Purposive Three-dimensional Reconstruction by Means of a Controlled Environment

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik

der Friedrich-Schiller-Universität Jena

von **Diplom-Informatiker Michael Trummer**

geboren am **12. September 1980** in **Zwickau**

Gutachter

1. **Prof. Joachim Denzler, Friedrich-Schiller-Universität, Jena**

2. **Prof. Heinrich Niemann, Friedrich-Alexander-Universität, Erlangen-Nürnberg**

3. **Prof. Vaclav Hlavac, Czech Technical University, Prag**

Tag der öffentlichen Verteidigung: 24. Juni 2011

Versicherung an Eides statt

Gemäß §5 der heute gültigen Promotionsordnung vom 8.7.2009 der Fakultät für Mathematik und Informatik der Friedrich-Schiller-Universität Jena bestätige ich hiermit folgende Sachverhalte eidesstattlich. Die eben bezeichnete Promotionsordnung der Fakultät ist mir bekannt. Die vorliegende Dissertation habe ich persönlich angefertigt, ohne hierbei Textabschnitte oder Ergebnisse von Dritten oder aus eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen zu haben. Eine Zitierung von Textpassagen anderer Autoren ist stets mit Quellenangaben versehen. Auswahl und Auswertung des der Dissertation zugrundeliegenden Materials als auch die Erstellung des Manuskriptes erfolgten eigenständig. Dabei habe ich weder die Hilfe eines Promotionsberaters beansprucht noch geldwerte Leistungen an Dritte erbracht, die mit dem Inhalt der Promotion in Zusammenhang stehen. Die vorliegende Dissertation ist bei keiner anderen Stelle als der Fakultät für Mathematik und Informatik der Friedrich-Schiller-Universität Jena als Prüfungsleistung eingebracht oder als Dissertation eingereicht.

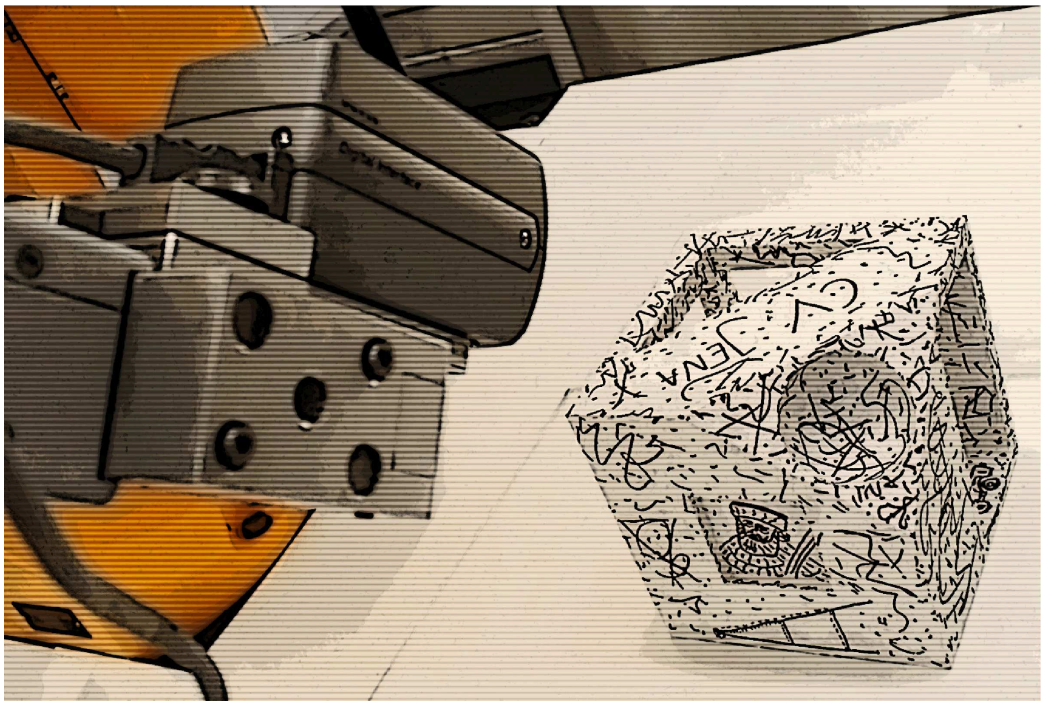
Ort, Datum Unterschrift

Abstract

Retrieving three-dimensional data using imaging devices is a relevant task for many applications in medical imaging, surveillance, industrial quality control, and others. As soon as we gain procedural control over parameters of the imaging device, we encounter the necessity of well-defined reconstruction goals and we need methods to achieve them. Hence, we enter next-best-view planning. The vast part of relevant literature discusses planning approaches to sample the whole surface of an object using a range scanning device. In this work, we present a formalization of the abstract view planning problem and deal with different planning aspects, whereat we focus on using an intensity camera without active illumination. As one aspect of view planning, employing a controlled environment also provides the planning and reconstruction methods with additional information. We incorporate the additional knowledge of camera parameters into the Kanade-Lucas-Tomasi method used for feature tracking. The resulting Guided KLT tracking method benefits from a constrained optimization space and yields improved accuracy while regarding the uncertainty of the additional input. Serving other planning tasks dealing with known objects, we propose a method for coarse registration of three-dimensional surface triangulations. By the means of exact surface moments of surface triangulations we establish invariant surface descriptors based on moment invariants. These descriptors allow to tackle tasks of surface registration, classification, retrieval, and clustering, which are also relevant to view planning. In the main part of this work, we present a modular, online approach to view planning for three-dimensional reconstruction. Based on the outcome of the Guided KLT tracking, we design a planning module for accuracy optimization with respect to an extended E-criterion. Further planning modules endow non-discrete surface estimation and visibility analysis. The modular nature of the proposed planning system allows to address a wide range of specific instances of view planning. The theoretical findings in this work are underlined by experiments evaluating the relevant terms.

Zusammenfassung

Die Rekonstruktion dreidimensionaler Modelle von Objekten unter Verwendung bildgebender Sensoren ist eine bedeutende Aufgabenstellung hinsichtlich vieler Anwendungsbereiche in der Medizin, bei der Überwachung, bei der industriellen Qualitätskontrolle und anderswo. Sobald sich die Sensorik innerhalb einer kontrollierten Umgebung befindet, also gewisse Parameter zur Laufzeit angepaßt werden können und müssen, befindet man sich im Anwendungsfeld der Ansichtenplanung. Die Ansichtenplanung zur dreidimensionalen Rekonstruktion innerhalb einer kontrollierten Umgebung umfaßt Methoden mit der Zielstellung, wohldefinierte Rekonstruktionsziele durch absichtsvolle Wahl der Parameter der kontrollierten Umgebung zu erreichen. Ein Großteil der Literatur zur Ansichtenplanung beschäftigt sich mit der Aufgabe, die Oberfläche eines Objektes mit einem Tiefensensor vollständig abzutasten. Diese Arbeit befaßt sich, basierend auf einer abstrakten Formalisierung, mit verschiedenen Teilaspekten der Ansichtenplanung, wobei die Verwendung einer Intensitätskamera ohne aktive Lichtprojektion angenommen wird. Ein Aspekt der Ansichtenplanung ist die Anwendung zusätzlicher Information, die durch die kontrollierte Umgebung geliefert wird. Hinsichtlich dessen stellt diese Arbeit eine Erweiterung der Merkmalsverfolgung nach Kanade, Lucas und Tomasi vor. Als Ergebnis nutzt die erweiterte Merkmalsverfolgung Wissen über die Kameraparameter zur Einschränkung des Suchraumes der Optimierung unter Berücksichtigung von Unsicherheit, was erhöhte Genauigkeit zur Folge hat. Des weiteren präsentiert diese Arbeit ein Verfahren zur Grobregistrierung von dreidimensionalen Oberflächentriangulationen mittels Momentinvarianten. Zu diesem Zweck werden exakte Oberflächenmomente berechnet, welche die Grundlage bilden zur Erstellung von Momentinvarianten. Die so erzeugten Merkmale von Oberflächentriangulationen können ebenfalls in Klassifikations- und Erkennungsaufgaben Anwendung finden, so zum Beispiel in der modellbasierten Ansichtenplanung, die Planungsziele bezüglich bekannter Objektmodelle umsetzt. Im Hauptteil der Arbeit wird ein modularer, schritthaltender Ansatz zur Ansichtenplanung bezüglich verschiedenster Teilziele vorgestellt. Basierend auf der erweiterten Merkmalsverfolgung wird eine Methode zur Genauigkeitsoptimierung entwickelt. Weitere Planungsmodule leisten die Schätzung einer kontinuierlichen Oberfläche und Sichtbarkeitsanalyse, wodurch der Weg zum Planungsziel Vollständigkeit bereitet wird. Der modulare Aufbau des Planungssystems bietet die Möglichkeit, viele konkrete Instanzen des abstrakten Problems Ansichtenplanung zu bearbeiten. Die theoretischen Ergebnisse dieser Arbeit werden durch Experimente und Auswertung der relevanten Größen untermauert.



Owing to the fact that all experience is a process, no point of view can ever be the last one.

William James

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory chapter provides motivating thoughts concerning this entire work as well as a concise problem specification. We grow the plot of this work, give reasoning for its structure, and describe further framing aspects. In addition, we endeavor to elaborate strictly theoretical approaches to the task of next-best-view (NBV) planning, and reveal their implications and practical boundaries.

1.1 3D Reconstruction and View Planning

The basic topic of this work is the task of three-dimensional (3D) reconstruction of a rigid object, given intensity images of this specific object. Mapping an object to an intensity image reduces the dimension of the representation from 3D to 2D, which makes the inverse – the 3D reconstruction – hard to determine. Without further information, we have to use more than one image of the object to estimate its 3D shape. In addition, we assume a positional difference, i. e. a translation between the imaging intensity cameras, whereat one moving camera is equivalent to two stationary cameras assuming a constant appearance of the scene. This problem formulation is known as the *structure-from-motion* problem. In general, structure-from-motion approaches try to estimate the 3D shape of an object. Simultaneously recovering the camera poses and internal properties is called *self-calibration*. Either way, such approaches use *correspondences* of image points, cf. [Hartley and Zisserman, 2003] page 238 and [Faugeras and Luong, 2001] page 19, which are mappings of the same 3D world points, respectively.

Figure 1.1: An intensity camera mounted on a robotic arm serving as controlled environment.

Many methods for 3D reconstruction make use of additional prior knowledge about the cameras. For instance, the knowledge of the internal camera properties facilitates to apply the Tomasi-Kanade factorization [Tomasi and Kanade, 1992] which yields camera motion and object shape. The case of known cameras, in particular internal properties and camera motion, allows standard triangulation to compute the 3D object shape, cf. [Hartley and Zisserman, 2003] page 312. These examples illustrate the incorporation of prior knowledge for the benefit of accuracy, robustness, the simplicity of computation, or optimality assertions. Chapter 3 gives a more in-depth survey of approaches to 3D reconstruction with and without using prior knowledge.

Now, we go one step further and allow to control the camera parameters during the reconstruction procedure. In doing so, we enter *active* 3D reconstruction. Within this work, the concept of active reconstruction does not touch special lighting conditions such as structured ligth patterns, but remains in the control of camera parameters. Figure 1.1 shows a controlled environment consisting of an intensity camera mounted on a robotic arm.

The work at hand focuses on view planning, i. e. active 3D reconstruction, using an intensity camera without controlled illumination, which we call *passive camera* in the following. As we show in Chapter 2, there are many view planning approaches for different kinds of range scanners, including laser scanners and fringe projection systems. However, the use of a passive camera is worthy of a more thorough examination for several reasons. First, the price tag compared to range sensors is smaller for a passive camera. Second, range scanners such as fringe projection systems cannot cope with severe optical structure on the object surface, since the optical structure interferes with the light pattern. This is exactly the field of application where passive cameras use the optical structure to solve the correspondence problem, which also promotes combined sensor systems for view planning. Third, the wide application of passive cameras in all-day life favors to transfer and scale view

planning elements to related areas of interest.

In the following section we elaborate the consequences of using a controlled environment for 3D reconstruction. We formalize the task and argue that the additional controlling ability implies the need for explicitly defined reconstruction goals.

1.1.1 Problem Existence and Motivation

A controlled environment, like a camera mounted on a robotic arm as shown in Figure 1.1, provides the reconstruction procedure with additional prior knowledge and active controlling abilities. These controlling abilities require to purposively adjust the camera parameters during the reconstruction procedure, which we call Next-best-view Planning or simply View Planning.

Initially, let us assume that we use a controlled environment, but do not make any efforts to adapt our reconstruction procedure for purposive actions, i. e. we do not apply view planning. In this case, we still want to avoid random camera movements in an at least 6D parameter space of 3D rotation and 3D translation, and we want to avoid a complete sampling of this continuous 6D parameter space for obvious reasons. A human expert could now find a simple non-random solution that leads the camera on a helix path around the object, while the camera is directed to the object at each position. The expert fixes the parameters of this path in order to achieve, for instance, a complete 3D reconstruction of the object with a constant large number of views, or to perform a quick reconstruction. For the former case, the reconstruction algorithm is likely to process lots of data with only little contribution to the result, which also means heavy load for the data fusion; the latter case, on the contrary, is likely to miss details of the 3D object shape. In either case, the first reconstruction attempt may miss the implicitly designated goals, which would necessitate further runs and, hence, further costs.

From the simple example described above we state the following observations.

1. We use a controlled environment for 3D reconstruction.

2. We do not aim at a random camera path, but we also work *without using a view planning algorithm.*

3. We have to move the camera somehow, so we consider what we want to achieve, and we try to figure out how to achieve our findings.

4. We try to reach our goals by pre-plotting a camera path and assess the result.

5. We are likely to find our goals unmet, at least we do not know exactly.

6. We adjust the plot of our camera path and hope to improve the result in further runs.

7. We want to better adjust the camera parameters for a certain purpose, in fact we *do view planning*.

Utilizing the above observations, we draw the following conclusions.

- Our observations contain a contradiction; we started by assuming that we do not use view planning, but in the end we realized that we actually did. We resolve this contradiction by the following assertion: *If we want to carry out non-random 3D reconstruction within a controlled environment, we implicitly perform at least a simple kind of view planning.*

- View planning should optimize the reconstruction procedure as well as the result in terms of formalized, pre-defined reconstruction goals, making obsolete any user interaction during the reconstruction process.

- We can think of contradictory reconstruction goals. Thus, view planning requires the user to define the planning goals and constraints consistently.

- View planning needs to be data-driven in order to serve as a valid approach for arbitrary objects. Otherwise, the planning method would not be able to incorporate information that deviates from the intial assumptions.

In summary we note that any 3D reconstruction using a controlled environment includes a more or less sophisticated way of view planning. Accordingly, a view planning method should incorporate data feedback in order to meet well-defined reconstruction goals in an optimal manner. We express our understanding of the view planning essentials in the abstract scheme of Figure 1.2, which we are going to formalize in the next section.

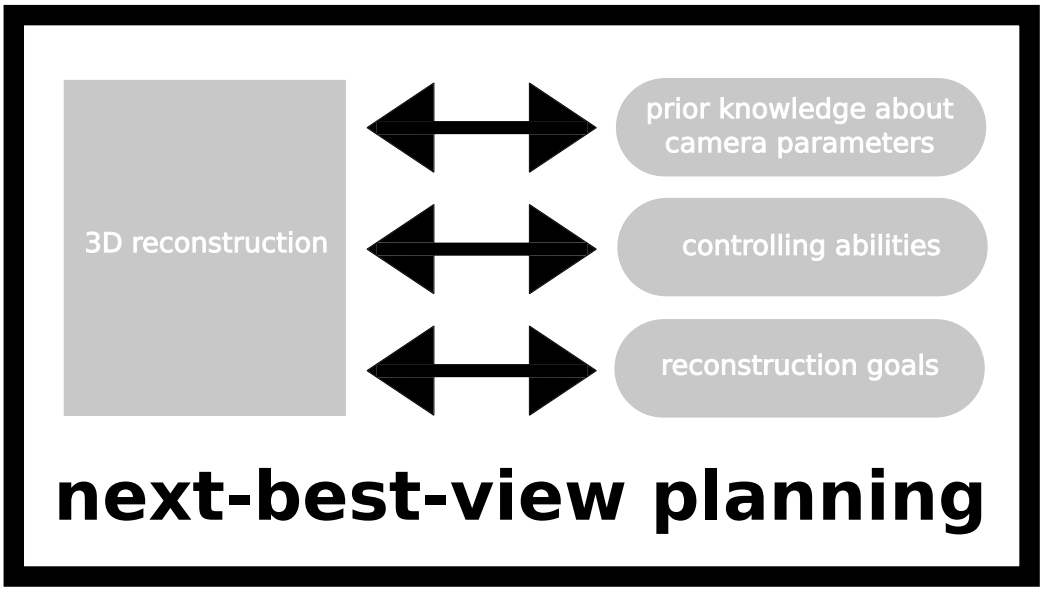


Figure 1.2: An abstract scheme of view planning essentials. The method of view planning comprises 3D reconstruction and relates it to prior knowledge, uses controlling abilities, and optimizes reconstruction procedure and result with respect to well-defined reconstruction goals.

1.1.2 Problem Formalization

With regard to standard methods for 3D reconstruction, view planning is a high-level approach that parameterizes the reconstruction procedure for a certain purpose. Likewise, view planning adjusts this parameterization to the input data during the reconstruction process. In the following we present a possible way to formalize next-best-view planning as a constrained optimization procedure.

Optimization goals and constraints. The purpose of the 3D reconstruction is to be defined by a set G of planning goals,

$$\mathsf{G} = \{g_1, g_2, \ldots\}\,, \tag{1.1}$$

whereat the goals g_i are formulated as equations, inequalities, or optimization problems. For instance, we can aspire to reconstruct points of a certain part of an object, to improve our reconstruction as long as the accuracy is below a certain value, or to produce a maximal accuracy, respectively. While some

planning goals may bring along an infinite optimization process, we find it necessary to apply a set C of constraints,

$$\mathsf{C} = \{c_1, c_2, \ldots\}, \tag{1.2}$$

that, for example, limit the number of views or demand a minimal change of some measure between consecutive views. The consistent definitions of G and C are due to the user and will not be addressed by this work.

Additional prior knowledge. Using a controlled environment provides prior knowledge K,

$$\mathsf{K} = \{k_1, k_2, \ldots\}. \tag{1.3}$$

An example is the case of given camera rotations and translations for each view.

In addition to the information K provided by the environment, we also know the current state of the **reconstructed 3D data** stored in a set R, which is important to data-driven view planning.

Active control over camera parameters. Another important aspect of a controlled environment is, trivially, the ability to actively control a set P of parameters,

$$\mathsf{P} = \{p_1, p_2, \ldots\}. \tag{1.4}$$

It is likely that we find P and K coming from the same basic set. Observing the fact that we know at least the parameters that we have actively adjusted, K contains the resulting parameters of previous planning steps.

Further we formally introduce the respective notations for view v. Given the additional prior knowledge $\mathsf{K}^{(v)}$, the view planning method yields a set of parameters $\mathsf{P}^{(v)}$ and computes a 3D reconstruction $\mathsf{R}^{(v)}$. Having taken $v \geq 2$ views, a 3D reconstruction $\mathsf{R}^{(k)}$ with $k > v$ denotes an expected 3D reconstruction $\widehat{\mathsf{R}}^{(k)}$, which can also be seen as an a priori estimation. Without using view indices, we denote the expected 3D reconstruction by $\widehat{\mathsf{R}}$.

View Planning as an overall, constrained optimization. As stated above, let R be a set comprising all reconstructed data at the current state, which are, for instance, 3D points, 3D triangles, and further attributes to these entities. View planning then tries to optimally transform R in order to reach the planning goals G while sustaining the constraints C. Here, optimality

can be associated both with the process and the result of view planning and 3D reconstruction. We simplify this fact by just pointing out that either of these optimality criteria may be formulated in terms of planning goals and constraints. Thus, by defining a suitable distance measure $d_{\mathsf{G}}(\mathsf{R}) \rightarrow \mathbb{R}$ we are able to assess the reconstruction qualities with regard to the planning goals. Eventually, view planning adjusts the parameters in P in order to minimize the distance measure d_{G} applied to the expected 3D reconstruction $\widehat{\mathsf{R}}$. As a central statement of this work we conclude that one-step next-best-view planning solves the constrained optimization problem

$$\arg\min_{\mathsf{P}} d_{\mathsf{G}}\left(\widehat{\mathsf{R}}\right) \quad \text{subject to} \quad \mathsf{C}, \tag{1.5}$$

where the reconstruction R depends on the actively chosen parameters P.

The expression in (1.5) is abstract enough to be valid for any kind of sensing technique, planning goals, and constraints. Still, the actual objective function $d_{\mathsf{G}}(\widehat{\mathsf{R}})$ has to be chosen individually with respect to particular planning goals and reconstruction representations. The design and computation of this objective function is one of the great challenges in developing view planning methods.

Having found an optimal set of parameters P^*, we are able to actively adjust our controlled environment accordingly. After the following sensing process, our reconstruction data gathered in R is updated. By this means, different choices of P influence the objective function in (1.5). We formalize the way of computing the one set of optimal parameters for the one next best view, so we regard one-step planning. Planning multiple steps can be formalized as optimizing the objective function from (1.5) for a sequence, or tuple, of n parameter sets starting from view v,

$$\arg\min_{(\mathsf{P}^{(v+1)},\ldots,\mathsf{P}^{(v+n)})} d_{\mathsf{G}}\left(\widehat{\mathsf{R}}^{(v+n)}\right) \quad \text{subject to} \quad \mathsf{C}. \tag{1.6}$$

While this formalization, naturally, is quite abstract, the next section will give practical examples on view planning and describe the formulation of the terms introduced above.

1.1.3 Applications of Solutions

Now, we take a closer look at actual view planning problems. We apply our formalization, and we describe possible applications for view planning

solutions. The following examples shall both fortify feasibility and provide handy prospect of our formal definitions in Section 1.1.2.

