan (bottom row), the advantage of the additional constraints can be seen: Deformable shows a higher error in the rear right leg due to missing data. The scan that served as reference can be found in Figure 6.13, right. The corresponding RMSE values are summarized in Table 6.1.

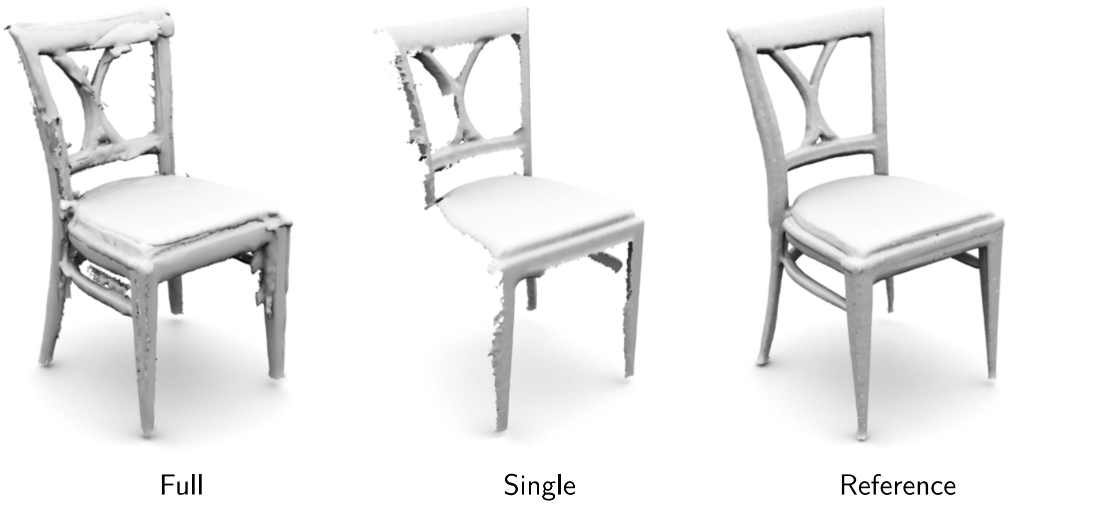


Figure 6.13: Depiction of the scans used to generate the results for the chair data sets (left and middle) and the scan used to generate the error visualizations shown in Figure 6.12 (right).

	RMSE			
Object	Rigid	Affine	Deformable	Symmetry
Chair, full	0.0296	0.0339	0.0239	0.0246
Chair, single	0.0307	0.0292	0.0246	0.0234

Table 6.1: RMSE values corresponding to the error visualizations shown in Figure 6.12. The deformable ICP variants exhibit lower errors; Symmetry is more constrained and therefore produces a slightly higher error in general. For the single-view scan of the chair, however, the error in Deformable is higher due to missing data that leaves one of the chair's legs unconstrained. This is consistent with the expectations. If data is missing, the symmetry constraints allow a better prediction of the shape in those areas. If all the data is available, the data cannot be accommodated to the extent that Deformable does.

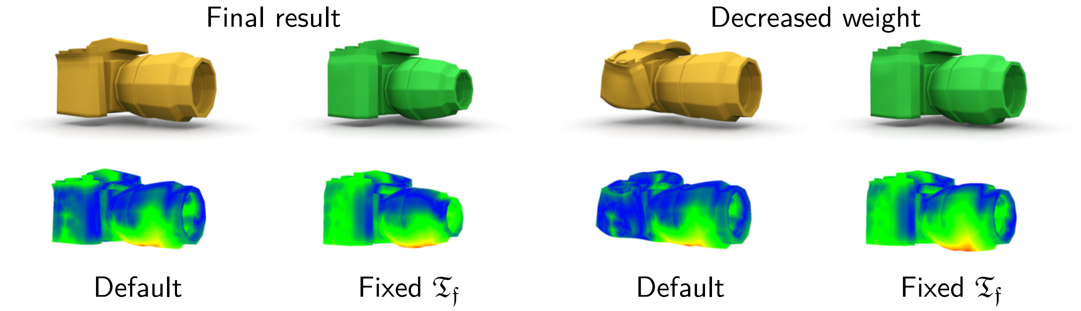

Final result
Decreased weight
Default
Fixed $\mathfrak{T}_{\mathfrak{f}}$
Default
Fixed $\mathfrak{T}_{\mathfrak{f}}$

Figure 6.14: Comparison between the results for the camera data set (Figure 6.7) and the results without updates to the symmetry transformations $\mathfrak{T}_{\mathfrak{f}}$. When the transformations are not updated, the scanned data cannot be accommodated well. To further illustrate the effect, the second row visualizes the error with respect to the scanned data. The scan is missing the whole bottom side of the camera and parts of the bottom of the lens, which can be seen in the visualization as high error values in the bottom part of the lens.

expected result, as all the additional symmetry constraints influence the result of the optimization procedure, thus preventing Symmetry from fitting the data as closely as Deformable. The RMSE is also slightly increased in this case. For the single-view scan, Symmetry achieves a better RMSE value than Deformable. Again, this is consistent with the expectations. A lot of data of the rear right of the chair is missing, and Deformable is unconstrained in these areas. Symmetry makes a better prediction of the object shape in this case, which leads to the lower RMSE value.

Update of the Symmetry Transformations Figure 6.14 examines the effect of the update of the symmetry transformations $\mathfrak{T}_{\mathfrak{f}}$ on the final result for the camera data set (Figure 6.7). To this end, the results were recreated without updating the initial transformations. As expected, the lacking update has a serious impact on the algorithm's capabilities to accommodate the scanned model. This result is consistent with the discussion in in Section 6.4.4.

User Guidance Deformable ICP often requires user guidance in addition to a rigid initialization. Again, symmetry constrains reduce the required effort. In Figure 6.15, an example is shown where this is demonstrated. In this example, the scan of a fan is registered to a manually-created template shape. The size of the rotor is not correctly matched in the initial results (see Figure 6.15, bottom). This is a result of the small sides of the blades providing insufficient point-to-plane constraints to successfully match

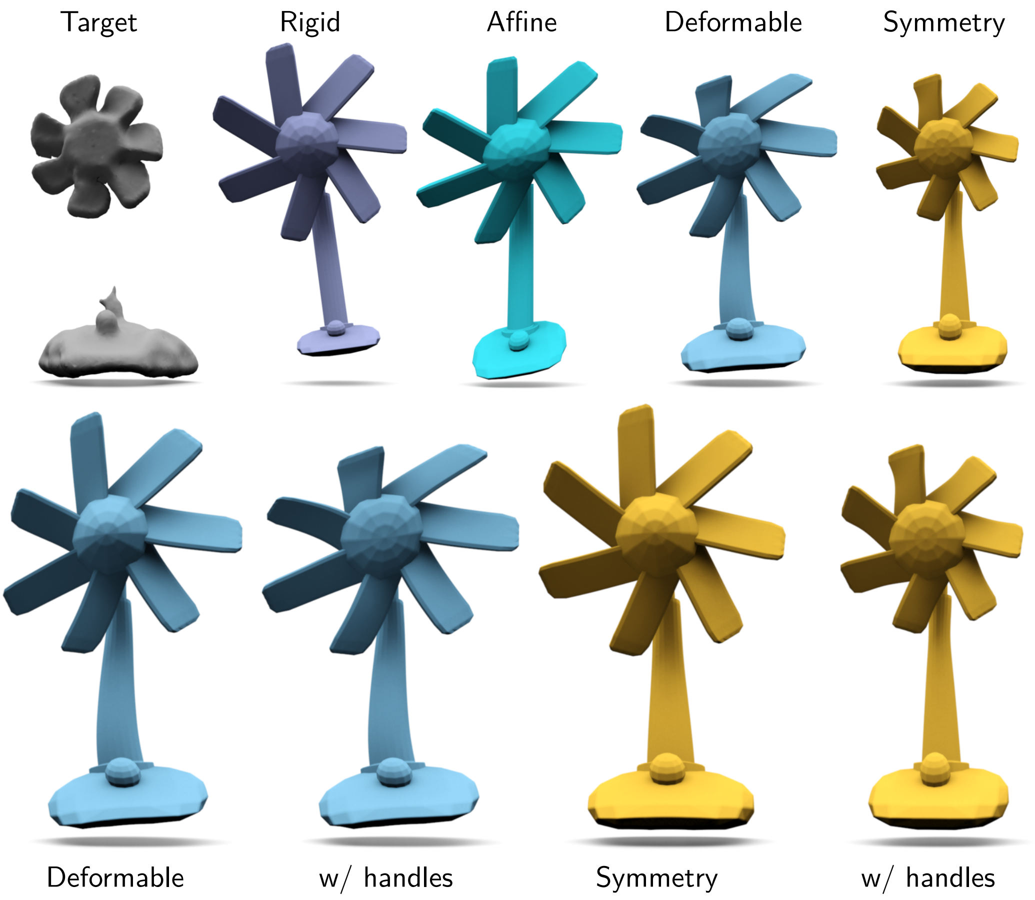


Figure 6.15: Scan of a small fan. For the deformable ICP variants, three handles have been used to shorten the blades of the fan and make the hub less pronounced in the final result. For Symmetry this is sufficient to get a good result due to the propagation of the deformation by the symmetry constraints. For Deformable, only the topmost blade is shortened. Comparisons with and without handles are on the bottom. To avoid distortions introduced by the handle constraints, the weights of the symmetry constraints have been increased tenfold.

the data. To fix the problem for Symmetry, three handle constraints are sufficient – two at the topmost blade to achieve a downward displacement, and one at the hub to fix the extrusion of the hub. The weight of the symmetry constraints ω_s is increased tenfold to avoid distortions due to the point-wise handles. To achieve the same result with Deformable alone, many more, consistently placed handles would be required. A comparison for the fan data set with and without the placed handle constraints is shown in Figure 6.15, bottom. As can be seen, the deactivation of the symmetry constraints leads to worse results. Symmetry-aware deformation in combination with handle constraints can be used to edit and fine-tune the results with greatly reduced effort.