Example A. One classical field of application for controlled environments is industrial quality control. The task is to assess the quality of an object that has been manufactured by machines based on a 3D CAD model. For this example, we assume a controlled environment as illustrated in Figure 1.1, i. e. a robotic arm and an intensity camera mounted upon. The motivation to use view planning here is given by several reasons. First, a quality assessment needs time and hardware usage, both of which cause costs that are to be minimized. Second, a quality assessment shall provide an objective quality measure that is reproducible in further runs. The latter cannot be achieved by human experts planning the camera positions that seem optimal to them. Formally, the reconstructed data $\mathsf{R}_A^{(v)}$ after view v comprises the K 3D points to be restored and their covariance matrices,

$$\mathsf{R}_A^{(v)} = \left\{ (\mathsf{X}^{(v)}, \mathsf{S}^{(v)}) \right\}, \tag{1.7}$$

$$\mathsf{X}^{(v)} = \left\{ \mathbf{X}_1^{(v)}, \ldots, \mathbf{X}_K^{(v)} \right\}, \tag{1.8}$$

$$\mathsf{S}^{(v)} = \left\{ \mathbf{\Sigma}_1^{(v)}, \ldots, \mathbf{\Sigma}_K^{(v)} \right\}. \tag{1.9}$$

The planning goal is to limit the maximal reconstruction uncertainty, for example, a suitable standard deviation $\sigma_k^{(v)}$, $k = 1, \ldots, K$, retrieved over all K points, hence

$$g_{A_1} : \max_k \widehat{\sigma}_k^{(v+1)} \leq b, \tag{1.10}$$

$$\mathsf{G}_A = \{ g_{A_1} \}, \tag{1.11}$$

with a threshold b. Since this planning goal is mandatory to achieve comparable 3D reconstructions, we do not apply any constraints,

$$\mathsf{C}_A = \emptyset. \tag{1.12}$$

In order to meet the planning goal, the view planning method uses the robotic arm to adjust camera rotation and translation $(\mathbf{R}^{(v+1)}, \mathbf{t}^{(v+1)})$, thus the actively configured parameters are

$$\mathsf{P}_A^{(v+1)} = \left\{ \left(\mathbf{R}^{(v+1)}, \mathbf{t}^{(v+1)} \right) \right\} \tag{1.13}$$

with view v being the last one taken. The parameters of all previous views are assumed to be known by the planning method,

$$\mathsf{K}_A^{(v+1)} = \left\{ \left(\mathbf{R}^{(k)}, \mathbf{t}^{(k)} \right) : k = 1, \ldots, v \right\}. \tag{1.14}$$

Finally, the objective function $d_{\mathsf{G}_A}(\widehat{\mathsf{R}}_A^{(v+1)})$ evaluates the distance between b and the largest expected standard deviation $\widehat{\sigma}_k^{(v+1)}$,

$$d_{\mathsf{G}_A}\left(\widehat{\mathsf{R}}_A^{(v+1)}\right) = \begin{cases} \max_k \widehat{\sigma}_k^{(v+1)} - b & , \max_k \widehat{\sigma}_k^{(v+j)} > b,\ j = 0, 1 \\ 0 & , \text{else} \end{cases} \tag{1.15}$$

For $\widehat{\sigma}_k^{(v+j)}$ and $j = 0$ the expected variance equals the one observed, which provides a further condition for termination. By this means, a proper view planning method continues the reconstruction process until the standard deviation of each point is smaller than or equal to the threshold b. This is done by maximally reducing the maximal expected variance at each step.

Example B. As a second example we consider the task of computing a 3D city model using a mobile laser scanning system. A mobile laser scanner can be attached to a car or can be lifted upon roofs, if necessary. In general, laser scanners produce large amounts of data that challenge any data fusion method. For this reason and in order to keep moving costs low, we want to optimize the completeness of our 3D reconstruction while not exceeding a number of $v_{\max}$ views. Formalizing completeness claims is a difficult task due to the discrete nature of 3D point reconstruction, as we will outline in Section 6.1.3.2. That is why we keep this example abstract enough to assume that we know a function $c(\mathsf{R}_B) \to [0, 1]$ measuring the completeness of reconstruction R_B. The 3D data $\mathsf{R}_B^{(v)}$ after view v stores the reconstructed 3D points as well as continuous surface information in the form of 3D triangles, hence

$$\mathsf{R}_B^{(v)} = \left\{ \left(\mathsf{X}^{(v)}, \mathsf{T}^{(v)} \right) \right\} \tag{1.16}$$

with a set of M points and a set of N triangles given by the indices of their corner points,

$$\mathsf{X}^{(v)} = \left\{ \mathbf{X}_1^{(v)}, \ldots, \mathbf{X}_M^{(v)} \right\}, \tag{1.17}$$

$$\mathsf{T}^{(v)} = \left\{ \mathsf{T}_1^{(v)}, \ldots, \mathsf{T}_N^{(v)} \right\} = \left\{ \left\{ i_{11}^{(v)}, i_{12}^{(v)}, i_{13}^{(v)} \right\}, \ldots, \left\{ i_{N1}^{(v)}, i_{N2}^{(v)}, i_{N3}^{(v)} \right\} \right\}, \tag{1.18}$$

$i_{kl}^{(v)} \in \mathbb{N}$, $1 \leq i_{kl}^{(v)} \leq M$, $k, l \in \mathbb{N}$, $1 \leq k \leq N$, and $1 \leq l \leq 3$. The additional prior knowledge $\mathsf{K}_B^{(v+1)}$ includes the sensor position $\mathbf{t}^{(v)}$ and sampling density $s^{(v)}$, whereat $\mathbf{t}^{(v)}$ may be known from an integrated GPS system, and $s^{(v)}$ is a hardware parameter of the laser scanner. Assuming we already have taken v views, we yield

$$\mathsf{K}_B^{(v+1)} = \left\{ \left(\mathbf{t}^{(k)}, s^{(k)} \right) : k = 1, \ldots, v \right\}. \tag{1.19}$$

Our goal is to optimize

$$\mathsf{P}_B^{(v+1)} = \left\{ \left(\mathbf{t}^{(v+1)}, s^{(v+1)} \right) \right\} \tag{1.20}$$

in order to achieve maximal completeness,

$$g_{B_1} : \max c \left(\widehat{\mathsf{R}}_B^{(v+1)} \right), \tag{1.21}$$

$$\mathsf{G}_B = \{ g_{B_1} \}. \tag{1.22}$$

We restrict ourselves to take not more than $v_{\max}$ views, so

$$c_{B_1} : v + 1 \leq v_{\max}, \tag{1.23}$$

$$\mathsf{C}_B = \{ c_{B_1} \}. \tag{1.24}$$

Using these terms, we state the view planning objective function

$$d_{\mathsf{G}_B} \left(\widehat{\mathsf{R}}_B^{(v+1)} \right) = 1 - c \left(\widehat{\mathsf{R}}_B^{(v+1)} \right). \tag{1.25}$$

Again, a suitable view planning method solves the constrained optimization problem from (1.5) and, in this case, optimizes the completeness of the 3D reconstruction within the constrained number of views.

The above examples formalize and illustrate two practical applications of next-best-view planning, and they also emphasize the need to use a controlled environment in such a purposive manner. Otherwise we compare reconstruction results that are not comparable (example A), and we waste effort and energy by collecting more data than actually necessary (example B). Furthermore, the examples show that the actual challenges in view planning come with the concrete formulation of the planning goals and the view planning objective function. In doing so, we have to answer the following questions.

- How do we represent our 3D reconstruction R?
- How do we express our planning goals with respect to R?
- What does an algorithm look like that actually optimizes the planning goals in some sense?

Once we have answered these questions for a certain problem formulation, it is possible to *transfer* and *scale* the solution to other problems. We find it worthy to point this out; while a view planning method provides a quite specific solution to a specific problem, it is still not confined to this initial problem, in general. For example, a view planning method solving the problem from example A may also give a valuable foundation to another application, that deals with optimizing the reconstruction accuracy using controlled airborne sensors. There are many other controlled imaging environments that allow for view planning approaches or, in our understanding, make a well-defined view planning necessary. These devices include solutions for industrial quality control, medical imaging, robots for hazardous and hostile areas, surveillance systems, and others, cf. Figure 1.3. Still, the above stated questions remain. In the next section we try to answer them from a strictly theoretical point of view.

1.2 Theoretical Approaches to View Planning

This section gives place to theoretical considerations about how to tackle the view planning optimization problem. In opposition to Chapter 2, where we provide a literature survey of known methods, the following lines describe mathematical ideas for next-best-view planning. We start these considerations regardless of practical circumstances and feasibility. Of course, we could find innumerous theoretical approaches without stating further restrictions. Instead, we concentrate ourselves on nearby ideas and have a look at interpolation and probabilistic estimation. Without getting lost in too much detail, we describe view planning in terms of the respective vantage point and examine limitations as well as implications in the context of this work.

1.2.1 Sampling and Interpolation

Given a real-world surface $\mathfrak{S}$, we take samples of this surface and compute an interpolated surface $\mathcal{S}$, which approximates $\mathfrak{S}$. This is the paradigm

Figure 1.3: Examples of controlled imaging environments. The areas of application comprise robots for hazardous and hostile areas, surveillance, assistance and entertainment, medical imaging, industrial quality control, and others.

of restoring a non-discrete function taking sample values. For the sake of simplicity and without loss of generality, we refer to 1D surfaces. If $\mathfrak{S}$ is continuous, we can find a polynomial surface that approximates $\mathfrak{S}$ inside an interval arbitrarily exact, which is stated by the approximation theorem of Weierstraß, cf. [Jeffreys and Jeffreys, 1988] page 446. Thus, the whole continuous surface $\mathfrak{S}$ finds an arbitrarily exact approximation by a polynomial or a piecewise polynomial representation.

Interpolation is based on sampling positions x_n that provide sampling values $\mathfrak{S}(x_n) = y_n$, $0 \leq n \leq N-1$. Using, for example, Lagrange polynomials, we achieve

$$\mathcal{L}(x) = \sum_{n=0}^{N-1} l_n(x), \quad l_n(x) = y_n \prod_{\substack{k=0 \\ k \neq n}}^{N-1} \frac{x - x_k}{x_n - x_k}, \tag{1.26}$$

[Jeffreys and Jeffreys, 1988] page 260. By setting $\mathcal{S}_L(x) = \mathcal{L}(x)$ we get $\mathfrak{S}(x_n) = y_n = \mathcal{S}_L(x_n)$ and, hence, a $(N-1)^{\text{th}}$-order smooth approximation of N points from $\mathfrak{S}$. Another way of interpolation is given by the Fourier transformation, [Bose, 2003] page 75, [Süße, 20xx], which takes a periodic function and transforms it into a trigonometric series. The periodic, discrete function is given by the samples $y(n) = \mathfrak{S}(x_n) = y_n$, $0 \leq n \leq N-1$, whereat we assume N-periodicity in the shape of $y_n = y_{n+kN}$, $k \in \mathbb{Z}$, as a theoretical prerequisit. Further we define discrete basis functions

$$\phi_n^{(k)} = \phi^{(k)}(n) = \frac{1}{\sqrt{N}} \mathrm{e}^{2\pi \mathrm{i} k \frac{n}{N}} \tag{1.27}$$

and the inner product of functions $\langle \cdot, \cdot \rangle$,

$$\langle f, g \rangle = \sum_{k=0}^{N-1} f_k \overline{g_k}, \tag{1.28}$$

where $\overline{g_k}$ is the conjugate complex of g_k. By this means we express the discrete Fourier transform of our periodic function y by computing the finit-number, discrete Fourier coefficients

$$\alpha_k = \langle y, \phi^{(k)} \rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} y_j \mathrm{e}^{-2\pi \mathrm{i} k \frac{j}{N}}. \tag{1.29}$$

The inverse Fourier transform

$$y(n) = y_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \alpha_k \mathrm{e}^{2\pi \mathrm{i} k \frac{n}{N}} \tag{1.30}$$

reproduces the original function y using the Fourier coefficients α_k, the *frequencies* of y. Eventually, we yield a trigonometric interpolation $\mathcal{S}_T(x)$ of the discrete function y by replacing the argument n in (1.30) with a real-valued $x \in [x_0, x_{N-1}]$,

$$y(x) = \frac{1}{\sqrt{N}} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \alpha_k \mathrm{e}^{2\pi \mathrm{i} k \frac{x}{X}} = \mathcal{S}_T(x). \tag{1.31}$$

This interpolation holds for $X = x_{N-1} - x_0$ and equidistant sampling positions $x_n = n\frac{X}{N}$, $0 \leq n \leq N-1$, which can be seen from simply substituting the positions x in (1.31). At this point, we may focus on the number N of samples necessary to achieve an exact reconstruction of $\mathfrak{S}$, which leads to the well-known *sampling theorem*, cf. [Bose, 2003] page 97. Given the continuous function $\mathfrak{S}(x)$ and $k_{\max}$ with the analog Fourier coefficients $\beta_k = 0$, $|k| > k_{\max}$, the sampling theorem states that we can exactly restore the original function $\mathfrak{S}(x)$ using at least $N_{k_{\max}} = 2k_{\max} + 1$ equidistant samples (x_n, y_n), $0 \leq n \leq N_{k_{\max}}$. In this case the original function $\mathfrak{S}(x)$ equals the trigonometric interpolation polynom $\mathcal{S}_T(x)$ from (1.31), and $k_{\max}$ is the maximal frequency limiting the bandwith of the original function.

The theoretical consequences in the context of view planning are the following. In order to yield a true reconstruction of the real-world surface, we have to sample the surface at a number of positions that depends on its limiting frequency. Since this limiting frequency is unknown in general, we do not know about the necessary number of samples. Nonetheless, the sampling positions should be equidistant on the surface in order to confine the well-known interpolation effect of uncontrolled oscillation. In this case, there is nothing left to do for the view planning method, also since the interpolation model does not regard camera positions. Another crucial assumption is a continuous real-world surface, which does not hold for many technical objects containing corners or steps. Such discontinuities cause high-frequency reconstruction errors known as the Gibbs phenomenon, cf. [Bose, 2003] page 232.

With regard to practical conducting, we have to take the passive imaging sensor into account. Since passive cameras measure light intensities, we are

not able to directly sample a 3D surface point. Furthermore, a passive camera cannot sense any point on the surface, because sufficient optical structure is a condition for solving the correspondence problem and thereby performing 3D reonstruction. So, the view planning method is not free to choose optimal sampling positions. In conclusion, the approach of sampling and interpolation shows to be inappropriate for view planning using a passive camera. While the approach remains an interesting theoretic consideration, the abstract pattern is not compatible with view planning using a passive camera. What we learn from this consideration is the fact that the actual sampling process itself is a challenging part in the context of this work, since sampling values are not just given by the sensor. This sampling process, which computes 3D surface information from passive sensor data, requires to solve the 2D correspondence problem as well as a proper camera motion optimized by view planning. The theoretical approach assumes exact samples (x_n, y_n) to be given. In the context of this work, the surface sampling, which is actually an indirect sampling by combining information of different domains, is an imperfect process and should as well be optimized by a suitable view planning method.

1.2.2 Probabilistic State Estimation

While the framework of sampling and interpolation does not provide the means to serve view planning, another reasonable theoretic approach is probabilistic state estimation, see [Denzler, 2003] and [Fisz, 1980] page 536. We refer to a steady system that owns a steady state $\mathbf{q} \in \mathbb{R}^s$. At time t, $1 \leq t \leq T$, we make an observation $\mathbf{o}_t \in \mathbb{R}^o$ depending on the system state $\mathbf{q}$, but we are not able to observe the system state directly. The observation $\mathbf{o}_t$ is given by an observation function,

$$\mathbf{o}_t = h_t(\mathbf{q}, \mathbf{w}_t) \tag{1.32}$$

that takes into account the system state $\mathbf{q}$ and noise $\mathbf{w}_t$, which is characterized by its probability density function $p(\mathbf{w}_t)$. It is this probabilistic noise modeling that prohibits a direct analytic solving of (1.32) to yield $\mathbf{q}$. Consequently, an estimator $\kappa(\cdot)$ uses the sequence of observations $\langle \mathbf{o} \rangle_t = \langle \mathbf{o}_1, \ldots, \mathbf{o}_t \rangle$ to produce an estimate $\hat{\mathbf{q}}$ of the hidden state $\mathbf{q}$, so

$$\widehat{\mathbf{q}}_t = \kappa(\langle \mathbf{o} \rangle_t). \tag{1.33}$$

The task is then to find an estimator κ that is *unbiased*, i. e. the expectation value of the estimate equals the true state, $\mathbb{E}(\widehat{\mathbf{q}}) = \mathbf{q}$. Additionally, the estimator shall be *consistent*, which expresses an asymptotically vanishing variance $\mathbb{V}(\widehat{\mathbf{q}})$, i. e. $\lim_{t\to\infty}\mathbb{V}(\widehat{\mathbf{q}}_t) = \lim_{t\to\infty}\mathbb{E}((\widehat{\mathbf{q}}_t - \mathbf{q})(\widehat{\mathbf{q}}_t - \mathbf{q})^{\mathsf{T}}) = \mathbf{0}$. As for estimation methods, we distinguish whether the incorporation of prior knowledge is modeled or not. A well-known estimation method not using prior information is maximum likelihood estimation. Here, the estimator $\kappa^{(\mathrm{ML})}$ selects the state that maximizes the likelihood function $\Lambda(\mathbf{q})$,

$$\widehat{\mathbf{q}}_t^{(\mathrm{ML})} = \kappa^{(\mathrm{ML})}(\langle\mathbf{o}\rangle_t) = \arg\max_{\mathbf{q}} \Lambda(\mathbf{q}) = \arg\max_{\mathbf{q}} p(\langle\mathbf{o}\rangle_t|\mathbf{q}). \tag{1.34}$$

An example of using prior information in the shape of a probabilistic density function $p(\mathbf{q})$ is given by the maximum a posteriori estimator $\kappa^{(\mathrm{MAP})}$. To achieve this, we apply the Bayesian rule,

$$p(\mathbf{q}|\langle\mathbf{o}\rangle_t) = \frac{p(\langle\mathbf{o}\rangle_t|\mathbf{q})p(\mathbf{q})}{p(\langle\mathbf{o}\rangle_t)}, \tag{1.35}$$

which relates the prior density $p(\mathbf{q})$, the likelihood $\Lambda(\mathbf{q}) = p(\langle\mathbf{o}\rangle_t|\mathbf{q})$, and the a posteriori density $p(\mathbf{q}|\langle\mathbf{o}\rangle_t)$. By means of (1.35) we are able to state the maximum a posteriori estimator as

$$\widehat{\mathbf{q}}_t^{(\mathrm{MAP})} = \kappa^{(\mathrm{MAP})}(\langle\mathbf{o}\rangle_t) = \arg\max_{\mathbf{q}} p(\langle\mathbf{o}\rangle_t|\mathbf{q})p(\mathbf{q}), \tag{1.36}$$

since $p(\langle\mathbf{o}\rangle_t)$ is independent of $\mathbf{q}$. The linkage between ML and MAP estimation shows up if we assume a uniformly distributed state prior $p(\mathbf{q})$, which actually symbolizes the absence of any prior information. In this case, the prior part in (1.36) does not depend on $\mathbf{q}$ and, hence, can be deleted from the objective function, which transforms (1.36) into (1.34). Thus, MAP estimation can be seen as a regularization of ML estimation by the prior $p(\mathbf{q})$.

For view planning, we first have to decide which entity shall be modeled by the system state $\mathbf{q}$.

Model I. A natural choice seems to be if we define the true 3D reconstruction $\mathfrak{R}$ as the steady system state. We formally switch the index from time to parameter domain. That means we make an observation $\mathbf{o}_{\mathsf{P}_t}$ depending on a set of actively adjusted parameters P_t. Likewise, all other variables originally marked by t do now depend on P_t. Hence we make estimates $\widehat{\mathsf{R}}_{\mathsf{P}_t}$

depending on the observed images $\mathbf{o}_{\mathsf{P}_1}$, $\mathbf{o}_{\mathsf{P}_2}$, ..., $\mathbf{o}_{\mathsf{P}_t}$. An appropriate prior $p_{\mathsf{K}}(\mathfrak{R})$ models the influence of additional prior knowledge K, thus establishing a maximum a posteriori framework as in (1.36). Right now, one flaw of this modeling is that our actual planning objective (1.5) is not regarded. So we can try to include the planning error $d_{\mathsf{G}}(\widehat{\mathsf{R}}_{\mathsf{P}_t})$ into the observation $\mathbf{o}_{\mathsf{P}_t}$. This requires us to model a proper likelihood $p(\mathbf{o}_{\mathsf{P}_1}, \ldots, \mathbf{o}_{\mathsf{P}_t}|\mathfrak{R})$ that refers the images and the planning objective to the unknown true 3D reconstruction. While finding a practical solution for this likelihood seems unlikely enough, one final problem remains. A MAP estimator modeled in this way optimizes the estimate $\widehat{\mathsf{R}}_{\mathsf{P}_t}^{(MAP)}$ of the true 3D reconstruction $\mathfrak{R}$, which is not a solution to the planning objective (1.5).

Model II. Learning from the shortcomings of our first modeling, we now render the optimum P_t^* of the planning objective (1.5) as the system state $\mathbf{q}_t$. Thus, we switch from static to dynamic state estimation and try to find the optimal set of parameters for the next view. As observations we state the sequence of 3D reconstructions $\langle \mathsf{R} \rangle_t$, while the additional prior planning knowledge is represented by the estimation prior $p_{\mathsf{K}}(\mathsf{P}_t)$. The likelihood $p(\langle \mathbf{o} \rangle_t | \mathbf{q}_t, \langle \widehat{\mathbf{q}} \rangle_{t-1})$, which also takes the former system states into account for the dynamic case, actually relates the set of parameters to adjust $\widehat{\mathsf{P}}_t^* = \widehat{\mathbf{q}}_t$ to the current 3D reconstruction $\mathsf{R}_t = \mathbf{o}_t$. A drawback of this modeling is that the 3D reconstruction is stated as an absolute observation and is not included in the optimization process.

Compared to the approach of sampling and interpolation, the optimization method of probabilistic state estimation provides better means to model the complex task of view planning for 3D reconstruction. However, the standard approach of state estimation as well does not allow to directly represent all inherent relations. Further it is a hard problem itself to find an actual optimization procedure to calculate an MAP estimate. In particular, such an optimization procedure surely depends on the current specification of the planning goals G and constraints C, which imposes additional difficulty on the MAP framework. While certain specific and constrained view planning tasks may find an appropriate MAP solution, the MAP estimation does not directly provide the means to deal with the general view planning problem as described in Section 1.1.2.

Additionally, we want to mention the possibility to design a numerical

optimization scheme. Using an objective function according to (1.5) and an initial solution allow to perform iterative, non-linear numerical optimization. For this approach, the main effort is to find a suitable initial solution that lies within the basin of attraction of the global optimum. Thus, the whole task is just transferred to hold another title, but remains unsolved. Trivial or random initial solutions do not cope with the complexity of the problem. Hence, pure numerical optimization methods for view planning do not meet the claim stated by the complex and various problem of next-best-view planning. The next section will outline the view planning approach presented in this work.

1.3 Selected Approach and Contribution

The previous section has shown that it is hard to reference each detail of the view planning task within the depicted theoretical approaches. While we made these descriptions on high level, we expect even more modeling incapabilities when moving towards low level. For sampling and interpolation in Section 1.2.1, we demonstrated that important aspects as the sampling process itself are not regarded. As for state estimation in Section 1.2.2, we were also not successful in modeling all possible dependencies, and we suppose it merely feasible even to compute a good approximation of the respective likelihood function. The main reason for these difficulties is the inherent level of abstraction of the view planning task. View planning as such does not state what is wanted; it rather says that we know well-defined reconstruction goals, constraints, and prior knowledge, such that a properly designed method optimizes defined adjustable parameters in order to minimize a defined view planning objective function. This section gives a survey on how we construct a novel online optimization pattern to deal with the general view planning problem using a passive camera.

Instead of pursuing a top-down approach, which did not appear to be very promising in Section 1.2, we initially dedicate our considerations to the actual requirements of 3D reconstruction using a passive camera. Starting with intensity images, we need to solve the 2D correspondence problem, which we tackle by means of the well-known KLT feature tracking [Lucas and Kanade, 1981]. Given that we use a controlled environment providing additional prior knowledge, we state the requirement of each reconstruction step taking advantage of the additional prior knowledge. Hence we extend

the KLT tracking to use knowledge about extrinsic parameters in consideration of uncertainty [Trummer et al., 2008, Trummer et al., 2009b, Trummer et al., 2009a], and to implicitly perform robust 3D reconstruction [Trummer et al., 2009c]. The resulting tracking method we call *Guided* KLT tracking, or GKLT tracking. As the framework of feature tracking dictates a cycle consisting of camera movement, image acquisition, and feature tracking, we use this structure as a basis and develop a modular, online planning cycle. This online view planning method, as illustrated in Figure 1.4, performs cycles consisting of an active camera movement, image acquisition, GKLT feature tracking, and next-best-view planning. The view planning part of this cycle uses the collected data to adjust the extrinsic camera parameters optimally with respect to pre-defined reconstruction goals and constraints. By using GKLT tracking we are able to establish 3D covariance data during the 3D reconstruction. This data basis allows active accuracy optimization, which we perform by applying an extended E-criterion and developing a closed-form, optimal solution [Trummer et al., 2010a]. Further we facilitate completeness optimization, that may be found desirable since 3D reconstruction based on feature tracking only computes a finite set of 3D points. To fulfill needs of a continuous surface representation, we apply voxel-space techniques and consistency checking. By this means we compute a probabilistic, non-discrete 3D surface estimation that serves the respective planning goals. One particular aspect of view planning can be to detect and to improve special 3D object features requiring special views, such as notches on technical objects. To this end we introduce a novel approach to compute direct features of 3D surface triangulations for detection, classification and registration tasks [Trummer et al., 2009d]. Beholding its characteristics described later on, this novel 3D registration method can find broad application far beyond view planning. Finally, any view planning method shall be comparable to others in terms of how well the planning result meets pre-defined goals. While conditions constraining the reconstruction procedure are mostly trivially to compare, for instance, the number of views, the criteria for benchmarking 3D reconstructions state some severe questions. Attempting to answer these questions, we formulate a benchmarking scheme for 3D reconstructions that addresses accuracy and completeness issues [Munkelt et al., 2007]. Additionally we consider the 3D reconstruction method of factorization based on [Tomasi and Kanade, 1992]. We investigate the possibilities and effects due to actively adjusted parameters and draw implications for view planning.

In comprehension, we contribute to the field of 3D reconstruction and

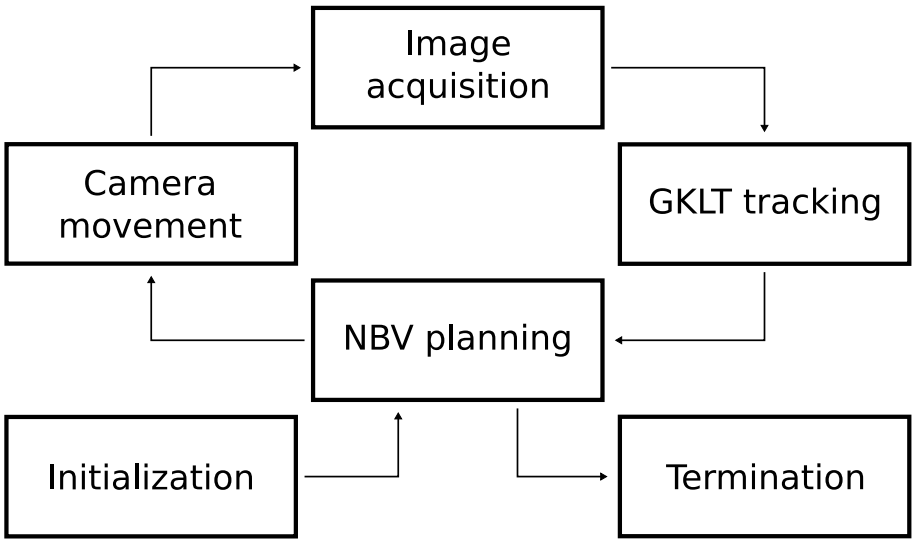


Figure 1.4: The online planning cycle proposed in this work.

next-best-view planning by the following achievements, mainly:

- a formalization of the general task of view planning,
- GKLT feature tracking using prior knowledge about camera parameters in combination with online, robust 3D reconstruction, cf. [Trummer et al., 2008, Trummer et al., 2009b, Trummer et al., 2009a, Trummer et al., 2009c],
- an online, combined optimization approach for next-best-view planning comprising tracking, reconstruction, and planning, cf. [Trummer et al., 2010a],
- an extended E-criterion and a closed-form solution for accuracy optimization, cf. [Trummer et al., 2010a],
- a voxel-based, probabilistic 3D surface estimation for addressing completeness issues,
- a way to compute invariant features of 3D surface triangulations and a corporate 3D registration method for general 3D surface triangulations, cf. [Trummer et al., 2009d],

- a benchmarking method regarding the special needs of view planning for 3D reconstruction, cf. [Munkelt et al., 2007],

- a methodology-based investigation of view planning using the factorization method for 3D reconstruction.

1.4 Structure of this Work

The thoughts in this chapter provide an introduction to next-best-view planning. Besides showing the existence of the view planning problem and motivating solutions, we give a concrete formalization that describes view planning as a constrained optimization problem. We underline the feasibility of the formal description by drafting very practical examples and applying our formalization. Since the formal description of the optimization problem does not define the optimization method, we examine standard approaches with respect to view planning, learn about view planning, and draw conclusions. Answering the drawbacks of standard optimization approaches, we state a novel online combined optimization approach for next-best-view planning and outline our further contributions to this field.

Chapter 2 provides a review of literature relevant to view planning and relates the known approaches to our contribution. A repetition of computer vision basics is given in Chapter 3. There, we sketch fundamental methods used in the later chapters.

The methodical starting point of our online view planning approach is the GKLT feature tracking, which we develop in Chapter 4. GKLT feature tracking extends the well-known KLT tracker to use known camera parameters regarding uncertainty. We emphasize that GKLT tracking is an integral component of our view planning approach as well as an independent feature tracking procedure.

For Chapter 5, we shortly deviate from the direct path to view planning. We describe a method to derive descriptors of 3D surface triangulations that are invariant to 3D Euclidean or similarity transformations, by choice. We compute these descriptors without sampling or projection by directly using the continuous surface itself. Further we propose an optimal 3D registration method based on these surface features. One application to view planning is, for instance, detecting special object features to be optimized. Beyond view planning, the methods depicted in this chapter find application in the

whole context of 3D reconstruction, particularly in dealing with recognition and registration tasks involving 3D surface triangulations.

Getting back to next-best-view planning, Chapter 6 specifies the modular, online planning framework proposed in this work. We describe the junction of GKLT tracking and online view planning, present an extended E-criterion for accuracy optimization, and we show probabilistic surface estimation allowing visibility checking. Further we propose a benchmark scheme to evaluate 3D reconstructions achieved by view planning. Additionally, we examine view planning with respect to the reconstruction method, for which we focus on the factorization method.

Finally, we provide an experimental evaluation of our findings in Chapter 7, which regard the view planning procedure as a whole. Further practical assessments of GKLT tracking and 3D registration are situated in the respective chapters. We conclude this work in Chapter 8.

Chapter 2

Literature Review of View Planning Methods

In the following we give a review of view planning methods known from literature. We already outlined in the previous chapter that view planning finds broad application for different tasks. However, the main interest of this work is view planning for 3D reconstruction, and therefore we concentrate on the respective view planning references. For the many possibilities to group known methods, we settle for reviewing model-based approaches in Section 2.1 and data-driven approaches in Section 2.2. While we hold up the clear focus on 3D reconstruction, we still have a look at view planning methods for other tasks in Section 2.3. Concluding this chapter, we provide a critical acclaim of the mentioned view planning methods in Section 2.4. Unless stated otherwise, all the approaches regard static scenes.

Standing above the sections of this chapter, the work of Scott, Roth, and Rivest [Scott et al., 2003] and the one of Chen, Li, Zhang, and Wang [Chen et al., 2008] provide surveys of view planning methods. While Scott et al. give an in-depth comparison of view planning in conjunction with range scanning, Chen et al. staple together view planning articles dealing with all kinds of tasks and sensors.

2.1 Model-based View Planning Approaches for 3D Reconstruction

Model-based view planning uses prior knowledge about the shape of an object – an object model – in order to improve the 3D reconstruction by active sensor control.

In [Tarbox and Gottschlich, 1995], Tarbox and Gottschlich present the concept of the binary measurability matrix representing which point of an object model is visible to a range scanner from which position of a discretized sensor space. Optimizing a viewpoint selection corresponds to the NP-complete minimum-set-cover problem, i. e. the goal is to find a minimal set of viewpoints that cover the whole object surface. Due to the mentioned runtime requirements, the authors compare different search strategies for complete surface reconstruction.

Chaumette, Boukir, Bouthemy, and Juvin [Chaumette et al., 1996] formulate a mainly model-based approach for applying a control law of visual servoing. The authors give implicit representations of geometric primitives as points, lines, spheres, and cylinders, which allow them to address the reduction of discretization and measurement errors in 3D reconstruction. As for a single point, the optimal controlled motion renders a camera path such that the intensity sensor moves at constant speed on a sphere around the point while keeping the point in the image center.

In [Madsen and Christensen, 1997], Madsen and Christensen investigate viewpoint planning with respect to polyhedral objects. They do not regard 3D reconstruction but the determination of the true angle between linked edges of the object using a passive camera. The work describes an iterative camera motion optimization that aims at a local extremum of the respective apparent angle in the image. This extremum is regarded as the case in which the true angle equals the apparent angle.

Whaite and Ferrie [Whaite and Ferrie, 1997] explore model-based view planning additionally applying a linear model of the measurement formation. The authors derive a maximum likelihood estimate of the surface model parameters and the according covariance matrix, which depends on the sensor parameters and the linear observation model. Minimizing the determinant of the mentioned covariance matrix performs the desired reduction of the uncertainty of the surface model parameters.

In [Marchand and Chaumette, 1999], Marchand and Chaumette extend

the approach of [Chaumette et al., 1996] by combining scene exploration and the reconstruction of 3D primitives, like cylinders and prisms, using a passive camera. The combined method performs an active estimation of detected primitives, followed by a global exploration in order to detect further primitives. The work also deals with the case of primitives occluding each other.

Banta, Wong, Dumont, and Abidi [Banta et al., 2000] exploit volumetric data representation to perform view planning by means of a range scanner. The approach uses model information given as a 3D CAD model or, if not available, created within an initial scanning stage. The actual view planner registers the acquired range data to the voxel space representing the scanner volume. Each voxel carries an attribute saying if it is occupied by the object surface or not. As a whole the voxels constitute the occupancy grid of the scanning volume. Additional information stores which parts of the scanning volume have been scanned and which have been occluded. The work proposes several heuristic criteria to both reach maximal information gain with the next view and perform complete surface reconstruction.

Reed and Allen [Reed and Allen, 2000] divide their view planning approach into two components, an incremental modeler building 3D models from range images and a planner optimizing the next sensor position. The modeler creates closed, bounded 3D surfaces containing surface tags marking imaged regions, occlusion boundary, and boundary of measurement space, all of which are updated after each scanning step by regularized set intersection. As the second step of the planning cycle, the method computes sets representing sensor imaging constraints, sensor placement constraints, occlusion constraints, and further stated constraints in terms of 3D subspaces. Finally, the planner uses set arithmetics to determine the resulting set holding valid positions for the next best view. Hence it is due to the definition of further constraints, if the method achieves a single next best view that is optimal in some sense.

Scott [Scott, 2009] formulates general requirements on view planning using laser scanning, in which he mainly aims at complete surface reconstruction. The work regards accuracy issues only in terms of positional errors of the scanning device. By establishing a measurability matrix connecting sensor positions and scannable areas, Scott tackles the specific multi-view planning problem applying optimization techniques to solve the set covering problem for the measurability matrix.

2.2 Data-driven View Planning Approaches for 3D Reconstruction

The focus of this work is on view planning for 3D reconstruction without using knowledge of the object structure. According data-driven approaches have also been studied in previous works.

One of the pioneer works of view planning is the one of Aloimonos, Weiss, and Bandyopadhyay [Aloimonos et al., 1988]. While there have been investigations on purposive, active sensor control before, for instance, in [Fraser, 1982], the work of Aloimonos et al. establishes a reasoning on the virtues of active vision. The work outlines the benefits of active vision using the examples of shape from shading, shape from contour, shape from texture, area-based optical flow, and structure from motion. Anyhow, the authors locate their active vision paradigm in the low- and intermediate-level computer vision, which, by the many research articles following, has shown to be an underestimation of the concept. In the fundamental work [Bajcsy, 1988], Bajcsy presents her understanding of active perception as an intelligent data acquisition, and she argues on a more abstract level than Aloimonos et al. She already states the necessity of modeling that regards scanner physics, signal processing, and data disturbance. As further important components she names the study of control strategies and the understanding of interactions between scanning and processing.

Shmuel and Werman [Shmuel and Werman, 1990] focus on accuracy optimization of depth estimation based on an a priori estimation of the correlation function. The basic idea of this work is to tackle the problem of image correspondence search by correlation analysis and to derive an uncertainty measure from the correlation function. To this end, an approximation of the correlation function by auto-correlation helps to find a camera translation that optimizes the a priori estimate of the image correlation.

Maver and Bajcsy [Maver and Bajcsy, 1993] examine view planning using a range scanner for complete scene reconstruction. Figuring out two kinds of occlusions for their scanning system, the authors approximate sensing shadows by polygons and compute viewing directions according to occluded parts of the scene. A histogram-based selection of the next best view shall then minimize the scene occlusions.

Massios and Fisher [Massios and Fisher, 1998] discuss view planning using a range scanner and a volumetric representation of the scene. The elements

of the measurement volume, the voxels, hold one of the attributes *seen*, *unseen*, *empty*, or *occlusion plane*. Additionally, the voxels are marked by a quality attribute that relates the scanning direction and the direction of the estimated surface normal preferring collinear scanning direction and surface normal. In conclusion this approach allows the weighted selection of the next best view with respect to completeness and accuracy.

Pito [Pito, 1999] describes an automated approach for complete surface reconstruction by means of a range scanner. Pito outlines that, especially for NBV algorithms, each step of the method has impact on the whole method. In particular, he exhibits the need to respect the physics of the scanner hardware. Further he designs the so called positional space algorithm for next-best-view planning. This algorithm uses a discretization of the space of hardware parameters to search for the next best view. Pito also defines desirable constraints for view planning using a range scanner.

In [Olague and Mohr, 2002], Olague and Mohr discuss view planning for accurate 3D reconstruction using a passive camera. The authors restrict the method to situations of a fixed distance between the camera and the objective points, which might raise the question how this distance can be known for points that are to be reconstructed. The approach in this work consists of two steps: First, the assumption of a Gaussian-distributed point position and a first-order Taylor expansion of the triangulation function yield a first-order approximation of the positional covariance matrix with respect to the triangulation process. Second, a genetic algorithm minimizes the maximum diagonal entry, which is the maximal variance of the respective approximated covariance matrix. In [Dunn et al., 2006], Dunn, Olague, and Lutton extend this kind of accuracy optimization for optimizing camera networks using an evolutionary algorithm.

Sablatnig, Tosovic, and Kampel [Sablatnig et al., 2003] place passive cameras, a laser scanner, and a light source around a turntable and examine the combination of these active and passive devices for view planning. The work employs the methods of shape from shading and active light patterns for 3D reconstruction. While the main focus of this work is an early data fusion of actively and passively yielded data for volumetric 3D reconstruction, the authors also describe a way to actively adjust the angle of the turntable in order to achieve a complete 3D representation of the object.

Chen and Li [Chen and Li, 2005] analyze the concept of the trend surface to predict the unseen part of a surface and to place a range scanner accordingly. For establishing the trend surface, the authors determine the

local smoothness of the measured surface boundary and use smooth regions to estimate the structure of the unseen surface. The next best view is then given in terms of sensor position and rotation such that a large part of the predicted surface is being scanned collinear to the estimated average surface normal.

Low and Lastra [Low and Lastra, 2006] concentrate on complete 3D reconstruction of indoor scenes using a range scanner. They authors develop a hierarchical method to evaluated feasible views. The fitness of a view is assessed by a view metric that takes into account reconstruction quality in terms of completeness and sampling density as well as acquisition constraints. These evaluation criteria are similar to the view planning constraints proposed by Pito [Pito, 1999].

Wenhardt, Deutsch, Hornegger, Niemann, and Denzler [Wenhardt et al., 2006] propose active accuracy optimization based on 3D reconstruction using a passive camera and an extended Kalman filter. This work models reconstruction uncertainty in terms of the entropy of a point estimate. Stating the assumption that the reconstruction error is caused by Gaussian noise, the optimization criterion minimizes an a priori estimate of the state entropy. In consequence of this modeling, the determinant of the state covariance matrix is being minimized. The optimization procedure performs a complete search over the set of possible camera positions. Once the optimal camera position out of presampled candidates is found, the planning method moves the camera on a direct path to that position while continuously taking images for feature tracking. The consecutive planning step is postponed until the camera has reached the previously planned position. In [Wenhardt et al., 2007], the authors compare the criteria of D-, E-, and T-optimality, which refer to the determinant, the eigenvalue, and the trace of the covariance matrix, respectively. The comparison does not yield a clear winner.

In [Loriot et al., 2008], Loriot, Seulin, and Gorria outline the growing importance of creating 3D models using a controlled environment and holding up well-defined quality criteria, i. e. the importance of next-best-view planning. The authors present a hierarchical approach for laser scanning cultural heritage. While the first step establishes a coarse 3D model of the object of interest, the second step detects wholes and computes optimal sensor positions to reach a complete 3D reconstruction. The work is extended for practical issues in [Rozenwald et al., 2010].

Munkelt, Trummer, Kühmstedt, Notni, and Denzler [Munkelt et al., 2009] propose a borderline approach between model-based and data-driven view

planning. The authors combine a fringe-projection measurement device together with a time-of-flight camera, which generates a rough volumetric 3D model during an initial scanning step. The following actual view planning does then carry out volumetric view planning using the fringe-projection device.

He, Long, and Li [He et al., 2010] present an engineering solution for completeness-oriented view planning using a laser scanning device. Their approach is based on evaluating candidate sensor positions along limit visible regions with respect to the gain of viewed volume. The authors themselves state their method suffering from shortcomings when dealing with concavities and self-occlusions.

2.3 Further Aspects of View Planning

View planning has also been studied for fields of application other than 3D reconstruction. Since these applications depart from the focus of this work, we just want to mention a few of them in order to illustrate the broad usage of view planning methods.

In [Zheng et al., 1991], Zheng, Chen, and Tsuji describe a scheme for active object manipulation using a controlled gripper as well as a controlled robot arm holding a passive camera. The active system switches between tasks of gripper movement and view planning in order to manipulate objects in a given manner. In doing so, the work uses motion vectors and image positions, but does not perform 3D reconstruction.

Hwang and Ahuja [Hwang and Ahuja, 1992] give an extensive survey of motion planning for robot navigation avoiding collisions. The authors assume polytopic representations of robots and obstacles and examine the complexities of planning algorithms for static and dynamic obstacles.

Roberts and Marshall [Roberts and Marshall, 1998] present model-based view planning for object recognition and inspection. In this work, the authors adapt the concept of the aspect graph and outline a search strategy to minimize the number of views that visit the whole object surface.

Shih and Gerhardt [Shih and Gerhardt, 2006] address complete model inspection by view planning. The view planning is based on several heuristic criteria like the total edge length or the number of visible edges. The goal of the planning procedure is to prepare a data basis for optimal sampling of the object surface.

Wang and Gupta [Wang and Gupta, 2007] deal with view planning in the special context of exploration and path planning for range sensors. The authors model the knowledge of the robot configuration space probabilistically, which allows to compute the next configuration in terms of maximally reducing the expected entropy. A finite, discrete set of candidate robot parameters forms the basic set for searching the optimal instance with respect to the entropy criterion.

The work of Chen and Davis [Chen and Davis, 2008] deals with networks of passive cameras for object tracking in dynamic scenes. A quality metric for camera configurations uses probabilistic modeling of the scene dynamics and of possible obstacles. Finally, the probability that at least two cameras can see a feature point describes the quality of a camera configuration.

2.4 Critical Acclaim

The previous work on view planning provides well-performing solutions to most differing understandings of the underlying problem. In addition, the known approaches employ a variety of assumptions and constraints. The overwhelming majority of previous view planning methods addresses range scanning devices, while we have argued in Section 1.1 that there are good reasons to investigate view planning for passive cameras and that the general task of view planning is not confined to a certain kind of sensing, cf. Section 1.1.3. Focusing on range scanners widely corresponds to a strict definition of view planning, which mainly aims at seeing the whole surface of an object, be it with or without prior knowledge of the object shape. While this definition is perfectly valid for this special task, it somewhat narrows our understanding of the general task of view planning that we formalized in Section 1.1.2. A proof that the idea of view planning is mightier than range-scanning a whole surface is given by view planning literature itself, which deals, for instance, with accuracy optimization, exploration, and mimimal-cost movements using different kinds of sensors and combinations of them. A further limiting aspect seen in the literature is a complete search of an optimal solution within a discretized search space. First, a discretization of the space of solutions most likely conceals the true, continuous optimum and, second, an exhaustive search raises efficiency issues, which oppose an online approach as presented in Chapter 6. Some researchers use heuristic criteria like contours and edges to find next best views. These works do not provide general conclusions, but

are only able to state results for the specific test objects used.

A serious issue, also stated in [Scott et al., 2003], is the quantitative evaluation of view planning methods, which we address in Section 6.1. Many authors provide mainly images of their reconstruction results. Despite the reasonable appearance, such a pictographic evaluation does not allow a quantitative comparison of different view planning methods. One difficulty in benchmarking view planning results is the computation of a measure of completeness. This difficulty becomes obvious trivially if we consider a gap in the 3D reconstruction as an appearance of non-completeness and the 3D reconstruction itself as a discrete, finite set of 3D points. With regard to the continuous 3D space, a discrete, finite set of 3D points shows unavoidable gaps everywhere. This may be the reason for many authors to stick with pictographic evaluation, and only some of them try to find useful quantitative measures expressing completeness.

Chapter 3

Relevant Basics of Computer Vision

This chapter presents a short repetition of computer vision basics that we consider as being relevant for this work. The benefit of such a repetition may seem questionable to the reader when taking into account splendid introductions to computer vision such as [Hartley and Zisserman, 2003], [Faugeras and Luong, 2001], [Trucco and Verri, 1998], and others. Nonetheless, we provide some of the fundamental concepts for the reasons of synchronizing terminology, introducing notation, and repeating some mathematical basics. Starting with Sections 3.1 and 3.2, we describe possibilities to model the image formation and to determine the parameters of the respective models. These prerequisites build the foundation to deal with 3D reconstruction in Section 3.3 and the epipolar geometry in Section 3.4. Finally, Section 3.5 illustrates the correspondence problem, which is fundamental in computer vision concerning 2D image space as well as 3D space.