Timings The results presented in this chapter were generated using a single-threaded C++ implementation running on an Intel Core i7-840QM processor at 1.87 GHz with 8GiB RAM. For Deformable and Symmetry, the average computation time is summarized in Table 6.2. The computational cost is dominated by the construction of the system matrix. This is more costly for the symmetry constraints than for the other constraints, because the symmetry constraints are integrated over large, overlapping areas. For the same reason, the chosen basis functions also critically influence the computation time. The linear basis functions have small support with two overlapping functions. The smooth basis functions on the other hand have larger support with four overlapping functions for the B-splines and a value of seven overlapping functions chosen for the Wendland functions, which significantly increases the computational burden. During implementation, there was no emphasis placed on the optimization of the numeric algorithms. The performance can thus still be improved significantly over the values given here.

6.8 Parameters

The parameters given in this section apply to all results shown in this chapter if not noted otherwise.

Deformation Model The grid spacing is uniformly set to $\epsilon_{\text{grid}} = 0.1$, as all template models are scaled to unit length during pre-processing. The weight of the symmetry constraints is set to $\omega_s = 3$, and the weight of the regularizer to $\omega_r = 1$. A weight $\omega_c = 100$ is chosen for the handle constraints. To emulate the behavior of affine ICP, the regularizer weight is set to $\omega_r = 10{,}000$ during the generation of Affine.

During empiric evaluation, it was found that setting the weight of the symmetry constraints ω_s to a value at least three times as high as the weight of the ICP-like constraints, which have an implicit weight of 1, is sufficient. With this value, the constraints were then respected during the optimization procedure for all examples

	Linear basis functions		**Smooth basis functions**	
Object	Deformable	Symmetry	Deformable	Symmetry
Bar table	0.48 s	2.51 s	0.65 min	19.87 min
Cooking pot	1.18 s	22.86 s	1.97 min	195.23 min
Cone-shaped cup	0.85 s	7.52 s	4.96 min	31.30 min
Rounded cup	0.99 s	7.70 s	5.24 min	84.14 min
Frying pan	1.50 s	6.11 s	1.40 min	53.01 min
Chair, full	0.46 s	2.19 s	1.96 min	24.92 min
Chair, single	0.35 s	2.49 s	0.86 min	24.66 min
Oval table	1.17 s	9.06 s	2.26 min	69.47 min
Armchair	2.74 s	14.82 s	4.68 min	99.19 min
Armchair, simple	1.19 s	3.44 s	2.82 min	26.07 min
Square table	1.48 s	12.53 s	2.76 min	126.70 min
Stepladder	2.51 s	4.93 s	1.03 min	46.03 min
Office chair, full	0.68 s	2.15 s	1.62 min	15.05 min
Office chair, single	0.65 s	1.94 s	1.28 min	14.16 min
Hourglass	1.59 s	7.94 s	0.63 min	6.84 min
Bowl	1.15 s	3.02 s	0.73 min	2.28 min
Fan	0.40 s	1.96 s	0.17 min	1.54 min
Camera	0.93 s	4.56 s	0.53 min	3.51 min
TV	1.55 s	3.20 s	0.59 min	2.11 min

Table 6.2: Average computation time per iterations for Deformable and Symmetry. The timings in the upper part of the table are for results generated using Wendland basis functions, the timings in the lower part for results generated using the cubic B-spline basis functions. While there is no visual difference in the results between Wendland and cubic B-spline basis functions, the computation times using the latter are much shorter due to reduced overlap.

given in this chapter – the one exception being the camera data set (Figure 6.7). In this case, the weight of the continuous symmetry constraints was not high enough to prevent the body of the camera from distorting, as was favored by the ICP-like constraints. For the generation of the final result, the weight of the continuous symmetry constraints was therefore increased to a value of $\omega_s = 30$.

The weight of the symmetry constraints ω_s can be seen as the strength of a bias towards a symmetric solution. If a lower value is chosen, the bias is gradually removed and the results become more similar to those of traditional deformable ICP.

For the fan data set (Figure 6.15), handle constraints have been placed manually in addition to the ICP-like constraints. As the weight of the handle constraints $\omega_c = 100$ is significantly higher than that of the other constraints, the fan blade on which the constraint was placed exhibits local deformations. The weight of the symmetry constraints was increased to $\omega_s = 30$ to counteract this. As a result, the local deformations are reduced to being negligible, but consequently the handle constraints are no longer perfectly fulfilled.

Symmetry Detection For checking collinearity and coplanarity, a uniform angle threshold of 1 degree is used. The distance threshold for checking feature and geometry compatibility is set to $4 \min(\epsilon_{\text{grid}}, \mu)$, with μ being the length of the shortest feature line. For symmetry selection, a cutoff-threshold expressed as a percentage of the supporting points is set. This parameter is model-specific and derived by examination of the result of symmetry detection, but it is typically below 50 percent. If a detected symmetry does not have the required number of supporting points, it is discarded. The planes that the coplanarity constraints are derived from are required to cover at least 1 percent of the total surface area of the template model.

ICP-Like Constraints The percentile threshold for the generation of the ICP-like constraints is set to $\tau_{\text{d}} = 80$. As already mentioned, the template models are scaled to unit length during pre-processing, which allows the near field threshold to be uniformly set to $\tau_{\text{nf}} = 0.05$.

6.9 Limitations

Fixed Structure The symmetries are applied as detected in the template model. This is an important limitation, as it creates bias if the symmetry structure of the template model differs from that of the target shape. Manual intervention may be used to remove this bias: The continuous symmetries were deactivated for the bowl and hourglass data sets shown in Figure 6.11 in order to use the overly simple templates.

There are also issues with particular configurations of symmetry constraints if multiple symmetries comprise the same parts of the template mesh. For example, a reflective

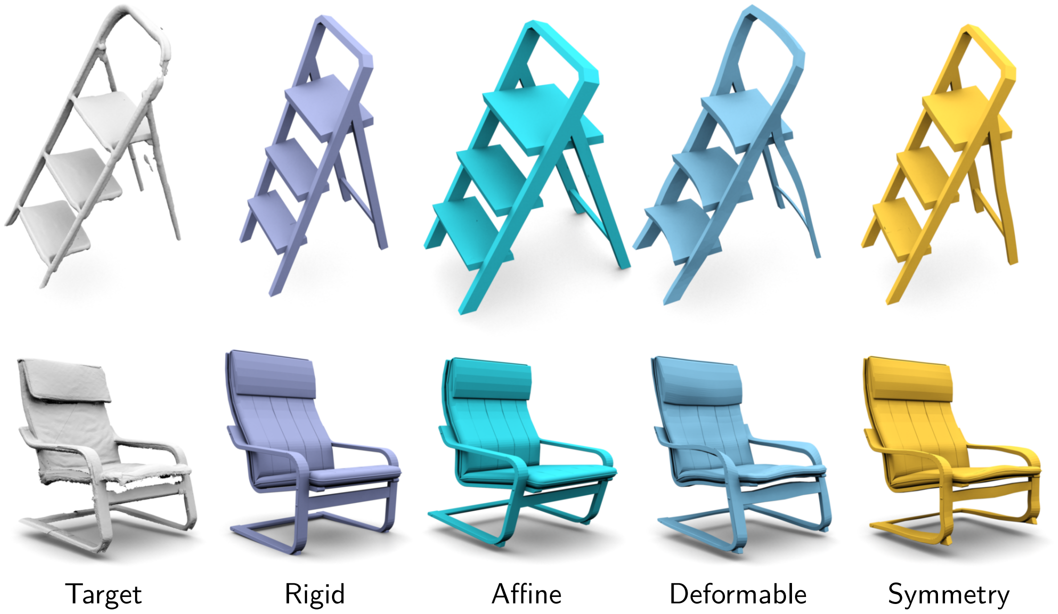


Figure 6.16: Full scan of a stepladder (top) and of the armchair from Figure 6.5 with a different template model (bottom). The Symmetry results are very close to the template model and consequently not a good fit for the target shapes. This is due to spurious symmetries in the template models: in combination with the continuous symmetry constraints, deformation is prevented.

symmetry might extend into a part that also has a rotational symmetry. Such configuration can prevent the template model from deforming by locking parts of the model in place. This is prominent in the stepladder and armchair data sets in Figure 6.16. To a lesser extent it is also visible in the office chair and oval table data sets (Figure 6.5, rows 2 and 4). Further illustration of this is provided in Figure 6.17. In this figure, a comparison of the results of Symmetry for different sets of symmetry constraints is shown. If only the most dominant symmetry (a global reflection) is enforced, the template is deformed to match the target almost perfectly. The ability to deform is subsequently suppressed by adding more constraints, until the application of all symmetry constraints detected leaves the template almost without deformation.

This problem could be solved by selectively breaking the constraints detected in the template model.

Local Convergence The symmetry constraints may prevent the local ICP from converging to a stable set of correspondences. Figure 6.18 shows results for the scan of a square desk, where this effect has occurred for Symmetry. The horizontal component of

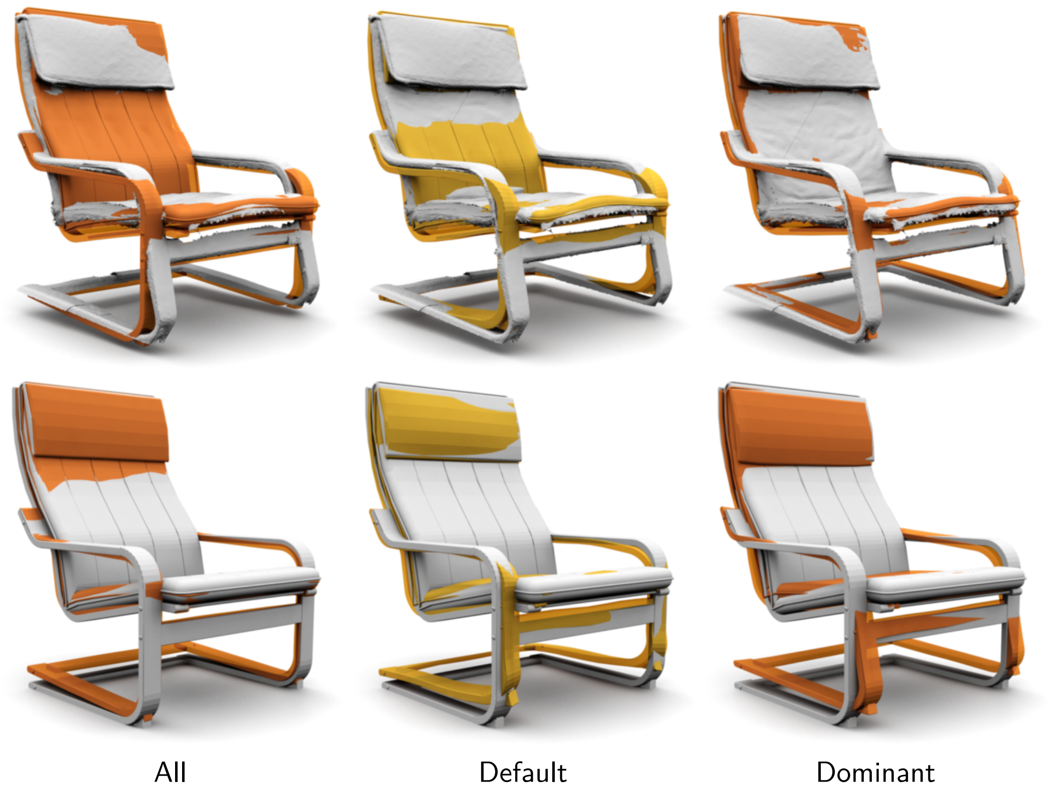


Figure 6.17: Overlays of Symmetry over the corresponding target shape (top) and the template model (bottom). The overlays are shown for different sets of symmetry constraints: For Dominant, only the most significant symmetry (a global reflective symmetry in this case) is kept. The deformed model largely deviates from the original model, but is almost a perfect match for the target shape. Default represents the set of symmetry constraints retained during normal processing. The additional symmetries compared to Dominant start to restrict the deformation, preventing the frame of the chair to be represented faithfully. All shows the result for all detected symmetry constraints enforced. There is almost no deformation due to interlocking constraints; only the bottom of the frame is able to extend upward.

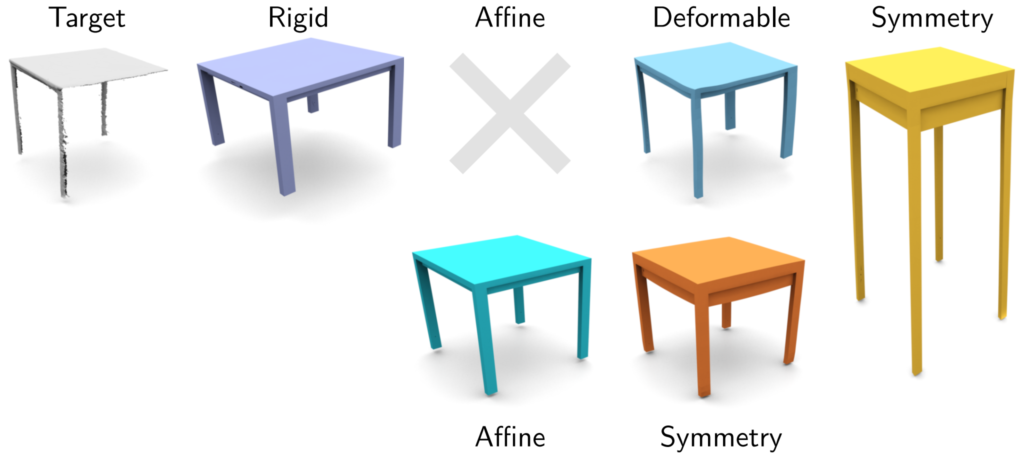


Figure 6.18: Partial scan of a square table. The elongation of the table in the Symmetry result is caused by the failure of the ICP algorithm to find a stable set of correspondences due to the constraints on the deformation field. Affine completely diverges for the same reason. For the results in the bottom row, a single handle constraint has been attached to a table leg manually.

the ICP constraints is countered by the symmetry constraints. At the same time, the vertical component of the ICP constraints makes the deformed table extend downwards. Different correspondences are then chosen in every subsequent step, which increases the effect with each iteration. Affine is also affected by this problem (see the result for the bar table in Figure 6.5). This behavior can largely eliminated by manually placing a single handle constraint. Deformable is not affected by these issues, because the influence of the regularizer on the ICP-like constraints is not as strong. This issue may also manifest in another form: The coplanarity constraints almost completely prevented the initial correspondences from updating for the TV data set (Figure 6.9). As a result, the algorithm converges with too little deformation.

To summarize, the new algorithm is a locally convergent shape matching technique that requires good initialization and possibly user guidance. It should be considered as a refined (deformable) ICP algorithm, and not as a fully-automatic shape matching system.

6.10 Discussion

In this chapter, a constrained optimization framework for symmetry-aware template mesh deformation has been presented. The method allows a user-provided 3D model to be fitted to low-quality scanning data, even if there is a high level of noise or only partial data available. During the ICP-based deformation and fitting procedure, the symmetry structure of the template geometry is preserved in a least-squares sense.

Reconstructions based on a number of low-quality Kinect Fusion scans with suitably chosen template models haven been shown for validation. The proposed method yields more plausible results in comparison to previous methods, yet the scanned geometry is closely respected in the reconstruction, even for templates quite different from the actual scanned object.

Future Work There are several very interesting avenues for future research. For example, it would be convenient to automatically select suitable subsets of symmetry constraints by analyzing both the template model and the scanned input data. Analysis of the latter is more complicated due to potentially high noise levels, distortions, and incomplete data. The symmetry axes and planes obtained during analysis might also enable a fully-automatic alignment of the template mesh and the scanned data for significantly different geometries.

Another goal for the future is to create a version of the framework that is suitable for real-time user interaction. This is currently prevented by the computational cost of the construction and evaluation of the symmetry constraints. A reformulation of the numeric computations for higher speed and possibly a suitable *general-purpose computing on graphics processing units (GPGPU)* implementation may lead to a performance at interactive frame rates.

The automatic selection of a suitable template mesh from a database would be another interesting direction for future work, as the user would then no longer be required to manually select a suitable template model. This could potentially be done by combining the guided real-time scanning approach presented by Kim et al. [84] with a conjoint symmetry analysis.

Of interest would also be the investigation of methods to overcome the limitations of the fixed template, which include fixed topology. To this end, a coarse initial template combined with implicit function fitting might enable topology-varying deformation with simultaneous preservation of the symmetry structure of the template.

In the future, there will most likely be an increasing demand for more general structure models which go beyond rigid symmetry. The incorporation of weakly supervised machine learning in the process of establishing correspondences and consequently the deduction of more general groups of admissible mappings may help to address this difficult challenge.

CHAPTER 7

Conclusion

In this thesis, contributions have been made to the areas of constrained camera motion estimation and constrained 3D reconstruction.

Constrained camera motion estimation has been the focus of Part I. The work presented in this part is directly aimed to improve the accuracy and robustness of structure from motion by joint estimation of camera position and orientation and the 3D structure of the scene.