3.1 Camera Models

When taking an image of the real word using an intensity camera, the fundamental aspect of the according information mapping is the dimension reduction from 3D to 2D, cf. Figure 3.1. Considering digital imaging devices, we additionally have to deal with discretization in the space domain, i. e. sampling, and intensity domain, i. e. quantization. The following camera models describe geometric transformations that map a 3D world point

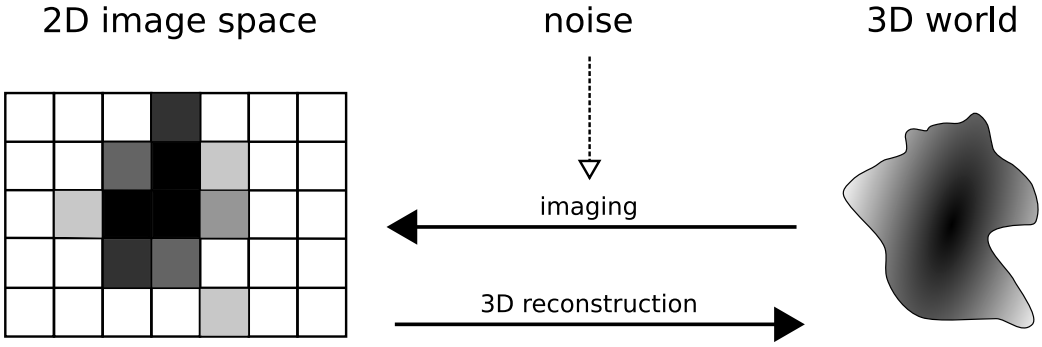


Figure 3.1: An abstract scheme of image formation and 3D reconstruction using a passive camera. The effects of dimension reduction, sampling, quantization, and noise within the imaging process state the inverse task of 3D reconstruction a difficult problem.

$\mathbf{X}_W = (X_W, Y_W, Z_W)^\mathsf{T}$ to an image point $\mathbf{x} = (x, y)^\mathsf{T}$.

Some of the following equations make use of n-dimensional *projective spaces* $\mathbb{P}^n$ and *homogeneous coordinates*, which we denote by $\tilde{}$. Likewise, we mark matrices used with homogeneous coordinates using the same symbol $\tilde{}$. A projective space $\mathbb{P}^n$ is a space $\mathbb{R}^{n+1} \setminus \{\mathbf{0}_{n+1}\}$. Without giving all the formal details, cf. [Faugeras and Luong, 2001] page 66, we note the correspondence of a point $\mathbf{x}$ in Cartesian coordinates and a point $\tilde{\mathbf{x}}$ in homogeneous coordinates as

$$\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T} \quad \longleftrightarrow \quad \tilde{\mathbf{x}} = \lambda(x_1, \ldots, x_n, 1)^\mathsf{T} \tag{3.1}$$

for $\lambda \in \mathbb{R}$, $\lambda \neq 0$. It follows that infinitely many elements $\tilde{\mathbf{x}} = \lambda(x_1, \ldots, x_n, 1)^\mathsf{T}$ of the projective space $\mathbb{P}^n$ represent the same element $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T} \in \mathbb{R}^n$. We also observe that we are not able to yield cartesian coordinates for points $\tilde{\mathbf{x}} = (x_1, \ldots, x_n, 0)^\mathsf{T}$. These points are called *ideal points* or *points at infinity*. We may also use homogeneous notation for describing lines. For example, a 2D line l is represented as a 3-vector $\tilde{\mathbf{l}}$, which we identify with the implicit equation

$$ax + by + c = 0 \tag{3.2}$$

for 2D points $\mathbf{x} = (x, y)^\mathsf{T}$. If we now use the homogeneous notation $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w})^\mathsf{T}$ for 2D points, we get

$$a\tilde{x} + b\tilde{y} + c\tilde{w} = 0 \tag{3.3}$$

and formally see the duality of points and lines in $\mathbb{P}_2$, which is part of the general duality of points and hyperplanes in $\mathbb{P}_n$.

Using homogeneous notation, we are able to give linear formulations for the following camera models. In general, we first consider a 3D-3D Euclidean transformation $\mathbf{T}$ to get the 3D homogeneous coordinates $\tilde{\mathbf{X}}_C$ with respect to the camera coordinate system using the 3D world point $\tilde{\mathbf{X}}_W$,

$$\tilde{\mathbf{X}}_C = \mathbf{T}\tilde{\mathbf{X}}_W = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix}, \tag{3.4}$$

where $\mathbf{T}$ is composed of a 3D rotation matrix $\mathbf{R}$ and a 3D translation vector $\mathbf{t}$,

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}. \tag{3.5}$$

Second, a 3D-2D mapping $\mathbf{M}$, which depends on the chosen camera model, transfers the 3D point $\tilde{\mathbf{X}}_C$ to an image point $\tilde{\mathbf{x}}$ in the image plane,

$$\tilde{\mathbf{x}} = \mathbf{M}\tilde{\mathbf{X}}_C = \mathbf{M}\mathbf{T}\tilde{\mathbf{X}}_W = \mathbf{P}\tilde{\mathbf{X}}_W \tag{3.6}$$

using a projection matrix $\mathbf{P} = \mathbf{M}\mathbf{T}$. For some camera models there exist dependencies between the parameters of $\mathbf{M}$ and $\mathbf{T}$, which makes a direct regard to $\mathbf{P}$ more useful.

For the sake of clarity, we want to comment on some important terms often used in the context of camera models. There are different ways to understand such a mapping of a 3D world point to a 2D image point. On the one hand, we use algebraic notation to describe the mapping. On the other hand, we often derive these algebraic models considering the geometric characteristics of the respective mapping. We even may understand it vice versa and see the geometry as a visual interpretation of the algebraic formulation. While the affine and projective camera models are strictly algebraic formulations, we describe special camera models that allow geometric interpretation of the image formation. An interesting fact is that throughout the literature all the simpler camera models are geometrically illustrated in terms of a special geometric interpretation of the perspective projection model, the pinhole camera, explained in Section 3.1.2.1. Despite the inconsequence that the respective entities of the pinhole camera are not defined in the simpler models, we stick to these interpretations for the sake of handy illustration, but note the respective borrowing of terms.

3.1.1 Affine Camera Models

The projection matrix of a general affine camera has the algebraic form

$$\mathbf{P}_{\text{aff}} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.7}$$

$P_{\text{aff}} \in \mathbb{R}^{3\times 4}$. An affine camera is also a projective camera, of course. The special restriction for affine cameras is that the last row of the respective projection matrix $\mathbf{P}_{\text{aff}}$ has the form $(0, 0, 0, x)$, which can be normalized to $(0, 0, 0, 1)$. Thus, an affine camera always maps ideal world points to ideal image points. Observing the fact that all intersection points of parallel world lines are located on the plane at infinity and that all these points are mapped to image points at infinity, we conclude that affine projections map parallel world lines to parallel lines in the image. Thus, affine projections preserve parallelism. A general affine projection as in (3.7) has eight parameters, or degrees of freedom. The special affine projections described below use less parameters and hence restrict the projection.

3.1.1.1 The Orthographic Camera Model

A very simple imaging model of mapping a 3D point $\tilde{\mathbf{X}}_C$ to the image plane is the orthographic projection, which essentially is a special parallel projection. The mapping $\mathbf{M}_{\text{o}}$ basically ignores the z-components of $\mathbf{T}$,

$$\mathbf{M}_{\text{o}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.8}$$

hence

$$\mathbf{P}_{\text{o}} = \mathbf{M}_{\text{o}}\mathbf{T} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.9}$$

Since an orthographic projection uses the same unit of measure as the world coordinate frame, applications are mainly restricted to cases of parallel projection where the absolute scaling is not of interest. The geometric interpretation of this camera model is that the 3D point $\tilde{\mathbf{X}}_C$ is orthographically mapped to the plane $Z = 0$ in the camera frame.

3.1.1.2 The Isotropically Scaled Orthographic Camera Model

The isotropically scaled orthographic projection performs a perpendicular parallel projection and an isotropic scaling by a factor s of the image coordinates. Hence

$$\mathbf{M}_{\text{iso}} = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.10}$$

and

$$\mathbf{P}_{\text{iso}} = \mathbf{M}_{\text{iso}}\mathbf{T} = \begin{pmatrix} sr_{11} & sr_{12} & sr_{13} & st_1 \\ sr_{21} & sr_{22} & sr_{23} & st_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.11}$$

While the parametric interpretation is simply an isotropic scaling of the orthographically projected image coordinates, we can also give a geometric interpretation that involves a special setup of a projective camera. A 3D point is first mapped orthographically to a support plane that is parallel to the image plane and leading through the gravity center of the 3D points. Second, the resulting point on the support plane is mapped to the image plane by central, or perspective, projection. This projection model is sometimes referred to as the *weak-perspective* projection model. The meaning of this imaging model is given by the facts that it can be seen as a zero-order approximation of the perspective projection, cf. [Horaud et al., 1994], and that the mapping is linear using Cartesian coordinates.

3.1.1.3 The Anisotropically Scaled Orthographic Camera Model

The anisotropically scaled orthographic model adds one more parameter, thus using s_x and s_y, to describe different scalings along the image axes. The projections matrices are

$$\mathbf{M}_{\text{aso}} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.12}$$

and

$$\mathbf{P}_{\text{aso}} = \mathbf{M}_{\text{aso}}\mathbf{T} = \begin{pmatrix} s_x r_{11} & s_x r_{12} & s_x r_{13} & s_x t_1 \\ s_y r_{21} & s_y r_{22} & s_y r_{23} & s_y t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.13}$$

3.1.1.4 The Paraperspective Camera Model

An affine camera model that is best described most clearly in geometric terms is the paraperspective camera model. Similar to the weak-perspective model, the paraperspective model uses a support plane at Z_0 in the camera frame, $Z = Z_0$. The model performs a parallel projection of 3D points to the support plane, and the direction is defined by the difference of a reference point $\mathbf{P}_0 = (X_0, Y_0, Z_0)^\mathsf{T}$ on the support plane and the camera center $\mathbf{C}$. Regarding the points projected to the support plane, it again follows a perspective projection to the image plane, see Figure 3.2.

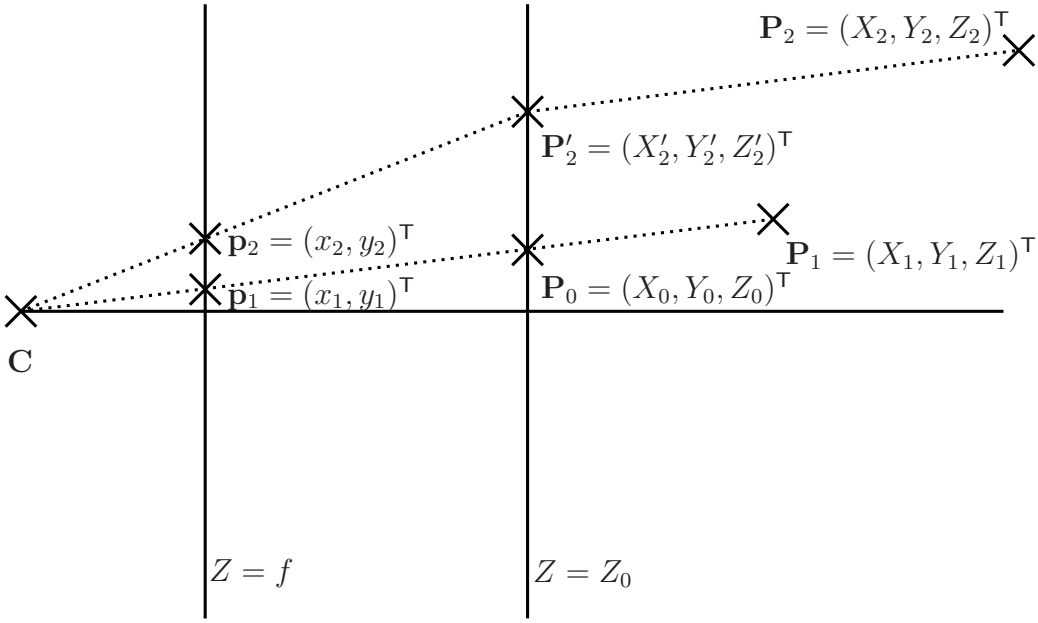


Figure 3.2: A geometric interpretation of the paraperspective camera model using terms of the perspective projection model, in particular the camera center $\mathbf{C}$ and the image plane $Z = f$. The first stage of the mapping is a parallel projection to the virtual support plane $Z = Z_0$. As the second stage, the model performs a perspective projection of points on the support plane to the image plane.

We now derive the algebraic formulation of the transformation performed within the paraperspective camera model. The point $\mathbf{P}_0$ on the virtual support plane $Z = Z_0$ defines the direction of the parallel projection such that

a point $P_1 = (X_1, Y_1, Z_1)^\mathsf{T}$ situated on the line passing through $\mathbf{C}$ and $\mathbf{P}_0$ is perspectively projected to the image plane. Hence, an arbitrary point $\mathbf{P}_2 = (X_2, Y_2, Z_2)^\mathsf{T}$ finds its projection $\mathbf{P}'_2 = (X'_2, Y'_2, Z'_2)^\mathsf{T}$ on the support plane parallel to the direction $\overline{\mathbf{P}_1\mathbf{P}_0}$. Since the mapping of $\mathbf{P}_1$ is a perspective projection, we yield $\overline{\mathbf{P}_1\mathbf{P}_0} = \overline{\mathbf{P}_0\mathbf{C}}$. For the projection of the arbitrary point $\mathbf{P}_2$ to $\mathbf{P}'_2$ follows

$$\frac{X_2 - X'_2}{Z_2 - Z_0} = \frac{X_0}{Z_0} \quad \text{and} \quad \frac{Y_2 - Y'_2}{Z_2 - Z_0} = \frac{Y_0}{Z_0} \tag{3.14}$$

and hence

$$X'_2 = \left(\frac{X_2}{Z_2 - Z_0} - \frac{X_0}{Z_0}\right)(Z_2 - Z_0) = X_2 - \frac{X_0 Z_2}{Z_0} + X_0, \tag{3.15}$$

$$Y'_2 = \left(\frac{Y_2}{Z_2 - Z_0} - \frac{Y_0}{Z_0}\right)(Z_2 - Z_0) = Y_2 - \frac{Y_0 Z_2}{Z_0} + Y_0, \tag{3.16}$$

while, of course, $Z'_2 = Z_0$. We yield the image point $\mathbf{p}_2 = (x_2, y_2)^\mathsf{T}$ as the perspective mapping of $\mathbf{P}'_2$ and as the paraperspective mapping of $\mathbf{P}_2$, respectively,

$$x_2 = \frac{fX'_2}{Z_0} = \frac{fX_2}{Z_0} - \frac{fX_0 Z_2}{Z_0^2} + \frac{fX_0}{Z_0}, \tag{3.17}$$

$$y_2 = \frac{fY'_2}{Z_0} = \frac{fY_2}{Z_0} - \frac{fY_0 Z_2}{Z_0^2} + \frac{fY_0}{Z_0}. \tag{3.18}$$

We express the resulting paraperspective projection matrix in terms of focal length f and rigid transformation $\mathbf{T}$ of the corresponding perspective camera and using the arbitrary reference point $\mathbf{P}_0 = (X_0, Y_0, Z_0)^\mathsf{T}$,

$$\mathbf{M}_{\text{para}} = \begin{pmatrix} \frac{f}{Z_0} & 0 & -\frac{fX_0}{Z_0^2} & \frac{fX_0}{Z_0} \\ 0 & \frac{f}{Z_0} & -\frac{fY_0}{Z_0^2} & \frac{fY_0}{Z_0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.19}$$

and

$$\mathbf{P}_{\text{para}} = \mathbf{M}_{\text{para}}\mathbf{T} = \begin{pmatrix} r_{11}\frac{f}{Z_0} - r_{31}\frac{fX_0}{Z_0^2} & r_{12}\frac{f}{Z_0} - r_{32}\frac{fX_0}{Z_0^2} & r_{13}\frac{f}{Z_0} - r_{33}\frac{fX_0}{Z_0^2} & t_1\frac{f}{Z_0} - t_3\frac{fX_0}{Z_0^2} + \frac{fX_0}{Z_0} \\ r_{21}\frac{f}{Z_0} - r_{31}\frac{fY_0}{Z_0^2} & r_{22}\frac{f}{Z_0} - r_{32}\frac{fY_0}{Z_0^2} & r_{23}\frac{f}{Z_0} - r_{33}\frac{fY_0}{Z_0^2} & t_2\frac{f}{Z_0} - t_3\frac{fY_0}{Z_0^2} + \frac{fY_0}{Z_0} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.20}$$

The paraperspective projection gains importance as being a closer approximation, namely first-order, to the perspective projection compared to the weak-perspective projection. This type of projection is capable to better model the effects of perspective projection while still being linear in the coordinates of the 3D point as shown in (3.17) and (3.18). Some methods, for example, in [Poelman and Kanade, 1997], make use of these linear properties. In the way formulated above, the paraperspective camera model is equivalent to the general affine camera model, cf. [Basri, 1996]. However, some methods, again instanced by [Poelman and Kanade, 1997], use the paraperspective model together with the assumption that the reference point $\mathbf{P}_0$ is equal to the gravity center of all points in the 3D scene. In this case, the paraperspective camera has, of course, less parameters than the general affine camera. A further way to look at this restriction is to consider it as a partial calibration of the paraperspective camera parameters. Setting the reference point $\mathbf{P}_0$ equal to the scene's gravity center yields parallel projections to the support plane that, in summation, provide the best first-order approximation of the perspective projection of the scene points. Fixing the reference point in the above manner is thus a reasonable setting of camera parameters and may be considered as a partial camera calibration.

3.1.2 Non-affine, Projective Camera Models

Algebraically, a general projective camera employs a projection matrix

$$\mathbf{P}_{\text{proj}} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & 1 \end{pmatrix} \tag{3.21}$$

with 11 degrees of freedom, $P_{\text{proj}} \in \mathbb{R}^{3\times 4}$. Since $p_{3i} \in \mathbb{R},\ 1 \leq i \leq 3$, ideal world points are mapped to image points that are not at infinity. Hence intersection points of parallel world lines are mapped to image points that are not at infinity. These image points are called the *vanishing points* of the respective lines. While a projective camera does not preserve parallelism, it does so with collinearity, which explains the term of *collineation* for a projective transformation.

3.1.2.1 The Pinhole Camera Model

The most important projective camera model describes the central, or perspective, projection that is performed by an ideal pinhole camera. Although modern cameras are not pinhole cameras and employ lenses, the pinhole model still is the most often used camera model for practical applications involving general digital cameras. Figure 3.3 illustrates the central projection of 3D points to an image plane.

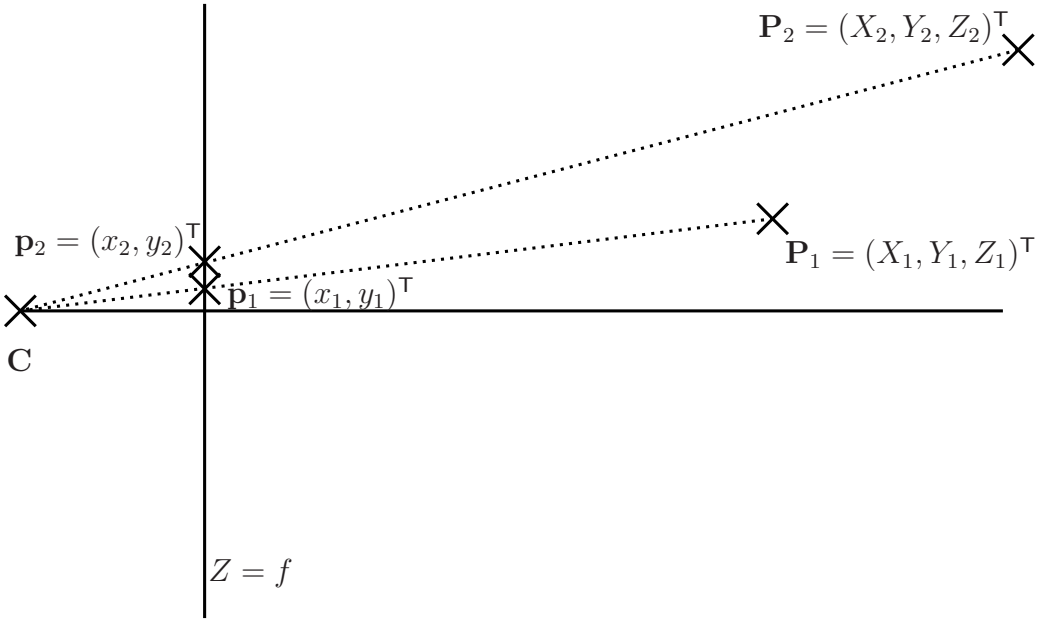


Figure 3.3: A geometric interpretation of the pinhole camera model using the camera center $\mathbf{C}$ and the image plane $Z = f$. The ray starting at $\mathbf{C}$ and passing through a point $\mathbf{P}_1$ is the ray of sight of $\mathbf{P}_1$. The intersection point of the ray of sight and the image plane defines the corresponding image point $\mathbf{p}_1$.

The mapping $\mathbf{M}_{\text{pin}} \in \mathbb{R}^{3\times4}$ is noted as matrix $\tilde{\mathbf{K}}$,

$$\tilde{\mathbf{K}} = \mathbf{M}_{\text{pin}} = \begin{pmatrix} \alpha_x & \gamma & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{3.22}$$

which is derived from the widely used denotation of the matrix $\mathbf{K}$,

$$\mathbf{K} = \begin{pmatrix} \alpha_x & \gamma & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.23}$$

Again, the whole projection matrix $\mathbf{P}_{\text{pin}}$ equals the catenation of the Euclidean 3D-3D transformation $\mathbf{T}$, which maps a world point $\tilde{\mathbf{X}}_W$ to a point $\tilde{\mathbf{X}}_C$ in the camera frame, and a 3D-2D projective transformation, in particular $\tilde{\mathbf{K}}$, mapping $\tilde{\mathbf{X}}_C$ to the image plane yielding the image point $\tilde{\mathbf{x}}$,

$$\mathbf{P}_{\text{pin}} = \tilde{\mathbf{K}}\mathbf{T} \tag{3.24}$$

and

$$\tilde{\mathbf{x}} = \mathbf{P}_{\text{pin}}\tilde{\mathbf{X}}_W = \tilde{\mathbf{K}}\mathbf{T}\tilde{\mathbf{X}}_W = \tilde{\mathbf{K}}\tilde{\mathbf{X}}_C. \tag{3.25}$$

Considering the matrix $\mathbf{K}$ in (3.23), we can find geometric interpretations of the separate terms. The parameters α_x and α_y allow anisotropic scaling of the image coordinates. If $\alpha_x = \alpha_y$, then we set $\alpha_x = \alpha_y = f$ and call f the camera's *focal length* given in pixel units. If, on the contrary, $\alpha_x \neq \alpha_y$, then a possible geometric interpretation is that the image *pixels*, which is the unit of measure in the image, are not squares. One particular point in the image plane is the *principal point* of the image, which is given by $(x_0, y_0)^\mathsf{T}$ in pixels and describes a coordinate offset in the image plane. The parameter γ facilitates a shearing of the image coordinates. Since most devices use sensors with a regular grid of square pixels, it is common choice and, again, a kind of partial calibration to set $\gamma = 0$. The whole bunch of these parameters of $\mathbf{K}$ does not depend on the camera position in the 3D world. That is why the five parameters of $\mathbf{K}$ are termed *internal* or *intrinsic camera parameters*. On the other hand, the elements of $\mathbf{T}$ define the position and rotation of the camera in the 3D world, which means six degrees of freedom. Consequently, they are called *external* or *extrinsic camera parameters*. The *camera center* $\mathbf{C}$, which is the center of projection, is a 3D world point expressing the camera position. With regard to the camera frame, $\mathbf{C}$ is the origin. Connecting the camera center and the principal point by a line defines the *optical axis* of the camera.

All the 11 parameters of the pinhole projection matrix $\mathbf{P}_{\text{pin}}$ have to be known in order to compute the image point of a 3D world point. The next section will deal with ways to calculate these camera parameters.