In Chapter 3 – Bundle Adjustment for Stereoscopic 3D (Kurz et al. [89]), a novel stereoscopic camera model for bundle adjustment has been presented. This camera model is able to reduce the number of spurious degrees of freedom for stereoscopic image sequences in comparison to traditional bundle adjustment. It is geared to accommodate a variety of different camera setups, from consumer to professional, and provides improved reconstruction accuracy. In addition, the computation time required for bundle adjustment is reduced.

In Chapter 4 – A Generalized Framework for Constrained Bundle Adjustment, the stereoscopic camera model presented in Chapter 3 was developed into a generalized framework for constrained bundle adjustment. Hierarchies of Euclidean transformations were introduced as a convenient and flexible tool to model important types of constraints – collinearity, coplanarity, parallelism, and angular relations – in a homogeneous fashion. The framework is able to handle constraints on the scene structure and the camera geometry simultaneously, and is able to represent moving objects in a multibody SfM setting. It is thus very flexible while still being comparatively easy to implement.

The focus of Part II has been on constrained 3D reconstruction. Estimation and optimization of the camera position and orientation is not in the scope of these algorithms.

In Chapter 5 – Global Connectivity Constraints for 3D Line Segment Reconstruction (Jain et al. [76]), a probabilistic formulation of a 3D line reconstruction problem has been presented. The formulation takes the global connectivity information of the individual 3D line segments into account and yields excellent results. By using a novel technique for line segment grouping and outlier elimination, the individual reconstructions for each frame are merged without requiring brittle and error-prone line segment matching methods.

In Chapter 6 – Symmetry-aware Template Deformation and Fitting (Kurz et al. [90]), a framework for symmetry-guided mesh deformation has been presented. The technique relies on the user to provide a suitable template mesh and 3D scan of a target object, and then produces a deformed version of the template mesh that closely matches the desired target shape. The deformation procedure is guided by the automatically analyzed symmetry structure of the template model, which is preserved when the output is generated. This has the benefit of yielding plausible, high-quality results for the resulting model in contrast to the scan data, which typically exhibits noise, outliers, and missing data due to partial occlusions. Compared to previous work in this area, the novel framework provides substantial improvement in areas of missing data. In addition, it greatly facilitates user interaction, as user input is automatically propagated by the symmetry constraints to affect symmetric parts in a uniform way. The amount of user interaction required to achieve many common tasks is thereby significantly reduced.

All the work presented seeks to further the ease and flexibility with which the presented parameter estimation problems can be solved. By giving more plausible and accurate results, these methods may serve to make the algorithms employed today more robust and reliable in the future, and to make this technology available to a larger user base, and ultimately the consumer market.

An interesting topic for future research are different parametrizations for Euclidean transformations. This would benefit both the stereoscopic camera model presented in Chapter 3 and the generalized framework for constrained bundle adjustment described in Chapter 4 by potentially providing even greater flexibility. Aside from the representation of the constraints, it would also be interesting to research methods for the automatic extraction of the constraints. This would greatly reduce the amount of user interaction required to build the constraint structures. To achieve this, it might be possible to employ a probabilistic formulation similar to that presented in the context of 3D line segment reconstruction in Chapter 5. This approach could also help to identify and eliminate outlier object points in the constraints, regardless if user-generated or not. In any case, a thorough examination of the constraint interdependency structures will be necessary, both for a better understanding of the influence between constraints and to exploit the structures for faster optimization.

In conjunction with augmenting the proposed SfM methods with a probabilistic formulation, the method for the probabilistic reconstruction of 3D line segments described in Chapter 5 itself could be extended to include the camera parameters in addition to further constraints on the scene structure. This approach could yield better results if the initial camera parameters obtained by SfM, which the reconstruction intrinsically has to rely on, were inaccurate.

A probabilistic formulation could possibly also help to make the symmetry-aware deformation approach of Chapter 6 more versatile if it could be leveraged to selectively break symmetries not supported by the scanned object. As a whole, the analysis of the symmetry structure of both the template model and the scanned objects is without doubt one of the more important areas of future research. A better understanding in this area could permit the automatic selection of template models from a database based on the observations made in the scan, possibly in real-time. The models could then be edited on the fly, even before the scanning process is complete, which would allow greater control over the process and present opportunities to guide the user in order to obtain the best possible result.

The advances in the areas this thesis has contributed to might lead to the creation of an all-encompassing canonical framework for constrained camera motion estimation and 3D reconstruction. Through the combination of high-level structural analysis and probabilistic reasoning, a better understanding about the structure of the observed objects and environments might be obtained, including the fully-automatic identification of arbitrary objects by exploiting database knowledge. In a first step, this knowledge could then be leveraged to reliably detect outliers, resolve ambiguities, and generate constraints for the constrained camera motion estimation procedure. Thus enhanced, the camera motion estimates would allow the structural and probabilistic analysis to be refined in turn. A completely integrated formulation developed in subsequent steps might finally allow computer vision applications to match or possibly exceed the performance of the human visual system in general settings.

Bibliography

[1] Motilal Agrawal, Kurt Konolige, and Morten R. Blas. CenSurE: Center surround extremas for realtime feature detection and matching. In David Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008. ISBN 978-3-540-88692-1. DOI:10.1007/978-3-540-88693-8_8. 18

[2] Moulay A. Akhloufi, Vladimir Polotski, and Paul Cohen. Virtual view synthesis from uncalibrated stereo cameras. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (MMCS 1999)*, volume 2, pages 672–677, Florence, Italy, June 1999. IEEE. ISBN 0-7695-0253-9. DOI:10.1109/MMCS.1999.778564. 35

[3] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 510–517, Providence, RI, USA, June 2012. IEEE. ISBN 978-1-4673-1226-4. DOI:10.1109/CVPR.2012.6247715. 18

[4] Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE features. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 214–227. Springer, 2012. ISBN 978-3-642-33782-6. DOI:10.1007/978-3-642-33783-3_16. 18

[5] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers*, pages 587–594, San Diego, CA, USA, July 2003. ACM. ISBN 1-58113-709-5. DOI:10.1145/1201775.882311. 94, 96, 101

[6] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR 2007), pages 1–8, Minneapolis, MN, USA, June 2007. IEEE. ISBN 1-4244-1179-3. DOI:10.1109/CVPR.2007.383165. 96

[7] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics (TOG)*, 24(3):408–416, July 2005. ISSN 0730-0301. DOI:10.1145/1073204.1073207. 96

[8] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics (TOG)*, 26(3):10:1–10:9, July 2007. ISSN 0730-0301. DOI:10.1145/1276377.1276390. 95

[9] Caroline Baillard, Cordelia Schmid, Andrew Zisserman, and Andrew W. Fitzgibbon. Automatic line matching and 3d reconstruction of buildings from multiple views. In *International Archive of Photogrammetry and Remote Sensing: Proceedings of the ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume XXXII, pages 69–80, Munich, Germany, September 1999. ISPRS. 73

[10] Adrien Bartoli and Peter Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision (IJCV)*, 52(1):45–64, April 2003. ISSN 0920-5691. DOI:10.1023/A:1022318524906. 51

[11] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding (CVIU)*, 100(3):416–441, December 2005. ISSN 1077-3142. DOI:10.1016/j.cviu.2005.06.001. 9, 74

[12] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. SURF: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006. ISBN 978-3-540-33832-1. DOI:10.1007/11744023_32. 18

[13] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. Variational harmonic maps for space deformation. *ACM Transactions on Graphics (TOG)*, 28(3):34:1–34:11, August 2009. ISSN 0730-0301. DOI:10.1145/1531326.1531340. 94

[14] Gerhard H. Bendels, Reinhard Klein, and Andreas Schilling. Image and 3D-object editing with precisely specified editing regions. In Thomas Ertl, Bernd Girod, Günther Greiner, Heinrich Niemann, Hans-Peter Seidel, Eckehard Steinbach, and

Rüdiger Westermann, editors, *Proceedings of the Vision, Modeling, and Visualization Conference (VMV 2003)*, pages 451–460, Munich, Germany, November 2003. AKA. ISBN 3-89838-048-3. 100

[15] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging faces in images. *Computer Graphics Forum (CGF)*, 23(3):669–676, August 2004. ISSN 1467-8659. `DOI:10.1111/j.1467-8659.2004.00799.x`. 96

[16] Michael Bleyer and Margrit Gelautz. Temporally consistent disparity maps from uncalibrated stereo videos. In *Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis (ISISPA 2009)*, pages 383–387, Salzburg, Austria, September 2009. IEEE. ISBN 978-953-184-135-1. 36

[17] Martin Bokeloh, Michael Wand, and Hans-Peter Seidel. A connection between partial symmetry and inverse procedural modeling. *ACM Transactions on Graphics (TOG)*, 29(4):104:1–104:10, July 2010. ISSN 0730-0301. `DOI:10.1145/1778765.1778841`. 96