3.2 Camera Calibration

Camera calibration methods determine the parameters of a camera model. The calibration method mainly depends on the chosen camera model and the knowledge of the scene mapped by the device being calibrated. There is an ever-present research going on that examines the calibration of most various devices using special camera models and particular knowledge of the 3D scene, for instance, [Brown, 1971], [Armstrong et al., 1996], [Cipolla et al., 1999], [Agrawal and Davis, 2003], [Furukawa and Ponce, 2009], or using active camera control as in [Brückner and Denzler, 2010].

For this section, we restrict ourselves to discuss two ways of perspective camera calibration that use most different prior knowledge.

3.2.1 Direct Calibration

One possibility to determine the parameters of a camera uses complete knowledge of the i-th 3D world point, $\mathbf{X}_i = (X_i, Y_i, Z_i)^\mathsf{T}$, and the corresponding image point in camera j, $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j})^\mathsf{T}$. This direct calibration may use a rigid calibration rig with known 3D geometry that allows establishing the image correspondences automatically. In the following, we denote the perspective projection matrix used in (3.21) simply by $\mathbf{P}$. For camera j, we aim at the computation of the parameters of the respective projection matrix $\mathbf{P}_j$. The projection equation is

$$\tilde{\mathbf{x}}_{i,j} = \mathbf{P}_j \tilde{\mathbf{X}}_i = \begin{pmatrix} p_{11,j} & p_{12,j} & p_{13,j} & p_{14,j} \\ p_{21,j} & p_{22,j} & p_{23,j} & p_{24,j} \\ p_{31,j} & p_{32,j} & p_{33,j} & p_{34,j} \end{pmatrix} \tilde{\mathbf{X}}_i, \tag{3.26}$$

which defines the image coordinates as

$$x_{i,j} = \frac{p_{11,j}X_i + p_{12,j}Y_i + p_{13,j}Z_i + p_{14,j}}{p_{31,j}X_i + p_{32,j}Y_i + p_{33,j}Z_i + p_{34,j}}, \tag{3.27}$$

$$y_{i,j} = \frac{p_{21,j}X_i + p_{22,j}Y_i + p_{23,j}Z_i + p_{24,j}}{p_{31,j}X_i + p_{32,j}Y_i + p_{33,j}Z_i + p_{34,j}}. \tag{3.28}$$

We transform these equations to yield homogeneous equations that are linear in the unknown parameters of $\mathbf{P}_j$,

$$x_{i,j}\left(p_{31,j}X_i + p_{32,j}Y_i + p_{33,j}Z_i + p_{34,j}\right) - \left(p_{11,j}X_i + p_{12,j}Y_i + p_{13,j}Z_i + p_{14,j}\right) = 0, \tag{3.29}$$

$$y_{i,j}\left(p_{31,j}X_i + p_{32,j}Y_i + p_{33,j}Z_i + p_{34,j}\right) - \left(p_{21,j}X_i + p_{22,j}Y_i + p_{23,j}Z_i + p_{24,j}\right) = 0. \tag{3.30}$$

Using the according equations of all image points from camera j, we find the homogeneous linear equation system

$$\mathbf{A}\mathbf{k} = \mathbf{0} \tag{3.31}$$

that employs the vector $\mathbf{k}$ of unknown parameters,

$$\mathbf{k} = \left(p_{11,j}, p_{12,j}, \ldots, p_{33,j}, p_{34,j}\right)^{\mathsf{T}}, \tag{3.32}$$

and the coefficient matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{pmatrix} \vdots & & & & & & & & & & & \vdots \\ -X_i & -Y_i & -Z_i & -1 & 0 & 0 & 0 & 0 & x_{i,j}X_i & x_{i,j}Y_i & x_{i,j}Z_i & x_{i,j} \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & y_{i,j}X_i & y_{i,j}Y_i & y_{i,j}Z_i & y_{i,j} \\ \vdots & & & & & & & & & & & \vdots \end{pmatrix}. \tag{3.33}$$

Trivially, $\mathbf{k} = \mathbf{0}$ solves (3.31). In order to avoid this trivial solution, the usual way is to formulate the constraint $\|\mathbf{k}\|_2 = 1$. Hence the optimization task is to find an optimal $\mathbf{k}^*$ with

$$\mathbf{k}^* = \arg\min_{\mathbf{k}} \|\mathbf{A}\mathbf{k}\|_2 \quad \text{subject to} \quad \|\mathbf{k}\|_2 = 1. \tag{3.34}$$

The constraint in (3.34) is reasonable considering that the parameters of the projection matrix as depicted in (3.26) are defined only up to global scaling. A closed-form solution of (3.34) can be achieved using the singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$ with orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ and a diagonal matrix $\mathbf{D}$, for details see [Hartley and Zisserman, 2003] page 585. All vectors $\mathbf{k}$ meeting (3.31) build the *nullspace*, or *kernel*, of $\mathbf{A}$, which is always

a vector space since it contains the zero vector and linear combinations of its own elements. For non-degenerate cases, $\mathbf{A}$ from (3.33) satisfies the rank condition $\mathsf{rk}(\mathbf{A}) = 11$. Since $\mathbf{A} \in \mathbb{R}^{2i \times 12}$, the nullspace of $\mathbf{A}$ is a subspace of $\mathbb{R}^{12}$. Regarding $\mathsf{rk}(\mathbf{A}) = 11$, we find that the nullspace of $\mathbf{A}$ has dimension one. Consequently, the solution for (3.31) is defined only up to a scaling factor, which corresponds to the fact that the projection matrix (3.26) is likewise. Fixing the scaling factor of $\mathbf{k}$ by the constraint $\|\mathbf{k}\|_2 = 1$ is thus a way to yield the true projection matrix up to an unknown scaling factor. From theory we know that the singular value decomposition of $\mathbf{A}$ constitutes a diagonal matrix $\mathbf{D}$ containing the singular values of $\mathbf{A}$, exactly one of which is zero. The one being zero corresponds to the 1D nullspace of $\mathbf{A}$, and the corresponding column of $\mathbf{V}$ is a basis of this 1D nullspace and hence exactly the solution to (3.31). Due to noise, discretization and measurement errors, a practical solution is to select the smallest singular value and the corresponding column of $\mathbf{V}$.

For the direct calibration of one pinhole camera we need to fix 11 degrees of freedom. As shown in (3.29) and (3.30), using one known 3D point yields two linear equations in the unknown parameters of $\mathbf{P}_j$. Thus, we need at least six known 3D points and the corresponding image points in order to compute the calibration. As a strict restriction, the direct calibration method requires the 3D points and the corresponding image points to be known, which favors laboratory applications using a 3D calibration rig. Furthermore, the direct calibration does not tribute to the special structure of the projection matrix as noted in (3.24). This fact causes a solution achieved as depicted above to be unstable.

3.2.2 Self-Calibration

In the last section we used complete knowledge of the 3D scene and the corresponding image points to perform camera calibration. Since this knowledge is not available in general, we state the question what we can achieve if we use only the image points and the information about which image points are mappings of the same 3D point, i. e. known 2D point correspondences. A first thing to observe is that, without knowledge of the 3D scene, we cannot fix the absolute scale factor of the 3D reconstruction. It is always possible that the 3D world is smaller and closer to the camera, which yields the same images following the theorem of intersecting lines.

For self-calibration we use J cameras $\mathbf{P}_j$, $1 \leq j \leq J$, and the image points

$\mathbf{x}_{i,j}$ of I 3D points $\mathbf{X}_i$, $1 \leq i \leq I$, in the J camera images. In contrast to the direct calibration, the 3D points $\mathbf{X}_i$ are unknown and make (3.31) a difficult, non-linear problem. When is it theoretically possible to solve this problem? Each camera has 11 parameters and each 3D point counts three parameters. On the other hand, we gain two equations per image point. Since we need at least as much equations as parameters, we reach

$$11J + 3I \leq 2IJ. \tag{3.35}$$

If we find a way to compute the unknowns as a one-parameter family depending on the global scaling factor, the number of unknowns is reduced to $11J + 3I - 1$. Nonetheless, in general we compute the global scaling but just do not know if it is the true one. Hence (3.35) holds for the general case. Setting $J = 1$, i. e. using one camera transforms (3.35) to $I \leq -11$, which is not possible. This is an expression of the fact that it is not possible to do 3D reconstruction using a passive camera and only one image. Since the image mapping projects a 3D point to a 2D image point, for the inverse task the best we can do using one image point is to determine the 3D ray of sight on which the 3D point is located. For the reconstruction of the 3D point we have to incorporate more information, for instance, a corresponding image point originating from a second camera at another position.

For some applications of self-calibration we may find special conditions that change the number of unknowns. For instance, it is an often valid assumption that the intrinsic camera parameters are constant for all images, which is the case when all images are taken with the same camera without changing the internal setup. The number of unknowns is then $6J + 3I + 5$. Without loss of generality, we may identify the world coordinate frame with the one of the first camera, thus reducing the number of unknowns by another six degrees of freedom, $6J + 3I - 1 \leq 2IJ$. As another example, we investigate a setup of two general cameras, $J = 2$. According to (3.35), we get $I \geq 22$ and hence need to know at least 44 image points in the two image, which define 88 non-linear equations in the unknowns. Considering the effort to yield the 44 image points and the expectably instable solution of a system of 88 non-linear equations and 88 unknowns, this last example lets self-calibration show up as a rather theoretical approach to determine lower bounds of necessary equations. A partial objection to this statement comes along with the factorization method that we examine in Section 6.7.

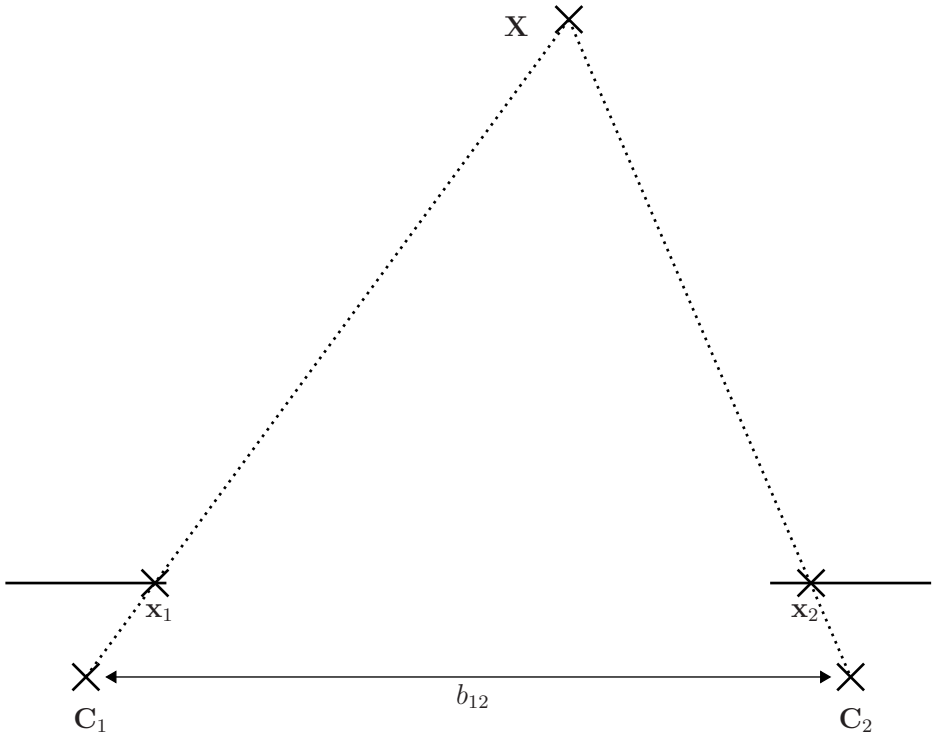


Figure 3.4: The principle of triangulation. Ideally, the task of 3D reconstruction using calibrated cameras consists of intersecting the rays of sight of at least two image point correspondences $\mathbf{x}_1$ and $\mathbf{x}_2$. In practice, one has to take into account various kinds of errors and noise. The cameras having centers at $\mathbf{C}_1$ and $\mathbf{C}_2$ define a baseline b_{12} that has influence on the robustness of the reconstruction of the 3D point $\mathbf{X}$.

3.3 3D Reconstruction

While the last section illustrated ways to compute the parameters of a camera model, we now assume these parameters to be known and discuss the 3D reconstruction of 3D points using image point correspondences. In this case, we know the positions and rotation of the cameras in the 3D world. By means of the image correspondences, we are able to compute the rays of sight belonging to a 3D point, see Figure 3.4.

Referring to camera j, we have complete knowledge of the camera pa-

rameters

$$\mathbf{P}_j = \tilde{\mathbf{K}}_j \mathbf{T}_j = \tilde{\mathbf{K}}_j \begin{pmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{pmatrix} \tag{3.36}$$

and introduce the notation

$$\mathbf{P}_j = (\mathbf{B}_j \mathbf{b}_j) \tag{3.37}$$

using

$$\mathbf{B}_j = \begin{pmatrix} p_{11,j} & p_{12,j} & p_{13,j} \\ p_{21,j} & p_{22,j} & p_{23,j} \\ p_{31,j} & p_{32,j} & p_{33,j} \end{pmatrix} \tag{3.38}$$

and $\mathbf{b}_j = (p_{14,j}, p_{24,j}, p_{34,j})^\mathsf{T}$. Exploiting $\mathbf{P}_j \tilde{\mathbf{C}}_j = \mathbf{0}$, which is given by the facts that $(0,0,0)^\mathsf{T}$ is an undefined image point and that the projection center is the only 3D point without a well-defined image point, we conclude

$$\tilde{\mathbf{K}}_j \begin{pmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{pmatrix} \tilde{\mathbf{C}}_j = 0, \tag{3.39}$$

$$\mathbf{R}_j \mathbf{C}_j + \mathbf{t}_j = 0 \tag{3.40}$$

$$-\mathbf{R}_j^\mathsf{T} \mathbf{t}_j = \mathbf{C}_j. \tag{3.41}$$

The ray of sight of image point $\mathbf{x}_j$ is located on a 3D line

$$\mathbf{x}_j(\lambda) = \mathbf{C}_j + \lambda \mathbf{s}_j \tag{3.42}$$

with a 3-vector $\mathbf{s}_j$. Applying the projection matrix in the form of (3.37) to this line yields

$$\tilde{\mathbf{x}}_j = \mathbf{B}_j (\mathbf{C}_j + \lambda \mathbf{s}_j) + \mathbf{b}_j = (\mathbf{B}_j \mathbf{C}_j + \mathbf{b}_j) + \lambda \mathbf{B}_j \mathbf{s}_j \tag{3.43}$$

and hence

$$\mathbf{s}_j = \lambda^{-1} \mathbf{B}_j^{-1} \tilde{\mathbf{x}}_j. \tag{3.44}$$

Equation (3.43) is justified by considering that the pinhole camera $\mathbf{P}_j$ maps each point on the line $\mathbf{x}_j(\lambda)$, except for the camera center $\mathbf{C}_j$, to the one image point $\tilde{\mathbf{x}}_j$. For two cameras, $j \in \{1, 2\}$, as illustrated in Figure 3.4, we are able to compute the lines of sight belonging to the 3D point $\mathbf{X}$ by means of the projection matrices $\mathbf{P}_j$ and the image points $\mathbf{x}_j$ of $\mathbf{X}$. Now, a logical way to find $\mathbf{X}$ would be to compute the intersection point between lines $\mathbf{x}_1(\lambda)$ and $\mathbf{x}_2(\lambda)$. The flaw of this approach is that the 3D lines of sight will not intersect in any practical application. Even if we assume noise and

measurement errors to be zero, we still have to cope with discretization effects as shown in Figure 3.1. As a second approach we could employ geometric considerations to find a 3D point that has a minimal sum of distances to both lines, which we then term an estimate $\widehat{\mathbf{X}}$ of the true 3D point $\mathbf{X}$. This second approach seems reasonable for two cameras, but with more cameras the geometry is getting more cumbersome.

A commonly used way to tackle this *triangulation* problem is to carry out an algebraic optimization known as the direct-linear-transform (DLT) algorithm. The proceeding is quite the same as for the direct calibration in Section 3.2.1 and ends up in solving a homogeneous linear equation system. In fact, we use the same equations (3.29) and (3.30) and only slightly adapt them to the homogeneous notation $\tilde{\mathbf{X}} = (\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{W})^\mathsf{T}$ of the 3D point $\mathbf{X} = (X, Y, Z)^\mathsf{T}$,

$$\begin{aligned} x_j \left(p_{31,j}\tilde{X} + p_{32,j}\tilde{Y} + p_{33,j}\tilde{Z} + p_{34,j}\tilde{W}\right) \\ - \left(p_{11,j}\tilde{X} + p_{12,j}\tilde{Y} + p_{13,j}\tilde{Z} + p_{14,j}\tilde{W}\right) = 0, \end{aligned} \tag{3.45}$$

$$\begin{aligned} y_j \left(p_{31,j}\tilde{X} + p_{32,j}\tilde{Y} + p_{33,j}\tilde{Z} + p_{34,j}\tilde{W}\right) \\ - \left(p_{21,j}\tilde{X} + p_{22,j}\tilde{Y} + p_{23,j}\tilde{Z} + p_{24,j}\tilde{W}\right) = 0. \end{aligned} \tag{3.46}$$

Again, we establish a homogeneous linear equation system $\mathbf{Ak} = \mathbf{0}$, but here the unknowns are the homogeneous coordinates of the 3D point, $\mathbf{k} = \tilde{\mathbf{X}}$. Applying the singular value decomposition to the accordingly constructed coefficient matrix, $\mathbf{A} = \mathbf{UDV}^\mathsf{T}$, gives the solution as the column of $\mathbf{V}$ corresponding to the smallest singular value. We again outline that the solution $\tilde{\mathbf{X}}^*$ is algebraically optimal in the sense that

$$\tilde{\mathbf{X}}^* = \arg\min_{\tilde{\mathbf{X}}} \|\mathbf{A}\tilde{\mathbf{X}}\|_2 \quad \text{subject to} \quad \|\tilde{\mathbf{X}}\|_2 = 1, \tag{3.47}$$

which has no geometric meaning. In order to reach a geometrically optimal solution, it is possible to perform a *bundle adjustment* that changes the 3D coordinates and the projection parameters, by choice, and minimizes the sum of distances between the true image points and the projected image points

using the parameters being adjusted – the so called *back-projection error*

$$\begin{aligned}\epsilon_{bp} = &\sum_j \left(x_j - \frac{p_{11,j}X + p_{12,j}Y + p_{13,j}Z + p_{14,j}}{p_{31,j}X + p_{32,j}Y + p_{33,j}Z + p_{34,j}} \right)^2 \\ &+ \sum_j \left(y_j - \frac{p_{21,j}X + p_{22,j}Y + p_{23,j}Z + p_{24,j}}{p_{31,j}X + p_{32,j}Y + p_{33,j}Z + p_{34,j}} \right)^2 .\end{aligned} \tag{3.48}$$

Bundle adjustment inherently is an iterative optimization setup and minimizes a defined geometric error, but needs an initial solution. This starting point may be given as the closed-form, algebraically optimal solution presented above.

Looking at (3.34) and (3.47), we observe a close formal relation between the tasks of camera calibration and 3D reconstruction. The statements about self-calibration in Section 3.2.2 underline this relation, since we are free to choose fixed and varying parameters of cameras and points. Likewise, bundle adjustment can perform camera calibration, 3D reconstruction, or either. A special 3D reconstruction method that also outlines the dependencies between camera calibration and 3D reconstruction is the *stratified reconstruction*, cf. [Hartley and Zisserman, 2003] page 267. The stratified approach first establishes a projective 3D reconstruction using not more than image correspondences. Projective reconstruction means that the true Euclidean position of point $\tilde{\mathbf{X}}_E$ and projective reconstruction $\tilde{\mathbf{X}}_P$ are related by a matrix $\mathbf{H} \in \mathbb{R}^{4\times4}$. The whole 3D reconstruction is thus defined up to 15 degrees of freedom, since one parameter of $\mathbf{H}$ can be normalized. Without any further information about camera parameters or scene constraints, this is all we can do. Including further knowledge we can update the reconstruction to an affine and, finally, to a Euclidean, or metric, 3D reconstruction. Roughly sketching the procedure, we start with the observation that an affine reconstruction always maps ideal points to ideal points. So if we are able to find a matrix $\mathbf{H}$ that preserves the *plane at infinity* $\pi_\infty = \{(X, Y, Z, 0)^\mathsf{T}, X, Y, Z \in \mathbb{R}\}$, then $\mathbf{H} = \mathbf{H}_{\text{aff}}$ and $\mathbf{H}_{\text{aff}}\tilde{\mathbf{X}}_P$ is an affine reconstruction. The final step towards metric reconstruction requires knowledge of the internal camera parameters, which is equivalent to knowing the image of the absolute conic and can also be derived from scene knowledge. As already mentioned above, the last degree of freedom is a global scaling factor of the whole scene including the camera positions. To resolve for this scaling factor, we need to know one true world distance.

As a concluding remark to this brief glance at 3D reconstruction we again state that the effects of dimension reduction, sampling, quantization, and noise strongly challenge according methods. Since 3D reconstruction and camera calibration are mutually dependent and methods allow to fix parameters, we are free to apply prior knowledge in order to improve the results. Respective reconstruction methods exploit special properties of prior knowledge, for instance, the knowledge of the geometry of a plane scene quadrangle as in [Trummer et al., 2005] and [Trummer et al., 2006], or the knowledge of general world planes as in [Kähler, 2009].

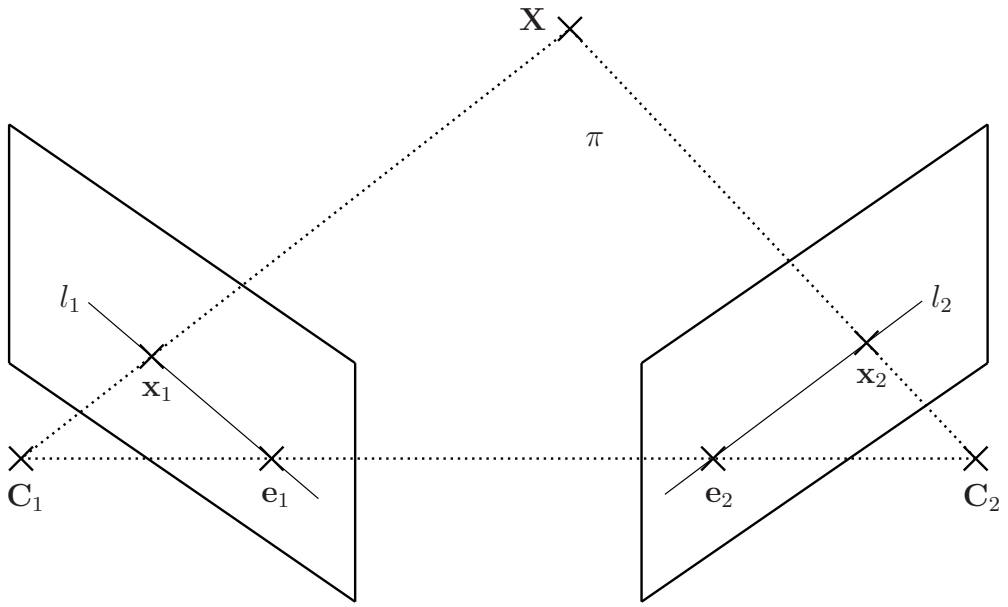


Figure 3.5: A setup of two cameras. The properties of the epipolar geometry are outlined in Section 3.4.