[18] Martin Bokeloh, Michael Wand, Vladlen Koltun, and Hans-Peter Seidel. Pattern-aware shape deformation using sliding dockers. *ACM Transactions on Graphics (TOG)*, 30(6):123:1–123:10, December 2011. ISSN 0730-0301. `DOI:10.1145/2070781.2024157`. 95, 104, 114, 115

[19] Martin Bokeloh, Michael Wand, Hans-Peter Seidel, and Vladlen Koltun. An algebraic model for parameterized shape editing. *ACM Transactions on Graphics (TOG)*, 31(4):78:1–78:10, July 2012. ISSN 0730-0301. `DOI:10.1145/2185520.2185574`. 95, 105, 115

[20] Didier Bondyfalat and Sylvain Bougnoux. Imposing euclidean constraints during self-calibration processes. In Reinhard Koch and Luc J. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 224–235. Springer, 1998. ISBN 978-3-540-65310-3. `DOI:10.1007/3-540-49437-5_15`. 51

[21] Mario Botsch and Leif Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics (TOG)*, 23(3):630–634, August 2004. ISSN 0730-0301. `DOI:10.1145/1015706.1015772`. 100

[22] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14 (1):213–230, January 2008. ISSN 1077-2626. `DOI:10.1109/TVCG.2007.1054`. 94

[23] Gary Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 25 (11):120, 122–125, November 2000. ISSN 1044-789X. 16

[24] Michael J. Brooks, Lourdes de Agapito, Du Q. Huynh, and Luis Baumela. Towards robust metric reconstruction via a dynamic uncalibrated stereo heads. *Computer Vision and Image Understanding (CVIU)*, 16(14):989–1002, December 1998. ISSN 0262-8856. `DOI:10.1016/S0262-8856(98)00064-X`. 35

[25] Benedict J. Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics (TOG)*, 26(3):21:1–21:9, July 2007. ISSN 0730-0301. `DOI:10.1145/1276377.1276404`. 95, 96, 101

[26] Duane C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, May 1966. 15, 16

[27] Duane C. Brown. *Advanced Methods for the Calibration of Metric Cameras*. U.S. Army Engineer Topographic Laboratories, 1968. 15

[28] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(3):500–513, March 2011. ISSN 0162-8828. `DOI:10.1109/TPAMI.2010.143`. 18

[29] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer, 2010. ISBN 978-3-642-15560-4. `DOI:10.1007/978-3-642-15561-1_56`. 18

[30] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 8(6):679–698, November 1986. ISSN 0162-8828. `DOI:10.1109/TPAMI.1986.4767851`. 76

[31] Manmoham Chandraker, Jongwoo Lim, and David Kriegman. Moving in stereo: Efficient structure and motion using lines. In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV 2009)*, pages 1741–1748, Kyoto, Japan, September 2009. IEEE. ISBN 978-1-4244-4420-5. `DOI:10.1109/ICCV.2009.5459390`. 34

[32] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas J. Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3D modeling. *ACM Transactions on Graphics (TOG)*, 30(4):35:1–35:10, July 2011. ISSN 0730-0301. `DOI:10.1145/2010324.1964930`. 96

[33] Lin Chen and Xiangxu Meng. Anisotropic resizing of model with geometric textures. In *Proceedings of the 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 289–294, San Francisco, CA, USA, October 2009. ACM. ISBN 978-1-60558-711-0. `DOI:10.1145/1629255.1629292`. 95

[34] Chia-Ming Cheng, Shang-Hong Lai, and Shyh-Haur Su. Self image rectification for uncalibrated stereo video with varying camera motions and zooming effects. In *Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2009)*, pages 21–24, Yokohama, Japan, May 2009. ISBN 978-4-901122-09-2. 36

[35] Roberto Cipolla, Paul A. Hadfield, and Nicholas J. Hollinghurst. Uncalibrated stereo vision with pointing for a man-machine interface. In *Proceedings of the IAPR Workshop on Machine Vision Applications (MVA 1994)*, pages 163–166, Kawasaki, Japan, December 1994. 35

[36] David Claus and Andrew W. Fitzgibbon. A rational function lens distortion model for general cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 213–219, San Diego, CA, USA, June 2005. IEEE. ISBN 0-7695-2372-2. `DOI:10.1109/CVPR.2005.43`. 16, 21

[37] Robert T. Collins. A space-sweep approach to true multi-image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1996)*, pages 358–363, San Francisco, CA, USA, June 1996. IEEE. ISBN 0-8186-7259-5. `DOI:10.1109/CVPR.1996.517097`. 78

[38] Sébastien Cornou, Michel Dhome, and Patrick Sayd. Architectural reconstruction with multiple views and geometric constraints. In Richard Harvey and Andrew Bangham, editors, *Proceedings of the British Machine Vision Conference (BMVC 2003)*, pages 76.1–76.10. BMVA Press, 2003. ISBN 1-901725-23-5. `DOI:10.5244/C.17.76`. 51

[39] Thao Dang, Christian Hoffmann, and Christoph Stiller. Continuous stereo self-calibration by camera parameter tracking. *IEEE Transaction on Image Processing (TIP)*, 18(7):1536–1550, July 2009. ISSN 1057-7149. `DOI:10.1109/TIP.2009.2017824`. 35

[40] Kaichang Di, Fengliang Xu, and Rongxing Li. Constrained bundle adjustment of panoramic stereo images for mars landing site mapping. In *Proceedings of the 4th International Symposium on Mobile Mapping Technology (MMT 2004)*, pages 29–31, Kunming, China, March 2004. 34

[41] Terri L. Fauber. *Radiographic Imaging & Exposure*. Mosby, fourth edition, March 2012. ISBN 978-0-323-08322-5. 13

[42] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. `DOI:10.1145/358669.358692`. 19

[43] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision – ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1998. ISBN 978-3-540-64569-6. DOI:10.1007/BFb0055675. 22

[44] Andrew W. Fitzgibbon and Andrew Zisserman. Multibody structure and motion: 3-D reconstruction of independently moving objects. In *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 891–906. Springer, 2000. ISBN 978-3-540-67685-0. DOI:10.1007/3-540-45054-8_58. 10, 51

[45] Wolfgang Förstner. Minimal representations for uncertainty and estimation in projective spaces. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, volume 6493 of *Lecture Notes in Computer Science*, pages 619–632. Springer, 2011. ISBN 978-3-642-19308-8. DOI:10.1007/978-3-642-19309-5_48. 51

[46] Jan-Michael Frahm, Kevin Köser, and Reinhard Koch. Pose estimation for multi-camera systems. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Bernhard Schölkopf, and Martin A. Giese, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 286–293. Springer, 2004. ISBN 978-3-540-22945-2. DOI:10.1007/978-3-540-28649-3_35. 34

[47] Pascal Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision (IJCV)*, 38 (2):153–171, July 2000. ISSN 0920-5691. DOI:10.1023/A:1008105802790. 51

[48] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM Transactions on Graphics (TOG)*, 23(3):652–663, August 2004. ISSN 0730-0301. DOI:10.1145/1015706.1015775. 96

[49] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1362–1376, August 2010. ISSN 0162-8828. DOI:10.1109/TPAMI.2009.161. 9

[50] Andrea Fusiello and Luca Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, Tampa, FL, USA, December 2008. IEEE. ISBN 978-1-4244-2174-9. DOI:10.1109/ICPR.2008.4761561. 35

[51] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics (TOG)*, 25(1):130–150, January 2006. ISSN 0730-0301. `DOI:10.1145/1122501.1122507`. 105

[52] Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. iWIRES: An analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics (TOG)*, 28(3):33:1–33:10, July 2009. ISSN 0730-0301. `DOI:10.1145/1531326.1531339`. 95

[53] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 1272–1279, Portland, OR, USA, June 2013. IEEE. `DOI:10.1109/CVPR.2013.168`. 10

[54] Natasha Gelfand and Leonidas J. Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP 2004)*, pages 214–223, Nice, France, July 2004. ACM. ISBN 3-905673-13-4. `DOI:10.1145/1057432.1057461`. 104, 105

[55] Simon Gibson, Jonathan Cook, Toby L. J. Howard, Roger J. Hubbold, and Dan Oram. Accurate camera calibration for off-line, video-based augmented reality. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, pages 37–46, Darmstadt, Germany, September 2002. IEEE. ISBN 0-7695-1781-1. `DOI:10.1109/ISMAR.2002.1115068`. 8, 22

[56] Simon Gibson, Roger J. Hubbold, Jonathan Cook, and Toby L. J. Howard. Interactive reconstruction of virtual environments from video sequences. *Computers & Graphics*, 27(2):293–301, January 2003. ISSN 0097-8493. `DOI:10.1016/S0097-8493(02)00285-6`. 51, 72

[57] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8. IEEE, October 2007. ISBN 978-1-4244-1631-8. `DOI:10.1109/ICCV.2007.4408933`. 9, 35

[58] Haipeng Guo and William Hsu. A survey of algorithms for real-time bayesian network inference. In *Proceedings of the AAAI/KDD/UAI Joint Workshop on Real-time Decision Support and Diagnosis Systems (RTDSDS 2002), AAAI Technical Report WS-02-15*, pages 1–12, Edmonton, Canada, July 2002. AAAI. ISBN 978-1-57735-168-9. 77

[59] Theo Hahn, editor. *International Tables for Crystallography, Volume A: Space-group Symmetry*. International Tables for Crystallography. Wiley, 2006. ISBN 978-0-7923-6590-7. DOI:10.1107/97809553602060000100. 104

[60] Dirk Hähnel, Sebastian Thrun, and Wolfram Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 915–920, Acapulco, Mexico, August 2003. Morgan Kaufmann. 96, 100

[61] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference (AVC 1988)*, pages 147–151, August 1988. 17

[62] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, March 2004. ISBN 978-0-521-54051-3. 7, 12, 24, 27, 28

[63] Richard I. Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1992)*, pages 761–764, Champaign, IL, USA, June 1992. IEEE. ISBN 0-8186-2855-3. DOI:10.1109/CVPR.1992.223179. 35

[64] Nils Hasler, Bodo Rosenhahn, Thorsten Thormählen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 224–231, Miami, FL, USA, June 2009. IEEE. ISBN 978-1-4244-3992-8. DOI:10.1109/CVPR.2009.5206859. 35

[65] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum (CGF)*, 28(2):337–346, March 2009. ISSN 1467-8659. DOI:10.1111/j.1467-8659.2009.01373.x. 96

[66] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 1842 of *Lecture Notes in Computer Science*, pages 759–773. Springer, 2012. ISBN 978-3-642-33708-6. DOI:10.1007/978-3-642-33709-3_54. 18

[67] Stephan Heuel and Wolfgang Förstner. Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 2, pages 517–524, Kauai, HI, USA, December 2001. IEEE. ISBN 0-7695-1272-0. DOI:10.1109/CVPR.2001.991006. 74

[68] Heiko Hirschmüller, Peter R. Innocent, and Jon M. Garibaldi. Fast, unconstrained camera motion estimatoin from stereo without tracking and robust statistics. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARCV 2002)*, volume 2, pages 1099–1104, Singapore, December 2002. IEEE. ISBN 981-04-8364-3. `DOI:10.1109/ICARCV.2002.1238577`. 34

[69] Steve E. Hodges and Robert J. Richards. Uncalibrated stereo for pcb drilling. In *Proceedings of the IEE Colloquium on Application of Machine Vision*, pages 4/1–4/6, London, UK, May 1995. IEEE. `DOI:10.1049/ic:19950746`. 35

[70] Steven Holmes, Gabe Sibley, Georg Klein, and David W. Murray. A relative frame representation for fixed-time bundle adjustment in SFM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 2264–2269, Kobe, Japan, May 2009. IEEE. ISBN 978-1-4244-2788-8. `DOI:10.1109/ROBOT.2009.5152596`. 51

[71] Jan Hrabáček and Frank A. van den Heuvel. Weighted geometric object constraints integrated in a line-photogrammetric bundle adjustment. In Hirofumi Chikatsu and Frank van den Heuvel, editors, *International Archive of Photogrammetry and Remote Sensing: Proceedings of the XIXth ISPRS Congress, Technical Commission V: Close-Range Techniques and Machine Vision*, volume XXXIV, pages 380–387, Amsterdam, The Netherlands, September 2000. ISPRS. 50

[72] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics (TOG)*, 25(3):1126–1134, July 2006. ISSN 0730-0301. `DOI:10.1145/1141911.1142003`. 99

[73] Qi-Xing Huang, Radomir Mech, and Nathan Carr. Optimizing structure preserving embedded deformation for resizing images and vector art. *Computer Graphics Forum (CGF)*, 28(7):1887–1896, October 2009. ISSN 1467-8659. `DOI:10.1111/j.1467-8659.2009.01567.x`. 95

[74] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–7, Rio de Janeiro, Brazil, October 2007. IEEE. ISBN 978-1-4244-1631-8. `DOI:10.1109/ICCV.2007.4409000`. 36

[75] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual*

ACM Symposium on User Interface Software and Technology (UIST 2011), pages 559–568, Santa Barbara, CA, USA, October 2011. ACM. ISBN 978-1-4503-0716-1. DOI:10.1145/2047196.2047270. 94

[76] Arjun Jain, Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 1586–1593, San Francisco, CA, USA, June 2010. IEEE. ISBN 978-1-4244-6984-0. DOI:10.1109/CVPR.2010.5539781. 5, 6, 71, 130

[77] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel. Exploring shape variations by 3D-model decomposition and part-based recombination. *Computer Graphics Forum (CGF)*, 31(2pt3):631–640, May 2012. ISSN 1467-8659. DOI:10.1111/j.1467-8659.2012.03042.x. 96

[78] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71:1–71:9, July 2007. ISSN 0730-0301. DOI:10.1145/1276377.1276466. 94

[79] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics (TOG)*, 24(3):561–566, July 2005. ISSN 0730-0301. DOI:10.1145/1073204.1073229. 94

[80] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):55:1–55:11, July 2012. ISSN 0730-0301. DOI:10.1145/2185520.2185551. 96

[81] Niklas Karlsson, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E. Munich. The vSLAM algorithm for robust localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 24–29, Barcelona, Spain, April 2005. IEEE. ISBN 0-7803-8914-X. DOI:10.1109/ROBOT.2005.1570091. 8

[82] Jae-Hak Kim, Hongdong Li, and Richard I. Hartley. Motion estimation for multi-camera systems using global optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, Anchorage, AK, USA, June 2008. IEEE. ISBN 978-1-4244-2242-5. DOI:10.1109/CVPR.2008.4587680. 34

[83] Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas J. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6):138:1–138:11, November 2012. ISSN 0730-0301. `DOI:10.1145/2366145.2366157`. 97

[84] Young Min Kim, Niloy J. Mitra, Qixing Huang, and Leonidas J. Guibas. Guided real-time scanning of indoor objects. *Computer Graphics Forum (CGF)*, 32(7): 177–186, November 2013. ISSN 1467-8659. `DOI:10.1111/cgf.12225`. 97, 127

[85] Jung-Hwan Ko, Chang-Ju Park, and Eun-Soo Kim. A new rectification scheme for uncalibrated stereo image pairs and its application to intermediate view reconstruction. In Bahram Javidi and Demetri Psaltis, editors, *Optical Information Systems II*, volume 5557 of *SPIE Proceedings*, pages 98–109. SPIE, October 2004. `DOI:10.1117/12.560274`. 35

[86] Vladislav Kraevoy and Alla Sheffer. Template-based mesh completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 13:1–13:10, Vienna, Austria, July 2005. Eurographics Association. ISBN 3-905673-24-X. 96

[87] Vladislav Kraevoy, Alla Sheffer, Ariel Shamir, and Daniel Cohen-Or. Non-homogeneous resizing of complex models. *ACM Transactions on Graphics (TOG)*, 27(5):111:1–111:9, December 2008. ISSN 0730-0301. `DOI:10.1145/1409060.1409064`. 95

[88] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory (TIT)*, 47(2):498–519, February 2001. ISSN 0018-9448. `DOI:10.1109/18.910572`. 81

[89] Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. Bundle adjustment for stereoscopic 3D. In André Gagalowicz and Wilfried Philips, editors, *Computer Vision/Computer Graphics Collaboration Techniques*, volume 6930 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2011. ISBN 978-3-642-24135-2. `DOI:10.1007/978-3-642-24136-9_1`. 5, 6, 33, 129

[90] Christian Kurz, Xiaokun Wu, Michael Wand, Thorsten Thormählen, Pushmeet Kohli, and Hans-Peter Seidel. Symmetry-aware template deformation and fitting. *Computer Graphics Forum Early View (Online Version of Record published before inclusion in an issue)*, March 2014. ISSN 1467-8659. `DOI:10.1111/cgf.12344`. 5, 6, 91, 130

[91] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference*

on Computer Vision (ICCV 2011), pages 2548–2555. IEEE, November 2011. ISBN 978-1-4577-1101-5. DOI:10.1109/ICCV.2011.6126542. 18

[92] Maxime Lhuillier and Long Quan. Quasi-dense reconstruction from image sequence. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision – ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2002. ISBN 978-3-540-43744-4. DOI:10.1007/3-540-47967-8_9. 22

[93] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. GlobFit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 Papers*, pages 52:1–52:12, Vancouver, Canada, August 2011. ACM. ISBN 978-1-4503-0943-1. DOI:10.1145/1964921.1964947. 97

[94] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):78:1–78:10, August 2008. ISSN 0730-0301. DOI:10.1145/1360612.1360677. 94

[95] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, November 2004. ISSN 0920-5691. DOI:10.1023/B:VISI.0000029664.99615.94. 18

[96] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981)*, volume 2, pages 674–679, Vancouver, Canada, August 1981. Morgan Kaufmann Publishers Inc. 17

[97] Robert Mandelbaum, Garbis Salgian, and Harpreet S. Sawhney. Correlation-based estimation of ego-motion and structure from motion and stereo. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV 1999)*, volume 1, pages 544–550, Kerkyra, Greece, September 1999. IEEE. ISBN 0-7695-0164-8. DOI:10.1109/ICCV.1999.791270. 35