3.4 The Epipolar Geometry

Using passive cameras without scene knowledge, the smallest unit performing 3D reconstruction is a setup of two cameras. For such a stereo setup seeing one 3D scene point we can derive geometric properties that are illustrated in Figure 3.5.

Mapping the 3D point $\mathbf{X}$ by two cameras $\mathbf{P}_1$ and $\mathbf{P}_2$, we yield two corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$. Further we consider the 3D line defined by the two camera centers $\mathbf{C}_1$ and $\mathbf{C}_2$ and the intersection points with the respective image planes, which results in the *epipoles* $\mathbf{e}_1$ and $\mathbf{e}_2$. The *epipolar lines* l_1 and l_2 are 2D lines in the image connecting the epipole and an image point. We observe that the 3D point $\mathbf{X}$, the corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$, the respective camera centers $\mathbf{C}_1$ and $\mathbf{C}_2$, the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$, and the epipolar lines l_1 and l_2 are all located in the same 3D plane π.

In the following, we use these properties to formulate relations between the entities in different images. We start by the epipolar line l_1, represented by a 3-vector $\tilde{\mathbf{l}}_1$, in terms of the epipole and the image point in the particular image,

$$\tilde{\mathbf{l}}_1 = \tilde{\mathbf{e}}_1 \times \tilde{\mathbf{x}}_1. \tag{3.49}$$

Applying the notation (3.37) for the projection matrices and the result (3.41), we find

$$\tilde{\mathbf{e}}_1 = \mathbf{P}_1\tilde{\mathbf{C}}_2 = \mathbf{B}_1\mathbf{C}_2 + \mathbf{b}_1 = \mathbf{B}_1\left(-\mathbf{B}_2^{-1}\mathbf{b}_2\right) + \mathbf{b}_1. \tag{3.50}$$

On the other hand, both image points are mappings of the 3D point $\mathbf{X}$. Therefore we get

$$\tilde{\mathbf{x}}_j = \mathbf{P}_j\tilde{\mathbf{X}}, \tag{3.51}$$

$$\tilde{\mathbf{x}}_j = \mathbf{B}_j\mathbf{X} + \mathbf{b}_j, \tag{3.52}$$

$$\mathbf{X} = \mathbf{B}_j^{-1}\left(\tilde{\mathbf{x}}_j - \mathbf{b}_j\right). \tag{3.53}$$

By interpreting $\tilde{\mathbf{x}}_j$ in (3.53) as a term that is known only up to scale, which gives us an expression of the respective line of sight, we now use that the lines of sight intersect in $\mathbf{X}$,

$$\mathbf{B}_1^{-1}\left(\tilde{\mathbf{x}}_1 - \mathbf{b}_1\right) = \mathbf{B}_2^{-1}\left(\tilde{\mathbf{x}}_2 - \mathbf{b}_2\right), \tag{3.54}$$

$$\mathbf{B}_1^{-1}\tilde{\mathbf{x}}_1 - \mathbf{B}_1^{-1}\mathbf{b}_1 = \mathbf{B}_2^{-1}\tilde{\mathbf{x}}_2 - \mathbf{B}_2^{-1}\mathbf{b}_2, \tag{3.55}$$

$$\tilde{\mathbf{x}}_1 = \mathbf{B}_1\mathbf{B}_2^{-1}\tilde{\mathbf{x}}_2 - \mathbf{B}_1\mathbf{B}_2^{-1}\mathbf{b}_2 + \mathbf{b}_1. \tag{3.56}$$

If we consider (3.49), we reach a quite bulky representation,

$$\tilde{\mathbf{l}}_1 = \left(\mathbf{B}_1\left(-\mathbf{B}_2^{-1}\mathbf{b}_2\right) + \mathbf{b}_1\right) \times \left(\mathbf{B}_1\mathbf{B}_2^{-1}\tilde{\mathbf{x}}_2 - \mathbf{B}_1\mathbf{B}_2^{-1}\mathbf{b}_2 + \mathbf{b}_1\right), \tag{3.57}$$

that relates the epipolar line in the first image to the camera matrices and the image point in the second image. Without loss of generality, we now identify

the world coordinate frame with the one of the second camera, which makes

$$\mathbf{P}_2 = \tilde{\mathbf{K}}_2 \begin{pmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{pmatrix} = \tilde{\mathbf{K}}_2 \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} = \tilde{\mathbf{K}}_2 \tag{3.58}$$

using the 3×3 identity matrix $\mathbf{I}_3$. It follows that

$$\mathbf{B}_2 = \mathbf{K}_2, \tag{3.59}$$

$$\mathbf{b}_2 = \mathbf{0} \tag{3.60}$$

and

$$\tilde{\mathbf{e}}_1 = \mathbf{b}_1, \tag{3.61}$$

$$\tilde{\mathbf{x}}_1 = \mathbf{B}_1 \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2 + \mathbf{b}_1. \tag{3.62}$$

In preparation of the next step, we introduce the operator $[\cdot]_\times$ that maps a 3-vector to a 3×3 matrix in the following manner. If $\mathbf{v} = (v_1, v_2, v_3)^\mathsf{T}$, then

$$[\mathbf{v}]_\times = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}. \tag{3.63}$$

The benefit of this notation is that we are able to express the cross product of vectors in terms of a multiplication of matrix and vector, so for another 3-vector $\mathbf{z}$ we may wright

$$\mathbf{v} \times \mathbf{z} = [\mathbf{v}]_\times \mathbf{z}. \tag{3.64}$$

It is known [Gantmacher, 1998] that a skew-symmetric matrix such as $[\mathbf{v}]_\times$ has an even-numbered rank. For the general case we also know $\mathsf{rk}([\mathbf{v}]_\times) \neq 0$, hence we have $\mathsf{rk}([\mathbf{v}]_\times) = 2$ for non-degenerate cases. Using the above notation, we apply (3.61) and (3.62) to (3.49),

$$\tilde{\mathbf{l}}_1 = \mathbf{b}_1 \times \left(\mathbf{B}_1 \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2 + \mathbf{b}_1\right) = [\mathbf{b}_1]_\times \left(\mathbf{B}_1 \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2 + \mathbf{b}_1\right) = [\mathbf{b}_1]_\times \mathbf{B}_1 \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2, \tag{3.65}$$

since $[\mathbf{b}_1]_\times \mathbf{b}_1 = \mathbf{0}$. We set $[\mathbf{b}_1]_\times \mathbf{B}_1 \mathbf{K}_2^{-1} = \mathbf{F}$ and call $\mathbf{F}$ the *fundamental matrix* corresponding to the cameras $\mathbf{P}_1$ and $\mathbf{P}_2$. Since $\mathsf{rk}([\mathbf{b}_1]_\times) = 2$, we also get $\mathsf{rk}([\mathbf{F}]_\times) = 2$ for non-degenerate cases. Repeating (3.65) using $\mathbf{F}$,

$$\tilde{\mathbf{l}}_1 = \mathbf{F} \tilde{\mathbf{x}}_2, \tag{3.66}$$

we reach the important result that the image point in the first image corresponding to $\mathbf{x}_2$ is located on the line defined by $\mathbf{F}\tilde{\mathbf{x}}_2$, which is called the

epipolar constraint. This fact implies that the fundamental matrix encodes the relative pose of the two cameras, which can also be seen from (3.65). Another important characteristic of $\mathbf{F}$ is shown by considering the relation (3.3) of points and lines in homogeneous notation. If point $\mathbf{x}_1$ is an element of the line l_1, then $\tilde{\mathbf{x}}_1^\mathsf{T}\tilde{\mathbf{l}}_1 = 0$. In connection with (3.66) we yield

$$\tilde{\mathbf{x}}_1^\mathsf{T}\tilde{\mathbf{l}}_1 = \tilde{\mathbf{x}}_1^\mathsf{T}\mathbf{F}\tilde{\mathbf{x}}_2 = 0. \tag{3.67}$$

The importance of this result is that the fundamental matrix is characterized only in terms of corresponding image points. The number of image correspondences we need to determine $\mathbf{F}$ depends on the number of degrees of freedom of the fundamental matrix. Of the nine elements of $\mathbf{F}$, we may normalize one element since we use homogeneous image coordinates. Further we know that $\mathbf{F}$ is a singular matrix. Finally we find that the number of parameters of $\mathbf{F}$ is less than or equal to seven.

In (3.67) we see that we get one linear equation in the unknown elements of $\mathbf{F}$ from one image correspondence. Thus we need at least seven image correspondences to compute $\mathbf{F}$. One classical way to calculate the fundamental matrix is the eight-point algorithm, originally presented in [Longuet-Higgins, 1981], which does not employ the rank-constraint on $\mathbf{F}$.

If one aims at the relative pose of two cameras, it is not necessary to regard the intrinsic calibrations of the cameras. Since, on the contrary, $\mathbf{F}$ does so, the computation of $\mathbf{F}$ is not the actual goal and also requires more data than just solving for the relative pose. One way out is to assume known intrinsic parameters and to use *normalized image coordinates*

$$\tilde{\mathbf{x}}' = \mathbf{K}^{-1}\tilde{\mathbf{x}}. \tag{3.68}$$

Thus we can see normalized image points as mappings using a camera with a special internal calibration, which is expressed by $\mathbf{K}' = \mathbf{I}_3$. Yet, this has no influence on the correspondence of image points, and with respect to (3.67) we reach

$$\left(\mathbf{K}_1\tilde{\mathbf{x}}_1'\right)^\mathsf{T}\mathbf{F}\mathbf{K}_2\tilde{\mathbf{x}}_2' = \tilde{\mathbf{x}}_1'^\mathsf{T}\mathbf{K}_1^\mathsf{T}\mathbf{F}\mathbf{K}_2\tilde{\mathbf{x}}_2' = 0. \tag{3.69}$$

The inner matrices are combined to yield the *essential matrix*

$$\mathbf{E} = \mathbf{K}_1^\mathsf{T}\mathbf{F}\mathbf{K}_2. \tag{3.70}$$

In contrast to the fundamental matrix, the essential matrix only depends on the relative pose and has five degrees of freedom, cf. [Hartley and Zisserman,

2003] page 257. The essential matrix is, of course, a singular matrix and of its singular values two are identical and the third one is zero.

The homogeneous linear equations in (3.67) and (3.69) provide a possibility to compute the fundamental and the essential matrix, respectively. Following the procedure described in Section 3.2.1, we may establish a homogeneous linear equation system and solve it via singular value decomposition. In this way, the method completely ignores the constraints applying to the rank and singular values of the wanted matrices. Consequently, according solutions are sensitive to noise and other types of errors. On the other hand, there are methods that directly address the mentioned constrained, but end up in complex non-linear equations that are hard to tackle, for instance, [Stewenius et al., 2006]. A comparison of according methods is given in [Brückner et al., 2008].

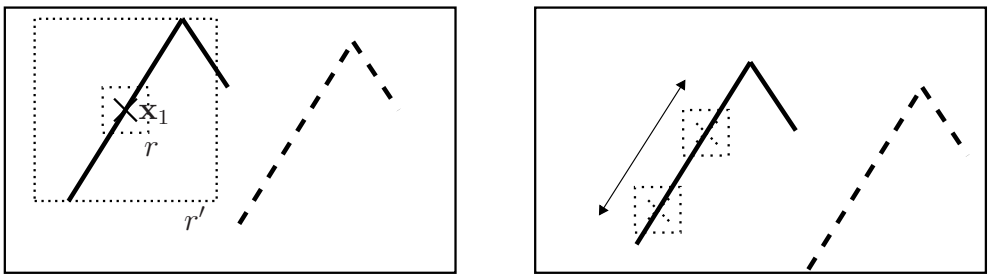


image1 **image2**

Figure 3.6: The aperture problem for 2D image correspondences. Given the image point $\mathbf{x}_1$ in the first image, we aim at the corresponding point in the second image. We observe the image structure around $\mathbf{x}_1$ inside a local region r, which is an edge. Due to the definition of the local region r, the image data does not provide enough information to decide the correct position of the corresponding point in the second image. We are left with the positional uncertainty along the image edge. The further description is given in Section 3.5.

3.5 The Correspondence Problem for Data Registration

When dealing with the epipolar geometry as in the last section, we are obliged to talk about *point correspondences*. While we assumed the correspondences to be known, this is not the case for a general application. In fact, if we want to compute the epipolar geometry, it is an essential and difficult prerequisite to determine correspondences between image points. That means we have to find image points that are mappings of the same 3D world point. Since we only have the image data to our disposal, the task is to somehow relate image positions based on the image data, i. e. to *register* image points. Likewise, the correspondence problem exists in 3D and is as fundamental as for images. For example, a stereo pair of cameras enables us to recover a partial 3D scene using one stereo shot. Moving the stereo camera pair to another position yields a further part of the 3D scene. A reasonable wish would now be to fuse the two partial 3D reconstruction in order to create one 3D reconstruction. If we have no information about the movement of the stereo camera, we are constrained to the image data and the reconstructed 3D data. Concerning the 3D data, the task is to find 3D points in the partial reconstructions that represent the same 3D world point, i. e. to register the according 3D points. These examples already highlight that the correspondence problem is as fundamental as difficult in computer vision.

Throughout the instances of the correspondence problem, one is confronted with the possibility of ambiguous data as illustrated in Figure 3.6. Historically, this issue was first related to the case of seeing the world through an aperture, so it was called the *aperture problem*. This visualization is still true, we just term the reason more abstractly as local data. In Figure 3.6 we only use the local data in region r around the point $\mathbf{x}_1$. We then observe that we are not able to decide for the true position of the corresponding point in the second image. Thinking about the region r, we further grasp that there is a larger region r' that includes the image corner, which suddenly endows us with the ability to uniquely solve this special 2D correspondence problem and find the corresponding point in the second image. What happened by using the larger local context r'? The information in region r says that $\mathbf{x}_1$ is located on an image edge, which is a 1D information. Since we have 2D image data, our solution remains with one degree of freedom, which is the position along the edge. The larger region r' around $\mathbf{x}_1$ includes a part of

the non-parallel second edge or even the corner between the two edges. By this means we yield the information that $\mathbf{x}_1$ is located in this special 2D structure. These 2D data fix all two degrees of freedom of the corresponding image position for the example in Figure 3.6. Nonetheless, we can imagine cases in which both images contain two of the depicted edge pairs, see the dashed lines in Figure 3.6. Again, we would not be able to find the true solution, if the local region would cover only the one edge pair around $\mathbf{x}_1$. We conclude that the aperture problem in correspondence search is theoretically ever-present for any non-global use of data.

In the following, we depict ways to tackle the correspondence problem. In Section 3.5.1 we describe an iterative approach to estimate the parameters of a transformation model using observed data. This algorithm also applies to 2D and 3D data and provides correspondences as a side effect. Sections 3.5.2 and 3.5.3 outline the specific problems for image data and 3D data, respectively.

3.5.1 RANSAC Approaches

The RANdom SAmple Consensus (RANSAC) is a randomized, iterative method for robust parameter estimation and was originally presented in [Fischler and Bolles, 1981].

Let us suppose a model $\mathcal{M}$ depending on a set of P parameters denoted as a vector $\mathbf{p} = (p_1, \ldots, p_P)^\mathsf{T}$, $\mathcal{M} = \mathcal{M}(\mathbf{p})$. Further assume an operator Γ that generates all possible elements out of a universe U supported by the model $\mathcal{M}(\mathbf{p})$,

$$\Gamma(\mathcal{M}(\mathbf{p})) \subseteq \mathsf{U}. \tag{3.71}$$

Finally, let $\mathsf{D} \subseteq \mathsf{U}$ be a set of observed data elements, $\mathsf{D} = \{d_1, d_2, \ldots\}$. We now want to find a model and adjust its parameters such that we are able to reproduce the observed data, i. e. we aim at a model $\mathcal{M}(\mathbf{p})$ and fixed parameters $\mathbf{p}^*$ with

$$\mathsf{D} \subseteq \Gamma(\mathcal{M}(\mathbf{p}^*)). \tag{3.72}$$

Our hopeful, or optimistic, assumption is that $\mathcal{M}(\mathbf{p}^*)$ describes the reality that we observed only by means of the elements of D. As this paradigm of modeling the reality applies to all natural sciences, so it applies to the RANSAC method.

The RANSAC procedure is outlined by a few simple actions. The model $\mathcal{M}$ as well as the basic set of the parameter vector $\mathbf{p}$ are assumed to be

known. Further we are given a set D of observed data. RANSAC performs the following steps:

1. Randomly select a minimal subset $\mathsf{S}_1 \subset \mathsf{D}$, $\#\mathsf{S}_1 = k$, that uniquely fixes the parameters $\mathbf{p}$ of model $\mathcal{M}$. Increment a counter c of random selections.

2. Use the fixed parameters $\mathbf{p}_1$ to determine a set S_1^* of elements, $\mathsf{S}_1 \subseteq \mathsf{S}_1^* \subseteq \mathsf{D}$, that are within an error tolerance t of $\Gamma(\mathcal{M}(\mathbf{p}_1))$. The set S_1^* is the *consensus* and contains the *inliers* with respect to the current parameters $\mathbf{p}_1$. The elements of $\mathsf{D} \setminus \mathsf{S}_1^*$ are termed *outliers* with respect to the current model.

3. If the ratio $(\#\mathsf{S}_1^*)(\#\mathsf{D})^{-1}$ is greater than or equal to a preset threshold r, then determine the parameters $\mathbf{p}_1^*$ using S_1^* and quit. If $(\#\mathsf{S}_1^*)(\#\mathsf{D})^{-1} < r$ and c is smaller than the boundary of iterations $c_{\max}$, then go to step 1. Otherwise go to step 4.

4. The maximum number of iterations $c_{\max}$ is reached without finding model parameters gaining sufficient support. By choice of the user, report the parameters supported by the largest consensus or quit without a solution.

One first property to observe is that the RANSAC algorithm is iterative, but does not require an initial solution. Second, we are able to find a reasonable value of $c_{\max}$ assuming a certain ratio i of inliers in the data. We do so by considering the probability p_{in} that we at least once select k inliers out of D in N tries. The inverse formulation yields the probability $(1 - i^k)^N$ not to select k inliers in N tries, which equals $1 - p_{\text{in}}$. Thus, we have to select a confidence level, for instance, $p_{\text{in}} = 0.9$, and find

$$N = \frac{\log(1 - p_{\text{in}})}{\log(1 - i^k)}. \tag{3.73}$$

The maximal number of iterations supporting a minimal probability p_{in} to at least once select a set of k inliers is given by

$$c_{\max} = \lfloor N + 1 \rfloor. \tag{3.74}$$

Example A – a 2D application. A classical application of the RANSAC algorithm is the robust estimation of the epipolar geometry using image points. This application employs the assumption that we have detected a set of feature points in each image and that these features are at least partially overlapping in the sense of mapping the same 3D scene points. The data D hence consists of the two sets of image points, the model $\mathcal{M}(\mathbf{p})$ is the epipolar geometry, and the parameter vector $\mathbf{p}$ may hold the elements of the fundamental matrix $\mathbf{F}$. Depending on the algorithm to estimate $\mathbf{F}$, RANSAC randomly establishes seven or eight image correspondences, respectively, computes the according fundamental matrix, and evaluates the distances between image points and their corresponding epipolar lines. The actual task performed by the RANSAC algorithm is the estimation of a proper fundamental matrix. As a side effect, we yield the 2D point correspondences used to compute the final set of parameters.

Example B – a 3D application. Similarly to the previous example, RANSAC may estimate a 3D Euclidean transformation to align two sets of 3D points. One iteration performs a random selection of three 3D correspondences to fix 3D rotation and translation. As an error measure, RANSAC uses the Euclidean distances between a transformed point and the closest one of the other point set.

The RANSAC algorithm is a powerful tool for parameter estimation and allows various extensions for particular applications. Nonetheless, the computational burden quickly converges to infeasibility for large, but practically relevant data sets if we directly apply the RANSAC scheme without further adaptations. For these cases, another possibility is to dramatically decrease the possibility p_{in}, which is likely to leave us without any or a bad solution. Further problems arise if we are not provided with the ratio of inliers i in the data.

3.5.2 The Correspondence Problem for Image Data

A general random algorithm, just like RANSAC, is able to solve most different problems. In the particular case of finding correspondences in two intensity images, the classical RANSAC approach does not directly use the information contained in the images. In the following, we briefly describe the

two most important basic methods that explicitly use the image contents to establish 2D point correspondences.

3.5.2.1 Stereo Matching

Stereo matching, or block matching or template matching, assumes a known relative pose between the images, i. e. between the cameras from which the images originate. In fact, stereo matching assumes that the images are simultaneously shot by cameras that build a stereo setup. A stereo setup consists of two cameras that are identically oriented and only translated along one image axis. The translation distance is again called baseline, cf. Figure 3.4. In opposite to a general setup of two cameras, the baseline is the only parameter of a stereo camera system. The work of Fusiello et al. [Fusiello et al., 1998] shows how to transform general image pairs such that the transformed images are the mappings of an according stereo camera setup up to perspective occlusion effects.

Considering epipolar geometry and Figure 3.5, we observe that all epipolar lines in one image run through the epipole. The epipole marks the point where the connection of the two camera centers meets the image plane. A stereo setup features image planes that are located in the same world plane. Therefore, each image plane is parallel to the connection of the camera centers, and the epipole lies at infinity. Since the epipole is the intersection point of all epipolar lines and it is at infinity, the epipolar lines are parallel for a stereo setup. We additionally know that an image point in the first image and the corresponding epipolar line in the second image do have identical y-coordinates, assumed that the cameras are shifted to each other along their x-axes.

The above statements show correspondence search in rectified images to be comfortable in the following way. Regarding a reference point $\mathbf{x}_1 = (x, y)^\mathsf{T}$ in the first image, the search space for the corresponding point in the second image theoretically is the line $l_2 = y$. Thus, the search space is not the whole 2D image, but the respective 1D line. As a measure of similarity that finally fixes the position of the correspondence, stereo matching uses the normalized cross-correlation. Using intensity functions $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ that respond the intensity value at position $\mathbf{x}$ and regarding image templates of size $N \times N$,

the normalized cross-correlation is

$$\xi(\mathbf{x}_1, \mathbf{x}_2) = \sum_{x,y=-\frac{N}{2}}^{\frac{N}{2}} \frac{\left(I_1\left(\mathbf{x}_1 + (x,y)^\mathsf{T}\right) - \mu_{I_1,\mathbf{x}_1,N}\right)\left(I_2\left(\mathbf{x}_2 + (x,y)^\mathsf{T}\right) - \mu_{I_2,\mathbf{x}_2,N}\right)}{\sigma_{I_1,\mathbf{x}_1,N}\sigma_{I_2,\mathbf{x}_2,N}} \tag{3.75}$$

with mean values $\mu_{I_j,\mathbf{x}_k,N}$ and standard deviations $\sigma_{I_j,\mathbf{x}_k,N}$ in the respective templates. By means of the normalized cross-correlation, one yields an image position $\mathbf{x}_2 = (x_2, y_2)^\mathsf{T}$ corresponding to $\mathbf{x}_1 = (x_1, y_1)^\mathsf{T}$ that provides the highest similarity with respect to the considered template size. Since $y_1 = y_2$, the corresponding point $\mathbf{x}_2$ is uniquely defined by $\mathbf{x}_1$ and the difference $x_2 - x_1$, which is called the *disparity* of the corresponding image points. Due to the formulation of stereo matching as a 1D search, most algorithms consider only integer positions, so $x_1, x_2 \in \mathbb{Z}$. This limits the achievable accuracy of the according algorithms in terms of image resolution. In particular for large baselines, stereo matching methods are encountered with the problem of occlusions in the images, which change the appearances of corresponding templates. One way to ease this effect is to select a smaller baseline, but this step negatively effects the possible accuracy since it shrinks the whole range of possible disparities to a smaller integer interval of constant resolution.