[98] Daniel Martinec and Tom Pajdla. Line reconstruction from many perspective images by factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, volume 1, pages 497–502, Madison, WI, USA, June 2003. IEEE. ISBN 0-7695-1900-8. DOI:10.1109/CVPR.2003.1211395. 74

[99] Larry Matthies and Steven A. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation (JRA)*, 3(3):239–248, June 1987. ISSN 0882-4967. DOI:10.1109/JRA.1987.1087097. 35

[100] Stephen J. Maybank and Olivier D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision (IJCV)*, 8(2):123–151, August 1992. ISSN 0920-5691. `DOI:10.1007/BF00127171`. 24

[101] J. Chris McGlone. Bundle adjustment with object space geometric constraints for site modeling. In David M. McKeown and Ian J. Dowman, editors, *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, volume 2486 of *SPIE Proceedings*, pages 25–36. SPIE, July 1995. `DOI:10.1117/12.213128`. 50

[102] Philip F. McLauchlan, Xinquan Shen, Anastasios Manessis, Phil Palmer, and Adrian Hilton. Surface-based structure-from-motion using feature groupings. In *Proceedings of the 4th Asian Conference on Computer Vision (ACCV 2000)*, pages 699–705, Taipei, Taiwan, January 2000. 50, 52, 67

[103] Jochen Meidow, Wolfgang Förstner, and Christian Beder. Optimal parameter estimation with homogeneous entities and arbitrary constraints. In Joachim Denzler, Gunther Notni, and Herbert Süße, editors, *Pattern Recognition*, volume 5748 of *Lecture Notes in Computer Science*, pages 292–301. Springer, 2009. ISBN 978-3-642-03797-9. `DOI:10.1007/978-3-642-03798-6_30`. 50

[104] Paul Merrell. Example-based model synthesis. In *Proceedings of the 2007 Symposium on Interactive 3D graphics and Games (I3D 2007)*, pages 105–112, Seattle, WA, USA, April 2007. ACM. ISBN 978-1-59593-628-8. `DOI:10.1145/1230100.1230119`. 96

[105] Willard Miller. *Symmetry Groups and their Applications.* Academic Press, July 1972. ISBN 978-0-124-97460-9. 103

[106] Dongbo Min and Kwanghoon Sohn. Edge-preserving simultaneous joint motion-disparity estimation. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, volume 2, pages 74–77, Hong Kong, China, August 2006. IEEE. ISBN 0-7695-2521-0. `DOI:10.1109/ICPR.2006.470`. 36

[107] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas J. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the Second Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP 2004)*, pages 22–31, Nice, France, July 2004. ACM. ISBN 3-905673-13-4. `DOI:10.1145/1057432.1057435`. 108

[108] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics (TOG)*, 25 (3):560–568, July 2006. ISSN 0730-0301. `DOI:10.1145/1141911.1141924`. 105

[109] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3D geometry: Extraction and applications. *Computer Graphics Forum (CGF)*, 32(6):1–23, February 2013. ISSN 1467-8659. `DOI:10.1111/cgf.12010`. 102

[110] Niloy J. Mitra, Michael Wand, Hao Zhang, Daniel Cohen-Or, and Martin Bokeloh. Structure-aware shape processing. Eurographics 2013 State-of-the-Art Report (STAR), May 2013. 92

[111] Nicholas Molton and Michael Brady. Practical structure and motion from stereo when motion is unconstrained. *International Journal of Computer Vision (IJCV)*, 39(1):5–23, August 2000. ISSN 0920-5691. `DOI:10.1023/A:1008191416557`. 35

[112] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research (JMLR)*, 11:2169–2173, August 2010. 83

[113] Theo Moons, David Frère, Jan Vandekerckhove, and Luc J. Van Gool. Automatic modelling and 3D reconstruction of urban house roofs from high resolution aerial imagery. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision – ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, pages 410–425. Springer, 1998. ISBN 978-3-540-64569-6. `DOI:10.1007/BFb0055681`. 73

[114] David Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 649–663. Springer, 2000. ISBN 978-3-540-67685-0. `DOI:10.1007/3-540-45054-8_42`. 22

[115] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 1, pages 652–659, Washington, DC, USA, June 2004. IEEE. ISBN 0-7695-2158-4. `DOI:10.1109/CVPR.2004.1315094`. 35

[116] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, second edition, 2006. ISBN 978-0-387-30303-1. `DOI:10.1007/978-0-387-40065-5`. 27, 52, 102

[117] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems (RAS)*, 43(4):215–229, June 2003. `DOI:10.1016/S0921-8890(03)00004-6`. 35

[118] Kemal E. Ozden, Konrad Schindler, and Luc J. Van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(6):1134–1141, June 2010. ISSN 0162-8828. `DOI:10.1109/TPAMI.2010.23`. 51

[119] Jae Seok Park and Myung Jin Chung. Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. *IEEE Transactions on Robotics and Automation (TRA)*, 19(2):250–258, April 2003. ISSN 1042-296X. DOI:10.1109/TRA.2003.808861. 35

[120] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3D scan completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 23:1–23:10, Vienna, Austria, July 2005. Eurographics Association. ISBN 3-905673-24-X. 96

[121] Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics (TOG)*, 27(3):43:1–43:11, August 2008. ISSN 0730-0301. DOI:10.1145/1360612.1360642. 105

[122] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second National Conference on Artificial Intelligence*, AAAI-82, pages 133–136, Pittsburgh, PA, USA, August 1982. AAAI Press. 81

[123] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1988. ISBN 978-1-55860-479-7. Revised Second Printing. 81

[124] Marc Pollefeys, Luc J. Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 59(3):207–232, September 2004. ISSN 0920-5691. DOI:10.1023/B:VISI.0000025798.50602.3a. 24

[125] Duncan P. Robertson and Roberto Cipolla. An interactive system for constraint-based modelling. In Majid Mirmehdi and Barry Thomas, editors, *Proceedings of the British Machine Vision Conference (BMVC 2000)*, pages 54.1–54.10, Bristol, UK, September 2000. BMVA Press. ISBN 1-901725-13-8. DOI:10.5244/C.14.54. 51

[126] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, volume 2, pages 1508–1515, Beijing, China, October 2005. IEEE. ISBN 0-7695-2334-X. DOI:10.1109/ICCV.2005.104. 17

[127] Michael Rubinstein, Ce Liu, and William T. Freeman. Towards longer long-range motion trajectories. In Richard Bowden, John Collomosse, and Krystian Mikolajczyk, editors, *Proceedings of the British Machine Vision Conference (BMVC*

2012), pages 53.1–53.11, Surrey, UK, September 2012. BMVA Press. ISBN 1-901725-46-4. `DOI:10.5244/C.26.53`. 18

[128] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2011)*, pages 2564–2571, Barcelona, Spain, November 2011. IEEE. ISBN 978-1-4577-1101-5. `DOI:10.1109/ICCV.2011.6126544`. 18

[129] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling (3DIM 2001)*, pages 145–152, Quebec City, Canada, May 2001. IEEE. ISBN 0-7695-0984-3. `DOI:10.1109/IM.2001.924423`. 107

[130] Parvaneh Saeedi, Peter D. Lawrence, and David G. Lowe. 3D motion tracking of a mobile robot in a natural environment. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*, volume 2, pages 1682–1687, San Francisco, CA, USA, April 2000. IEEE. ISBN 0-7803-5886-4. `DOI:10.1109/ROBOT.2000.844838`. 35

[131] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision (IJCV)*, 80(1):72–91, October 2008. ISSN 0920-5691. `DOI:10.1007/s11263-008-0136-6`. 18

[132] Grant Schindler, Panchapagesan Krishnamurthy, and Frank Dellaert. Line-based structure from motion for urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2006)*, pages 846–853, Chapel Hill, NC, USA, June 2006. IEEE. ISBN 0-7695-2825-2. `DOI:10.1109/3DPVT.2006.90`. 74

[133] Stephan Scholze, Theo Moons, and Luc J. Van Gool. A probabilistic approach to roof extraction and reconstruction. In *International Archive of Photogrammetry and Remote Sensing: Proceedings of the ISPRS Commision III Symposium: Photogrammetric Computer Vision*, volume XXXIV, pages 231–236, Graz, Austria, September 2002. ISPRS. 72

[134] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):180:1–180:11, November 2012. ISSN 0730-0301. `DOI:10.1145/2366145.2366199`. 96

[135] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994)*, pages 593–600. IEEE, June 1994. ISBN 0-8186-5825-8. `DOI:10.1109/CVPR.1994.323794`. 17

[136] Yasuhito Shimizu and Jun Sato. Visual navigation of uncalibrated mobile robots from uncalibrated stereo pointers. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000)*, volume 1, pages 346–349, Barcelona, Spain, September 2000. IEEE. ISBN 0-7695-0750-6. `DOI:10.1109/ICPR.2000.905349`. 35

[137] Nicolas Simond and Patrick Rives. Trajectography of an uncalibrated stereo rig in urban environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 4, pages 3381–3386, Sendai, Japan, September 2004. IEEE. ISBN 0-7803-8463-6. `DOI:10.1109/IROS.2004.1389939`. 35

[138] Richard A. Smith, Andrew W. Fitzgibbon, and Andrew Zisserman. Improving augmented reality using image and scene constraints. In Tony Pridmore and Dave Elliman, editors, *Proceedings of the British Machine Vision Conference (BMVC 1999)*, pages 295–304, Nottingham, UK, September 1999. BMVA Press. ISBN 1-901725-09-X. `DOI:10.5244/C.13.30`. 51

[139] Henrik Stewénius and Kalle Åström. Structure and motion problems for multiple rigidly moving cameras. In Tomáš Pajdla and Jiří Matas, editors, *Computer Vision – ECCV 2004*, volume 3023 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2004. ISBN 978-3-540-21982-8. `DOI:10.1007/978-3-540-24672-5_20`. 34

[140] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 Papers*, pages 80:1–80:8, San Diego, CA, USA, July 2007. ACM. `DOI:10.1145/1275808.1276478`. 99

[141] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 438–451. Springer, 2010. ISBN 978-3-642-15548-2. `DOI:10.1007/978-3-642-15549-9_32`. 18

[142] Niko Sünderhauf, Kurt Konolige, Simon Lacroix, and Peter Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In Paul Levi, Michael Schanz, Reinhard Lafrenz, and Viktor Avrutin, editors, *Autonome Mobile Systeme 2005*, Informatik aktuell, pages 157–163. Springer, 2006. ISBN 978-3-540-30291-9. `DOI:10.1007/3-540-30292-1_20`. 35

[143] Richard Szeliski and Philip H. S. Torr. Geometrically constrained structure from motion: Points on planes. In Reinhard Koch and Luc J. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, volume 1506

of *Lecture Notes in Computer Science*, pages 171–186. Springer, 1998. ISBN 978-3-540-65310-3. DOI:10.1007/3-540-49437-5_12. 50

[144] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(11):1021–1032, November 1995. ISSN 0162-8828. DOI:10.1109/34.473228. 74

[145] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1987)*, pages 205–214, Anaheim, CA, USA, July 1987. ACM. ISBN 0-89791-227-6. DOI:10.1145/37401.37427. 94

[146] Art Tevs, Qixing Huang, Michael Wand, Hans-Peter Seidel, and Leonidas J. Guibas. Relating shapes via geometric symmetries and regularities. In *ACM SIGGRAPH 2014 Papers*, Vancouver, Canada, August 2014. ACM. To appear. 105

[147] Thorsten Thormählen. *Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge*. Dissertation, University of Hannover, 2006. Fortschritt-Berichte, Reihe 10, Nr. 765, VDI Verlag. 7

[148] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, volume 2, pages 1824–1831, Bejing, China, October 2005. IEEE. ISBN 0-7695-2334-X-02. DOI:10.1109/ICCV.2005.221. 97

[149] Philip H. S. Torr. An assessment of information criteria for motion model selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997)*, pages 47–52, San Juan, Puerto Rico, June 1997. IEEE. ISBN 0-8186-7822-4. DOI:10.1109/CVPR.1997.609296. 17

[150] Philip H. S. Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding (CVIU)*, 78:138–156, April 2000. ISSN 1077-3142. DOI:10.1006/cviu.1999.0832. 19

[151] Bill Triggs. Autocalibration and the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997)*, pages 604–619, San Juan, Puerto Rico, June 1997. IEEE. ISBN 0-8186-7822-4. DOI:10.1109/CVPR.1997.609388. 24

[152] Bill Triggs. Optimal estimation of matching constraints. In Reinhard Koch and Luc J. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 1998. ISBN 978-3-540-65310-3. DOI:10.1007/3-540-49437-5_5. 52

[153] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – a modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer, 2000. ISBN 978-3-540-67973-8. DOI:10.1007/3-540-44480-7_21. 28, 50

[154] Hoang Trinh and David McAllester. Structure and motion from road-driving stereo sequences. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops: 3D Information Extraction for Video Analysis and Mining*, pages 9–16, San Francisco, CA, USA, June 2010. IEEE. ISBN 978-1-4244-7029-7. DOI:10.1109/CVPRW.2010.5543783. 36

[155] Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. A shape hierarchy for 3d modelling from video. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia (GRAPHITE 2007)*, pages 63–70, Perth, Australia, December 2007. ACM. ISBN 978-1-59593-912-8. DOI:10.1145/1321261.1321273. 51

[156] Sundar Vedula, Simon Baker, Peter Rander, Robert T. Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the 7th International Conference on Computer Vision (ICCV 1999)*, volume 2, pages 722–729, Kerkyra, Greece, September 1999. IEEE. ISBN 0-7695-0164-8. DOI:10.1109/ICCV.1999.790293. 36

[157] George Vogiatzis, Philip H. S. Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 2, pages 391–398, San Diego, CA, USA, June 2005. IEEE. ISBN 0-7695-2372-2. DOI:10.1109/CVPR.2005.238. 72

[158] Michael Wand, Philipp Jenke, Qixing Huang, Martin Bokeloh, Leonidas J. Guibas, and Andreas Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP 2007)*, pages 49–58, Barcelona, Spain, July 2007. Eurographics Association. ISBN 978-3-905673-46-3. 96, 100

[159] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. *Computer*

Graphics Forum (CGF), 30(2):287–296, April 2011. ISSN 1467-8659. DOI:10.1111/j.1467-8659.2011.01885.x. 95

[160] Juyang Weng, Paul Cohen, and Nicolas Rebibo. Motion and structure estimation from stereo image sequences. *IEEE Transactions on Robotics and Automation (TRA)*, 8(3):362–382, June 1992. ISSN 1042-296X. DOI:10.1109/70.143354. 35

[161] Marta Wilczkowiak, Peter Sturm, and Edmond Boyer. Using geometric constraints through parallelepipeds for calibration and 3D modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(2):194–207, February 2005. ISSN 0162-8828. DOI:10.1109/TPAMI.2005.40. 51

[162] Dong-Min Woo, Dong-Chul Park, and Seung-Soo Han. Extraction of 3D line segment using disparity map. In *International Conference on Digital Image Processing (ICDIP 2009)*, pages 127–131, Bangkok, Thailand, March 2009. IEEE. ISBN 978-0-7695-3565-4. DOI:10.1109/ICDIP.2009.31. 74

[163] Kai Xu, Hanlin Zheng, Hao Zhang, Daniel Cohen-Or, Ligang Liu, and Yueshan Xiong. Photo-inspired model-driven 3D object modeling. *ACM Transactions on Graphics (TOG)*, 30(4):80:1–80:10, July 2011. ISSN 0730-0301. DOI:10.1145/2010324.1964975. 96

[164] Weiwei Xu, Jun Wang, KangKang Yin, Kun Zhou, Michiel van de Panne, Falai Chen, and Baining Guo. Joint-aware manipulation of deformable models. *ACM Transactions on Graphics (TOG)*, 28(3):35:1–35:9, July 2009. ISSN 0730-0301. DOI:10.1145/1531326.1531341. 95

[165] Xiaoming Yin and Ming Xie. Estimation of the fundamental matrix from uncalibrated stereo hand images for 3D hand gesture recognition. *Pattern Recognition*, 36(3):567–584, March 2003. DOI:10.1016/S0031-3203(02)00072-9. 35

[166] Christopher Zach, David Gallup, and Jan-Michael Frahm. Fast gain-adaptive KLT tracking on the GPU. In *CVPR Workshop on Visual Computer Vision on GPU's*, Anchorage, AK, USA, June 2008. 18

[167] Bernhard Zeisl, Pierre F. Georgel, Florian Schweiger, Eckehard Steinbach, and Nassir Navab. Estimation of location uncertainty for scale invariant feature points. In Andrea Cavallaro, Simon Prince, and Daniel C. Alexander, editors, *Proceedings of the British Machine Vision Conference (BMVC 2009)*, pages 57.1–57.12, London, UK, September 2009. BMVA Press. ISBN 1-901725-39-1. DOI:10.5244/C.23.57. 26

[168] Ye Zhang and Chandra Kambhamettu. On 3-D scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man, and*

Cybernetics, Part B: Cybernetics (TSMCB), 33(4):592–606, August 2003. ISSN 1083-4419. `DOI:10.1109/TSMCB.2003.814284`. 36

[169] Zhengyou Zhang and Gang Xu. A unified theory of uncalibrated stereo for both perspective and affine cameras. *Journal of Mathematical Imaging and Vision (JMIV)*, 9(3):213–229, November 1998. ISSN 0924-9907. `DOI:10.1023/A:1008341803636`. 35

[170] Zhengyou Zhang, Quang-Tuan Luong, and Olivier Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Transactions on Robotics and Automation (TRA)*, 12(1):103–113, February 1996. ISSN 1042-296X. `DOI:10.1109/70.481754`. 35

[171] Youyi Zheng, Hongbo Fu, Daniel Cohen-Or, Oscar Kin-Chung Au, and Chiew-Lan Tai. Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum (CGF)*, 30(2):563–572, April 2011. ISSN 1467-8659. `DOI:10.1111/j.1467-8659.2011.01880.x`. 95, 96