Despite an elegant theoretical restriction of the image search space, stereo matching has some drawbacks. Performing a search itself inhibits performance. Gaining accuracy by searching real-valued disparities further boosts computation time. And in the end, noise and calibration errors soften the 1D restriction of the search space.

3.5.2.2 Feature Tracking

An approach opposing stereo matching in many ways is feature tracking. Evaluating intensity values inside an image template is the main similarity of the approaches. In opposite to stereo matching, feature tracking, namely KLT tracking as depicted in [Lucas and Kanade, 1981], employs an error measure based on intensity differences,

$$\epsilon = \sum_{x,y=-\frac{N}{2}}^{\frac{N}{2}} \left(I_2(W((x,y)^\mathsf{T})) - I_1((x,y)^\mathsf{T})\right)^2, \tag{3.76}$$

in order to estimate a warping W that maps the original template to the corresponding position in the consecutive image. Possible knowledge of the

relative camera pose will be ignored by this tracking method. Implicitly, this approach assumes that the mapping of a 3D world point and its close surrounding does not change seriously between the images. As two fundamental requirements follow the small-baseline and brightness-constancy assumptions. Once a transformation W provides the feature position in the second image, it is possible to continue the process in further images. This fact and the mentioned assumptions alike favor feature tracking for the application with video sequences, which implement the small-baseline assumption by nature. Furthermore, the formulation of the error function (3.76) allows to derive a non-linear optimization procedure to solve for W without searching.

Considering the properties of stereo matching and feature tracking, for some applications it is a desirable goal to combine the advantages of both methods. In particular, we would like to use an optimization procedure, just like feature tracking, and apply prior knowledge of camera parameters to formulate constraints on the search space, like stereo matching does. Since view planning using a passive camera is one of the mentioned applications, we present a possible, novel way to reach this goal, the *Guided KLT tracking*, in Chapter 4.

3.5.3 The Correspondence Problem for 3D Data

Inherently, the 3D correspondence problem is the same as in 2D images. Nonetheless, the bulk of practical applications reveals the main difference that 2D images provide a further point attribute, which is the intensity value. On the contrary, most of the practical 3D correspondence problems deal with finite, discrete sets of 3D points. A conceptual resemblance to RANSAC is hence notable for 3D correspondence algorithms that use not more than 3D point coordinates.

3.5.3.1 The Iterative-closest-points Algorithm

A famous example of a deterministic, iterative algorithm to establish 3D point correspondences is the iterative-closest-points (ICP) algorithm based on [Besl and McKay, 1992]. We describe the algorithm using two point sets $\mathsf{P}_1 = \{\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{M,1}\}$ and $\mathsf{P}_2 = \{\mathbf{X}_{1,2}, \ldots, \mathbf{X}_{N,2}\}$, whereat we consider P_1 to be the data set that shall be rotated and translated to best fit the model set P_2. Besl and McKay underline that these two entities may represent any geometric data that allow to find a registration, while the data set has to

be transformed to an actual point set P_1. The ICP algorithm acts in the following way:

1. Initialize a transformation $\mathbf{T} = \mathbf{I}_4$ as in (3.5).
2. For each point $\mathbf{X}_{j,1}$ of the data set P_1, find the closest point in the model set P_2.
3. Determine a transformation $\mathbf{T}_H$ using the correspondences found in step 2.
4. Transform the points of P_1 by $\mathbf{T}_H$. Update $\mathbf{T}$ by setting it to $\mathbf{T}_H\mathbf{T}$.
5. If the changes performed by $\mathbf{T}_H$ fall below a threshold t, then report $\mathbf{T}$ and quit. Otherwise go to step 2.

The literature offers a great deal of enhancements and adaptations of the ICP algorithm for particular problems. All of them have in common that the success strongly depends on a proper initial solution, i. e. a favorable initial state of P_1. In [Zinßer et al., 2003] the authors observe that the initialization has to be inside the basin of convergence of the true solution. Likewise, the quality of the initialization obviously has a strong impact on the runtime behavior of the algorithm.

3.5.3.2 Non-iterative Approaches

Analogous to the intensity point attribute for the case of 2D image correspondences, there are direct approaches to establish 3D correspondences that use more information than just point coordinates, for example, the knowledge of a 3D surface triangulation. We shift an in-depth discussion of these approaches to Chapter 5, where we present a novel way of directly computing invariant features of 3D surface triangulations that allow coarse 3D registration. One particular area of application of this new method is to produce suitable initial solutions for the ICP algorithm.

Chapter 4

Guided KLT Feature Tracking for Controlled Environments

As motivated and stated in Section 3.5.2, it is a promising goal to combine the advantages of stereo matching and feature tracking. Therefore, in this chapter we present a novel way to use knowledge of camera parameters for feature tracking regarding uncertainty, the *Guided KLT (GKLT) tracking method*. In doing so, we both emphasize the self-contained application of GKLT tracking and the important part it has in the view planning approach depicted in this work. Concerning the latter fact, we highlight the fundamental meaning of GKLT tracking for our approach to view planning in Section 4.1. In tribute to the independent application, we use this chapter to separately examine specific references in Section 4.2, to present the basic KLT method in Section 4.3 and the GKLT tracking method itself in Section 4.4, and to provide an experimental evaluation and an outlook to future work with respect to GKLT tracking in Sections 4.5 and 4.6.

4.1 Applications and Relevance to View Planning

As we will support by the following sections and the experimental evaluation, GKLT tracking gains some benefits compared to standard KLT tracking by incorporating additional prior knowledge of camera parameters. In particular, GKLT tracking endows better tracking accuracy and longer tracking

duration by means of softly restricting the optimization space. These benefits make the method valuable for any application of feature tracking given camera parameters, for instance, when using a robotic arm as in Figure 1.1 or a turntable as in [Kuehmstedt et al., 2001]. Provided that GKLT tracking improves the result, we can even think of a mutual optimization process for the case of unknown camera parameters, which we outline in Section 4.6.

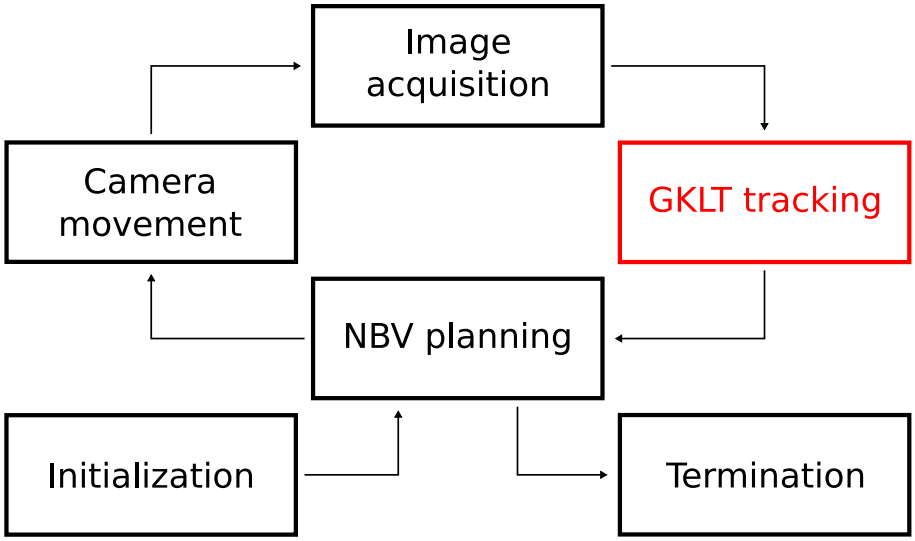


Figure 4.1: Guided KLT feature tracking is a fundamental component of the view planning method proposed in this work.

Returning to this work, we now concern ourselves with the task of view planning using a passive camera. We realize that tackling the correspondence problem is a natural starting point necessary for any approach to our specific view planning problem. As a first decision, we favor feature tracking over stereo matching for the reasons formulated in Section 3.5.2. Second, we observe that view planning comes with sets of prior knowledge K, goals G, constraints C, and actively adjusted parameters P, and we try to improve feature tracking using these sets. The information contained in G, C, and P mainly effects the purposive actions of the planning system. With respect to feature tracking, we may consider to actively change camera parameters in a way that promises better tracking results, but this has nothing to do with the actual tracking process and a justification depends on the contents of G. In the end we find that we at least feature additional prior knowledge

K containing information about camera parameters. Therefore, we present a novel method to incorporate information about camera parameters into feature tracking. We further augment the extended feature tracking with robust online 3D reconstruction, detection of tracking outliers, reinitialization of lost features, and 3D uncertainty estimation. By this means, the final GKLT tracking method provides the data basis for the active accuracy optimization by view planning presented in Chapter 6.

4.2 Previous Work

The original idea of tracking features by an iterative optimization process was presented by Lucas and Kanade in [Lucas and Kanade, 1981] and will be reviewed in Section 4.3. Since then a rich variety of adaptations and extension has been published, giving rise to surveys like [Baker and Matthews, 2004]. For instance, Fusiello et al. [Fusiello et al., 1999] handle the removal of spurious correspondences by using robust statistics. The problem of reselecting the template image is dealt with in [Zinßer et al., 2005]. Since these modifications and extensions are independent from applying camera parameters, we only mention very few of them. For more information the reader may be referred to [Baker and Matthews, 2004]. Heigl [Heigl, 2003] presents an example of applying knowledge of camera parameters to feature tracking. In his work, he moves features along their epipolar lines, but does not regard uncertainty.

4.3 Original KLT Feature Tracking

In this section we review a way to derive the optimization rule of the original KLT feature tracking [Lucas and Kanade, 1981].

Considering an initial template image, or frame, with an intensity function $T(\mathbf{x})$ giving the intensity at position $\mathbf{x} = (x, y)^\mathsf{T}$, the feature at position $\mathbf{x}$ is given by the intensity values inside of a region F around $\mathbf{x}$ called feature patch. In general, the set F of image coordinates belonging to the considered feature patch may contain any image positions. In spite, most of the applications use square regions centered at the considered feature position. Given another image with intensity function $I(\mathbf{x})$, feature tracking tries to estimate the optimal parameters $\mathbf{p}^*$ of a warping function $W(\mathbf{x}, \mathbf{p})$, $\mathbf{p} = (p_1, \ldots, p_n)^\mathsf{T}$,

in the sense of

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \epsilon(\mathbf{p}) \tag{4.1}$$

using the error measure

$$\epsilon(\mathbf{p}) = \sum_{\mathbf{x}\in\mathsf{F}} \left(I(W(\mathbf{x},\mathbf{p})) - T(\mathbf{x})\right)^2. \tag{4.2}$$

A common choice for $W(\mathbf{x},\mathbf{p})$ is affine warping by

$$W^a(\mathbf{x},\mathbf{p}^a) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \tag{4.3}$$

with $\mathbf{p}^a = (\Delta x, \Delta y, a_{11}, a_{12}, a_{21}, a_{22})^\mathsf{T}$. Following the additional approach, we reformulate (4.2) and yield

$$\epsilon(\Delta\mathbf{p}) = \sum_{\mathbf{x}\in\mathsf{F}} \left(I(W(\mathbf{x},\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x})\right)^2. \tag{4.4}$$

In order to resolve the functional dependencies between $\mathbf{p}$ and $\Delta\mathbf{p}$, we twice apply first-order Taylor approximation, which makes

$$\epsilon(\Delta\mathbf{p}) \approx \sum_{\mathbf{x}\in\mathsf{F}} \left(I(W(\mathbf{x},\mathbf{p}) + \nabla_\mathbf{p} W(\mathbf{x},\mathbf{p})\Delta\mathbf{p}) - T(\mathbf{x})\right)^2 \tag{4.5}$$

$$\approx \sum_{\mathbf{x}\in\mathsf{F}} \left(I(W(\mathbf{x},\mathbf{p})) + \nabla I \nabla_\mathbf{p} W(\mathbf{x},\mathbf{p})\Delta\mathbf{p} - T(\mathbf{x})\right)^2 = \epsilon'(\Delta\mathbf{p}). \tag{4.6}$$

using the gradient $\nabla I = (I_x, I_y) = \left(\frac{\partial I(\mathbf{x})}{\partial x}, \frac{\partial I(\mathbf{x})}{\partial y}\right)$ of the intensity function $I(\mathbf{x})$ and the Jacobian of the warping function,

$$\nabla_\mathbf{p} W(\mathbf{x},\mathbf{p}) = \begin{pmatrix} \frac{\partial W_x(\mathbf{x},\mathbf{p})}{\partial p_1} & \cdots & \frac{\partial W_x(\mathbf{x},\mathbf{p})}{\partial p_n} \\ \frac{\partial W_y(\mathbf{x},\mathbf{p})}{\partial p_1} & \cdots & \frac{\partial W_y(\mathbf{x},\mathbf{p})}{\partial p_n} \end{pmatrix} \tag{4.7}$$

with

$$W(\mathbf{x},\mathbf{p}) = \begin{pmatrix} W_x(\mathbf{x},\mathbf{p}) \\ W_y(\mathbf{x},\mathbf{p}) \end{pmatrix}. \tag{4.8}$$

We differentiate $\epsilon'(\Delta\mathbf{p})$ with respect to $\Delta\mathbf{p}$,

$$\frac{\partial\epsilon'(\Delta\mathbf{p})}{\partial\Delta\mathbf{p}} = 2\sum_{\mathbf{x}\in\mathsf{F}} \left(\nabla I \nabla_\mathbf{p} W(\mathbf{x},\mathbf{p})\right)^\mathsf{T} \left(I(W(\mathbf{x},\mathbf{p})) + \nabla I \nabla_\mathbf{p} W(\mathbf{x},\mathbf{p})\Delta\mathbf{p} - T(\mathbf{x})\right) \tag{4.9}$$

and set it to zero,

$$\frac{\partial \epsilon'(\Delta \mathbf{p})}{\partial \Delta \mathbf{p}} \stackrel{!}{=} \mathbf{0}. \tag{4.10}$$

By rearranging we get

$$\begin{aligned} &\overbrace{\sum_{\mathbf{x} \in \mathsf{F}} \left(\nabla I \nabla_{\mathbf{p}} W(\mathbf{x}, \mathbf{p})\right)^{\mathsf{T}} \left(\nabla I \nabla_{\mathbf{p}} W(\mathbf{x}, \mathbf{p})\right)}^{\mathbf{H} \in \mathbb{R}^{n \times n}} \Delta \mathbf{p} \\ &\qquad = \sum_{\mathbf{x} \in \mathsf{F}} \left(\nabla I \nabla_{\mathbf{p}} W(\mathbf{x}, \mathbf{p})\right)^{\mathsf{T}} \left(T(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p}))\right) \end{aligned} \tag{4.11}$$

and

$$\Delta \mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x} \in \mathsf{F}} \left(\nabla I \nabla_{\mathbf{p}} W(\mathbf{x}, \mathbf{p})\right)^{\mathsf{T}} \left(T(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p}))\right) \tag{4.12}$$

using the Hessian $\mathbf{H}$, which is an approximate Hessian with respect to the original error function (4.4). Equation (4.12) provides us with an iterative optimization rule that performs a locally approximated gradient descent. At the end of the i-th iteration we find the update $\mathbf{p}_{i+1} \leftarrow \mathbf{p}_i + \Delta \mathbf{p}_i$.

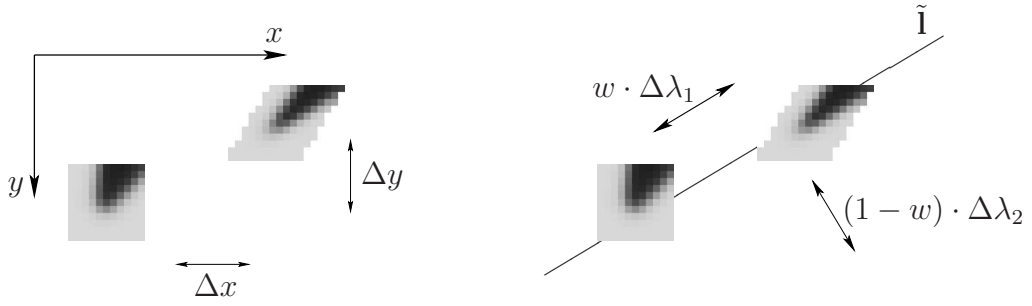
KLT tracking offers a purely appearance-based way to solve the 2D correspondence problem in image data. Iteratively adapting warping parameters demands small changes of the respective feature patch, which is stated as the small-baseline assumption and may be interpreted to further assume small rotations. The objective function (4.4) of the optimization implies the requirement of constant brightness. Further, a warping function cannot ever model transformations of the feature patch caused by perspective occlusions. This is due to the nature of the image formation as a 3D-2D mapping. Hence, the assumption of planarity applies, i. e. the assumption that the feature patch is the projection of a plane world region to the image. One inherent problem of all feature tracking methods is the goal to find mappings of the same 3D world point without any reference to the 3D entity, but just using image mappings. As consequences, the well-known motion-drift problem can occur, cf. [Rav-Acha and Peleg, 2006], and intensive care has to be taken for the selection of good features to track, cf. [Shi and Tomasi, 1994]. Violating the theoretical assumptions may cause an early loss of tracked features, i. e. low tracking duration, and low tracking accuracy of tracked features.

4.4 GKLT Tracking Using Known Camera Parameters in Consideration of Uncertainty

Considering the task of feature tracking and, in particular, the KLT tracking method, we pose the question how we can apply prior knowledge of camera parameters to improve feature tracking. Referring to Section 3.4, we note that we are able to compute the fundamental matrix $\mathbf{F}$ between two images using known camera parameters. Once yielded, the fundamental matrix $\mathbf{F}$ theoretically constrains the position $\mathbf{m}$ of a feature in the reference frame to a line

$$\tilde{\mathrm{l}} = \mathbf{F}\tilde{\mathbf{m}}. \tag{4.13}$$

In the following, we describe how we can use this constraint for feature tracking, see also [Trummer et al., 2008, Trummer et al., 2009b, Trummer et al., 2009a, Trummer et al., 2009c].



(a) Translation along image axes.

(b) Translation along and perpendicular to the epipolar line.

Figure 4.2: The epipolar geometry theoretically constrains the feature translation along the epipolar line. Section 4.4 provides further explanations.

4.4.1 GKLT Tracking with Manual Uncertainty Adjustment

Again, we regard an image feature at position $\mathbf{m}$ in the reference image and the fundamental matrix $\mathbf{F}$ known in terms of camera parameters describing the relative pose between the reference image and the consecutive

image. We know that the image point corresponding to $\mathbf{m} = (x_m, y_m)^\mathsf{T}$ in the second image is located on the epipolar line $\tilde{\mathbf{l}} = (l_1, l_2, l_3)^\mathsf{T}$ as in (4.13). A parameterized form of this line is

$$\mathbf{l}(\lambda) = \begin{pmatrix} \frac{-l_3}{l_1} \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} -l_2 \\ l_1 \end{pmatrix} \tag{4.14}$$

with parameter λ and $l_1 \neq 0$. Modeling pure translation of point $\mathbf{x} = (x, y)^\mathsf{T}$, we find an epipolar warping function

$$W_E^t(\mathbf{x}, \mathbf{p}_E^t, \mathbf{m}) = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{-l_3}{l_1} - \lambda l_2 \\ \lambda l_1 \end{pmatrix} \tag{4.15}$$

using $\mathbf{p}_E^t = \lambda$ and $(l_1, l_2, l_3)^\mathsf{T} = \mathbf{F}\tilde{\mathbf{m}}$. The according Jacobian is simply

$$\nabla_{\mathbf{p}_E^t} W_E^t(\mathbf{x}, \mathbf{p}_E^t, \mathbf{m}) = \begin{pmatrix} \frac{\partial W_{E,x}^t(\mathbf{x}, \mathbf{p}_E^t, \mathbf{m})}{\partial \lambda} \\ \frac{\partial W_{E,y}^t(\mathbf{x}, \mathbf{p}_E^t, \mathbf{m})}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} -l_2 \\ l_1 \end{pmatrix}. \tag{4.16}$$

We present a list of possible instances of the warping function $W(\mathbf{x}, \mathbf{p}, \mathbf{m})$ and the respective Jacobians in Appendix A.

At this point, we formulate a warping function $W_E(\mathbf{x}, \mathbf{p}_E, \mathbf{m})$ such that only one parameter λ defines the translation. Compared to the conventional modeling of translation as in (4.3), this reduces the optimization parameter space by one degree of freedom, which favors a more efficient optimization and more stable results. Since the feature position is restricted to the corresponding epipolar line, we also ease the requirement of a true 2D optical feature structure. By means of the positional restriction, it is also valid if the feature describes a 1D edge that is not parallel to the epipolar line. Combining such a 1D optical feature structure and the 1D positional restriction we yield, in general, the unique feature position. Therefore we find more image features valid for tracking.

Despite the great advantages of a restricted parameter space of the optimization, we have to pay attention to possibly noisy prior knowledge. Errors in the given camera parameters may cause an deficient fundamental matrix and hence a deficient positional restriction. For the sake of modeling the feature position with respect to a noisy fundamental matrix, we have to allow translations not along the epipolar line, for which we choose a translation perpendicular to the epipolar line. The according translation model along

the epipolar line but respecting uncertainty is

$$W^t_{EU}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m}) = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{-l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \tag{4.17}$$

with $\mathbf{p}^t_{EU} = (\lambda_1, \lambda_2)^\mathsf{T}$, $l_1 \neq 0$. The parameters λ_1 and λ_2 describe the translation along and perpendicular to the corresponding epipolar line, respectively. Thus we identify λ_1 with an epipolar translation and λ_2 with uncertainty. Now, if we apply the modified warping function (4.17) to the formulation of the standard KLT objective function in (4.2), we do theoretically reach no change but optimizing translation inside a coordinate frame rotated with respect to the image axes. Despite these facts, we have augmented the directions of translation with special meaning. To give the example of perfectly accurate camera parameters, we are able to set $\lambda_2 = 0$ and perform a 1D optimization of the epipolar translation λ_1. As a consequence of the special meaning of the directions of translation, we introduce a weighting factor $w \in [0, 1]$, the *epipolar weight*, that uses a weighting matrix

$$\mathbf{A}^t_w = \begin{pmatrix} w & 0 \\ 0 & 1-w \end{pmatrix} \tag{4.18}$$

to control which translations are finally executed. We reach this goal by adapting the standard KLT objective function (4.2) with respect to the general epipolar warping model $W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})$ and finally applying the weighting matrix $\mathbf{A}_w$,

$$\Delta\mathbf{p}_{EU,w} = \mathbf{A}_w \mathbf{H}^{-1}_{EU} \sum_{\mathbf{x}\in\mathsf{F}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}))^\mathsf{T} (T(\mathbf{x}) - I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}))) \tag{4.19}$$

using an adapted approximate Hessian $\mathbf{H}_{EU}$ according to (4.11). For another warping model, for example, the affine model, the according weighting matrix $\mathbf{A}^a_w$ is likewise a diagonal matrix weighting the parameter changes $\Delta\lambda_1$ and $\Delta\lambda_2$ as in (4.18), but applying weight 1 to each of the other parameters.

At this stage, we optimize the translational part of the warping function $W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})$ in directions depending on the epipolar geometry. Standard KLT tracking uses translations along the image axes. These directions are arbitrary regarding the actual movement of the feature position. The modified warping function uses the epipolar geometry to *guide* the feature along the corresponding epipolar line, but allows deviations due to uncertainty. Regulating possible deviations from the epipolar line is in control of

the epipolar weight $w \in [0, 1]$. In particular, $w = 1$ ignores any translation perpendicular to the epipolar line and thus corresponds to a perfectly exact epipolar geometry. On the contrary, $w = 0.5$ does not favor any direction and hence applies to an absolutely uncertain epipolar geometry. By now, the Guided KLT (GKLT) tracking requires manual adjustment of a proper epipolar weight.

4.4.2 GKLT Tracking with Integrated Uncertainty Estimation

In the previous section we replaced the standard warping function of the KLT tracking method by a warping that modifies translations with respect to the epipolar geometry. Additionally, we extended the final parameter update rule by a weighting matrix allowing manual tuning of the optimization method regarding the level of uncertainty. This kind of manual uncertainty control is adverse for cases where the uncertainty of the camera parameters is unknown. Thus it is a valuable goal to integrate an uncertainty estimation into the GKLT tracking method. We reach this goal by incorporating the weighting matrix not just into the final update rule, but into the initial objective function of the optimization. In doing so, we propose an EM-like, cf. [Dempster et al., 1977], alternating optimization scheme that concurrently estimates the warping parameters as well as the uncertainty of the epipolar geometry.

Modifying the objective function. Instead of extending the final update rule (4.12) of the standard KLT method, we now directly apply the weighting matrix to the objective function (4.4) yielding

$$\epsilon(\Delta\mathbf{p}_{EU}, \Delta w) = \sum_{\mathbf{x}\in\mathsf{F}} (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU} + \mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU}, \mathbf{m})) - T(\mathbf{x}))^2. \quad (4.20)$$

In analogy to the additional approach regarding parameter changes, (4.20) uses the reparameterized weighting matrix

$$\mathbf{A}_{w,\Delta w} = \begin{pmatrix} w + \Delta w & 0 & 0 & \cdots & 0 \\ 0 & 1 - (w + \Delta w) & 0 & & \\ 0 & 0 & 1 & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & & & \cdots & 0 \quad 1 \end{pmatrix}. \quad (4.21)$$

First-order Taylor approximations help isolating the terms to optimize,

$$\begin{aligned}
&\epsilon(\Delta\mathbf{p}_{EU}, \Delta w) \\
&\approx \sum_{\mathbf{x}\in\mathsf{F}} (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}) + \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU}) - T(\mathbf{x}))^2 \qquad (4.22) \\
&\approx \sum_{\mathbf{x}\in\mathsf{F}} (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})) + \nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU} - T(\mathbf{x}))^2 \\
&= \epsilon'(\Delta\mathbf{p}_{EU}, \Delta w) \qquad (4.23)
\end{aligned}$$

and make up a linearized approximation $\epsilon'(\Delta\mathbf{p}_{EU}, \Delta w)$ of the objective function. In particular, (4.23) allows differentiation with respect to the warping parameters and the epipolar weight.

Optimizing the warping parameters. Using the linearization (4.23) of the objective function, we differentiate with respect to the warping parameters and set to zero,

$$\begin{aligned}
&\frac{\partial\epsilon'(\Delta\mathbf{p}_{EU}, \Delta w)}{\partial\Delta\mathbf{p}_{EU}} \\
&= 2\sum_{\mathbf{x}\in\mathsf{F}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w})^{\mathsf{T}} (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})) \\
&+ \nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU} - T(\mathbf{x})) \stackrel{!}{=} \mathbf{0}, \qquad (4.24)
\end{aligned}$$

which provides us with an optimality criterion for changing the warping parameters. Rearranging (4.24) yields

$$\begin{aligned}
&\overbrace{\sum_{\mathbf{x}\in\mathsf{F}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w})^{\mathsf{T}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w})}^{\mathbf{H}_{\mathbf{p}_{EU}} \in \mathbb{R}^{n\times n}} \Delta\mathbf{p}_{EU} \\
&= \sum_{\mathbf{x}\in\mathsf{F}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w})^{\mathsf{T}} (T(\mathbf{x}) - I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}))) \qquad (4.25)
\end{aligned}$$

and

$$\Delta\mathbf{p}_{EU} = \mathbf{H}_{\mathbf{p}_{EU}}^{-1} \sum_{\mathbf{x}\in\mathsf{F}} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m})\mathbf{A}_{w,\Delta w})^{\mathsf{T}} (T(\mathbf{x}) - I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}))). \qquad (4.26)$$

Based on the image data and the current states of $\mathbf{p}_{EU}$ and w, (4.26) allows to compute an optimal parameter change in the sense of steepest descent. In this step, we set $\Delta w = 0$ and shift the computation of an optimal change of the uncertainty estimate to the next stage.

Optimizing the uncertainty parameter. For the adaptation of the epipolar weight, which in a dual way defines the uncertainty of the epipolar geometry, we follow the same approach as for the warping parameters. We start by differentiating (4.23) with respect to the change of the epipolar weight,

$$\begin{aligned}
&\frac{\partial \epsilon'(\Delta \mathbf{p}_{EU}, \Delta w)}{\partial \Delta w} \\
&= 2 \sum_{\mathbf{x} \in \mathsf{F}} \left(\frac{\partial}{\partial \Delta w} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}) \mathbf{A}_{w,\Delta w} \Delta \mathbf{p}_{EU}) \right) (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU})) \\
&+ \nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}) \mathbf{A}_{w,\Delta w} \Delta \mathbf{p}_{EU} - T(\mathbf{x})) \stackrel{!}{=} \mathbf{0}.
\end{aligned} \tag{4.27}$$

We specify

$$\begin{aligned}
&\frac{\partial}{\partial \Delta w} (\nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}) \mathbf{A}_{w,\Delta w} \Delta \mathbf{p}_{EU}) \\
&\qquad = \nabla I \nabla_{\mathbf{p}_{EU}} W_{EU}(\mathbf{x}, \mathbf{p}_{EU}, \mathbf{m}) \frac{\partial \mathbf{A}_{w,\Delta w}}{\partial \Delta w} \Delta \mathbf{p}_{EU}
\end{aligned} \tag{4.28}$$

and find

$$\frac{\partial \mathbf{A}_{w,\Delta w}}{\partial \Delta w} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & & \\ 0 & 0 & 0 & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & & \cdots & 0 & 0 \end{pmatrix}. \tag{4.29}$$

Rearranging (4.27) and applying (4.28) yields

$$\overbrace{\sum_{\mathbf{x}\in\mathsf{F}}(\nabla I\nabla_{\mathbf{p}_{EU}}W_{EU}(\mathbf{x},\mathbf{p}_{EU},\mathbf{m})\frac{\partial\mathbf{A}_{w,\Delta w}}{\partial\Delta w}\Delta\mathbf{p}_{EU})(\nabla I\nabla_{\mathbf{p}_{EU}}W_{EU}(\mathbf{x},\mathbf{p}_{EU},\mathbf{m}))}^{\mathbf{h}_w\in\mathbb{R}^n}\,\mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU}$$

$$=\underbrace{\sum_{\mathbf{x}\in\mathsf{F}}(\nabla I\nabla_{\mathbf{p}_{EU}}W_{EU}(\mathbf{x},\mathbf{p}_{EU},\mathbf{m})\frac{\partial\mathbf{A}_{w,\Delta w}}{\partial\Delta w}\Delta\mathbf{p}_{EU})(T(\mathbf{x})-I(W_{EU}(\mathbf{x},\mathbf{p}_{EU},\mathbf{m})))}_{e\in\mathbb{R}},$$

$$\text{i.e.}\qquad \mathbf{h}_w\mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU}=e. \tag{4.30}$$

This is one linear equation in the unknown Δw, since $\mathbf{h}_w=(h_1,\ldots,h_n)$ is a row vector and $\Delta\mathbf{p}_{EU}=(\Delta\lambda_1,\Delta\lambda_2,\Delta p_3,\ldots,\Delta p_n)^\mathsf{T}$ is a column vector. Hence we can reformulate (4.30) and get

$$\Delta w=\frac{e-h_2\Delta\lambda_2-h_3\Delta p_3-\ldots-h_n\Delta p_n}{h_1\Delta\lambda_1-h_2\Delta\lambda_2}-w. \tag{4.31}$$

By means of the image data, the current state of the epipolar weight, and the last change of the warping parameters, we reach a new estimation of the epipolar weight, which also represents a new estimation of the uncertainty of the given epipolar geometry.

Modifying the optimization algorithm. In comparison to the KLT and GKLT tracking, we now have two update rules: one for $\mathbf{p}_{EU}$ and one for w. These update rules, just as in the previous KLT versions, compute optimal parameter changes in the sense of least-squares estimation found by steepest descent of an approximated error function. We combine the two update rules in an EM-like approach. For one iteration of the optimization algorithm, we calculate $\Delta\mathbf{p}_{EU}$ (using $\Delta w=0$) followed by the computation of Δw with respect to the $\Delta\mathbf{p}_{EU}$ just computed in this step. Then we apply the change to the warping parameter using the updated w.

The modified optimization algorithm is as follows.

1. Initialize $\mathbf{p}_{EU}$ and w.
2. Compute $\Delta\mathbf{p}_{EU}$ by (4.26).

3. Compute Δw by (4.31) using $\Delta \mathbf{p}_{EU}$.

4. Update $\mathbf{p}_{EU}$: $\mathbf{p}_{EU} \leftarrow \mathbf{p}_{EU} + \mathbf{A}_{w,\Delta w} \Delta \mathbf{p}_{EU}$.

5. Update w: $w \leftarrow w + \Delta w$.

6. If changes are small, then stop; else go to step 2.

This novel optimization algorithm for feature tracking with known camera parameters uses the update rules derived from the extended optimization error function (4.20) for GKLT tracking. Most importantly, these steps provide a combined estimation of the warping and the uncertainty parameters. Hence, there is no more need to manually adjust the uncertainty parameter.

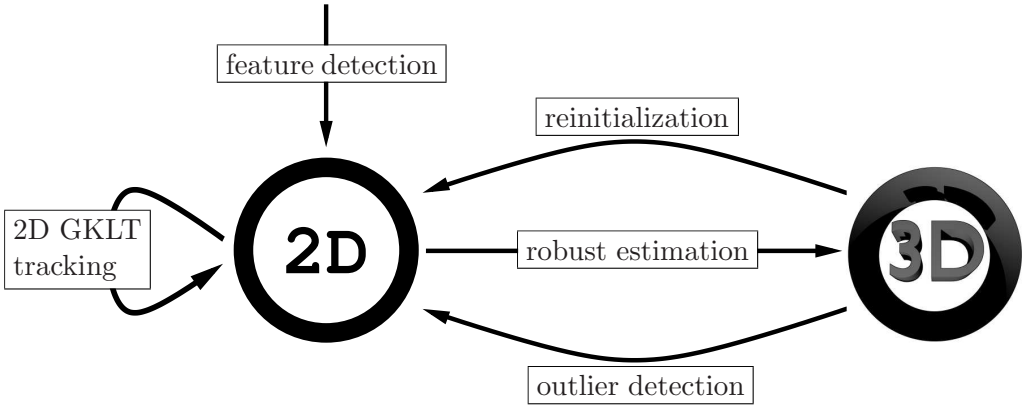


Figure 4.3: The GKLT$_{3D}$ framework from a relational point of view. Integrating 3D estimation in the tracking process allows additional extensions.

4.4.3 Combined GKLT Feature Tracking and 3D Reconstruction

All the current extensions of GKLT tracking refer to the optimization process using image data. However, knowledge of the camera parameters in combination with feature positions additionally infers on the 3D position of the respective world point. In the following we want to answer the question how a concurrent 3D estimation can improve feature tracking. We could argue that a 3D estimation is an implication of the above mentioned information and does hence not provide any additional advantage for the tracking process. In this sense, the standard approach of feature tracking using image

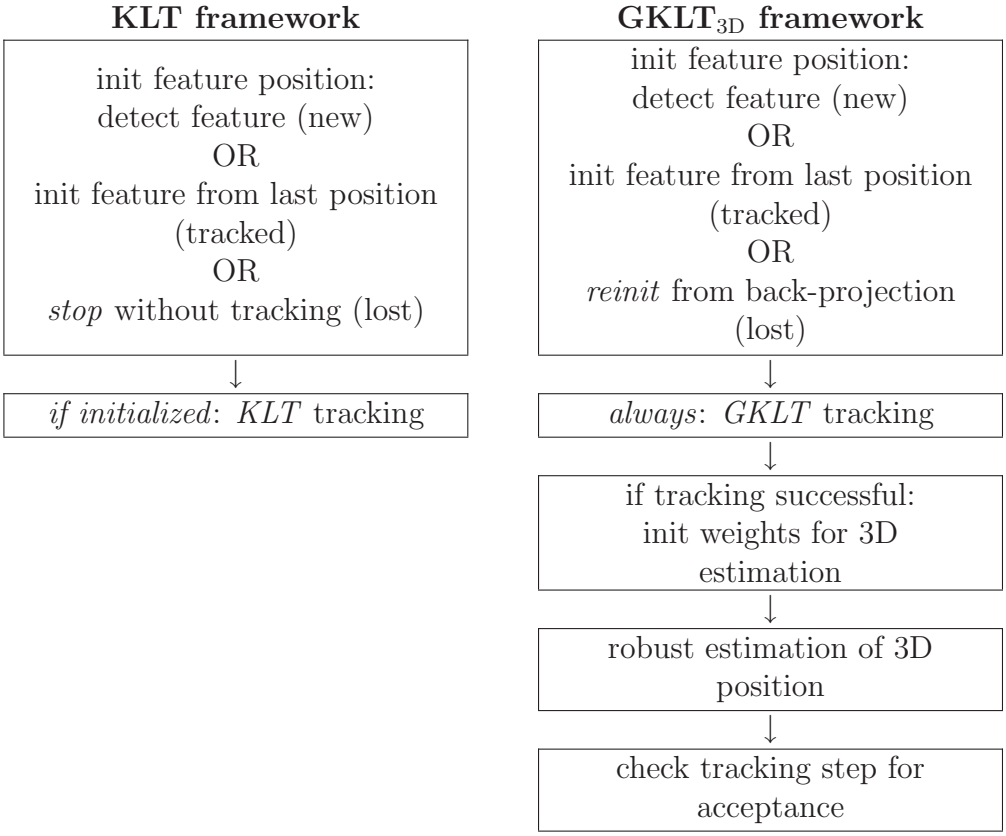
data and performing a single and final 3D reconstruction would be equivalent to feature tracking with a concurrent 3D estimation. On the other hand, the transformation parameters of the warping function do only respect the feature positions in the template and the current image, but reconstructed 3D point typically contains the information of all image points available. And this is the difference. A 3D reconstruction represents all the information regarding the positions of the specific feature gathered in the previous frames.

Figure 4.3 illustrates in a relational way that including the 3D estimation into the tracking process actually makes further extensions possible. The whole 3D framework includes 2D GKLT tracking with integrated uncertainty estimation, robust 3D estimation, reinitialization of lost features, and the detection of tracking outliers. We denote this framework by GKLT_{3D}. The GKLT_{3D} framework comprises the computational steps listed in the flowchart of Table 4.1. In the following we explain each component of the flowchart.

Re-/Initialize the feature position. Since tracking in the KLT sense is an iterative optimization of feature transformation parameters, an initial solution is required. If the feature was tracked in the previous frame, it is straightforward to use the last parameter estimation as the initialization for the current frame, which corresponds to the condition of small baselines between consecutive frames. We also use this initialization technique for GKLT_{3D}. In addition, GKLT_{3D} reinitializes features that were lost in the previous frame or earlier and that were tracked in at least one frame. Thus a 3D estimation from at least two frames exists, in particular from the frame where the feature was detected and from at least one frame of successful tracking. For lost features, we use the back-projection of the estimated 3D point to reinitialize the feature position for GKLT_{3D} tracking.

GKLT tracking. Having initialized the feature transformation, we perform 2D feature tracking by the GKLT method with integrated uncertainty estimation. In fact, this step of the GKLT_{3D} method can be performed by any other tracking method including standard KLT tracking. However, we find it natural to further extend the existing GKLT tracking method that already uses knowledge of camera parameters.

Table 4.1: A comparison of the KLT and GKLT$_{3D}$ frameworks in flowcharts. The listed steps describe the performed actions for tracking one feature in one frame. Further explanations are given in Section 4.4.3.



Initialize the weights for 3D estimation. After successfully tracking the feature, we include the additional information about the actual feature position in the 3D estimation. Since we use an iterative estimation and robust statistics, we need to initialize each weight $w_i \in [0,1]$ for the feature position $\mathbf{x}_i$ in frame i. The only w_i we can know for sure is $w_0 = 1$, since frame 0 is the initial frame where the feature is detected and hence defined. The feature positions tracked in the following frames are afflicted with increasing uncertainty. It is more likely for them to be outliers. Thus we propose a strictly decreasing sequence $(w_i^{(\text{init})})_{i=0,1,...,I}$ with

$$w_0^{(\text{init})} = 1 \text{ and } \forall i > 0 : w_i^{(\text{init})} < w_{i-1}^{(\text{init})} \tag{4.32}$$

as initialization for the weights w_i. In the presence of output weights from a previous 3D estimation, we initialize the position weights with

$$w_i^{(\text{init})} = \begin{cases} 1 & , \quad i = 0 \\ w_i^{(\text{prev})} & , \quad 1 \leq i \leq I-1 \\ 0.5 & , \quad i = I \end{cases} \tag{4.33}$$

and hence ensure that $w_0^{(\text{init})} = 1$, initialize the weight regarding the latest tracked position as $w_I^{(\text{init})} = 0.5$, and use the previously adapted weights $w_i^{(\text{prev})}$, $i = 1, ..., I-1$.

Robust 3D estimation. The core of the extensions discussed in this section is a robust estimation of the 3D position for the tracked feature, i. e. the position of the 3D world point that is mapped to the feature being tracked. For 3D reconstruction we use the known camera parameters and a robust adaptation of the standard direct linear transform (DLT) algorithm for 3D triangulation, see Section 3.3, to perform iteratively reweighted least squares (IRLS) estimation as described in [Maronna et al., 2006]. In combination of the two techniques, we use robust iteratively reweighted DLT (IRDLT) estimation of the 3D position. We apply the error norm $\rho(\cdot)$ proposed by Huber in [Huber, 1964] as a robust estimator,

$$\rho(e) = \begin{cases} \frac{1}{2}e^2 & , \quad |e| < t \\ t|e| - \frac{1}{2}t^2 & , \quad |e| \geq t \end{cases}, \tag{4.34}$$

which yields the weight function

$$w(e) = \frac{1}{e}\frac{\partial \rho(e)}{\partial e} = \begin{cases} 1 & , \quad |e| < t \\ -\frac{t}{e} & , \quad |e| \geq t \wedge e < 0 \\ \frac{t}{e} & , \quad |e| \geq t \wedge e \geq 0 \end{cases} \tag{4.35}$$

for error e and outlier boundary t. The IRDLT estimation algorithm performs the following steps to compute an estimation $\widehat{\mathbf{X}}$ of 3D point $\mathbf{X}$ from image points $\mathbf{x}_i$ and projection matrices $\mathbf{P}_i$ using weights w_i, $i = 0, 1, \ldots, I$.

1. Initialize weights w_i as shown below.
2. Perform 3D reconstruction using the weighted DLT algorithm.
3. Recompute weights w_i according to (4.35).
4. If the changes of w_i are small, then stop; else go to step 2.

These steps endow a costly-inexpensive and robust 3D estimation $\widehat{\mathbf{X}}$ of the world point $\mathbf{X}$ and provide adapted weights w_i representing the amount of support that each feature position gives to the current 3D estimate.

Detecting tracking outliers. Besides a robust estimation of the 3D position, the IRDLT procedure yields weights $w_i \in [0, 1]$. These weights indicate how likely it is for a position $\mathbf{x}_i$ to be an outlier, whereat $w_i = 1$ states that position $\mathbf{x}_i$ in image i perfectly supports the estimated 3D position $\widehat{\mathbf{X}}$. We use the weight w_I corresponding to the last tracked position $\mathbf{x}_I$ to decide for acceptance of the whole tracking step. If $w_I < t_w$, e.g. $t_w = 0.5$, we roll back the whole tracking step of $\text{GKLT}_{3\text{D}}$, i. e. we restore the previous 3D estimation and delete position $\mathbf{x}_I$. In this case the current feature position is reinitialized from the 3D estimate instead of the outlying tracked position for the consecutive frame.

Employing the above steps, the $\text{GKLT}_{3\text{D}}$ framework exploits the benefits of a concurrent, robust 3D estimation by using the interdependencies of the data in the 2D and the 3D domain. In doing so, we create a framework that estimates hidden information, uses this information to improve tracking, updates the hidden information and so on. On a somewhat abstract level, this approach is comparable to the integrated uncertainty estimation of GKLT_2, which itself is used inside the $\text{GKLT}_{3\text{D}}$ framework as outlined in Table 4.1.

4.5 Experimental Evaluation

The previous section depicted ways to extend the standard KLT tracking using prior knowledge of camera parameters. In doing so, we designed different

levels of the Guided KLT tracking. For the sake of a compact notation, we denote the different stages of expansion as follows.

- $\underline{\text{GKLT}_1}$: Guided KLT tracking with manual uncertainty adjustment as in Section 4.4.1.
- $\underline{\text{GKLT}_2}$: Guided KLT tracking with integrated uncertainty estimation as in Section 4.4.2.
- $\underline{\text{GKLT}_{3\text{D}}}$: Guided KLT tracking with integrated uncertainty estimation in combination with a concurrent robust 3D estimation as in Section 4.4.3.

In the following, we describe methodology and results of the experimental evaluation comparing the different GKLT and the standard KLT tracking methods.

In our experiments, we evaluate the *tracking duration* and *accuracy*. We measure the tracking duration, or trail length, as the number of frames in which a feature is seen, which includes the initial frame. For a bunch of features, we note the mean trail length μ_L and the standard deviation σ_L. Unless noted otherwise, the accuracy is given in dual manner by examining the 3D error, i. e. the distance between the reconstructed 3D point and a ground-truth model of the observed object in terms of the mean error μ_E and the according standard deviation σ_E. This kind of distance measure requires a registration step. For this step, we perform an interactive 3D registration of the ground-truth model into the measurement space. First, we manually determine the image positions of many features spread over the whole object in many frames. Theoretically, we only need three features in general position seen in two images. The resulting 3D positions allow to establish 3D correspondences using the respective 3D points of the ground-truth model. Finally, we use unit dual quaternions to estimate an optimal Euclidean transformation in closed form, see [Walker et al., 1991]. Regarding 3D ground-truth, we either use a 3D CAD model or a high-precision 3D reconstruction of the respective object produced by a fringe-projection system as in [Kühmstedt et al., 2007]. Given the standard measurement error of 70μm achieved by this system, the resulting 3D model is valid as ground-truth data for a passive camera setup with VGA resolution. The reason for this kind of accuracy evaluation is given by the fact that accurately tracked features are necessary for an accurate 3D reconstruction. In reverse,

we imply that a more accurate 3D reconstruction is based on a more accurate feature tracking, provided that other variables remain constant. Throughout the experiments, we employ features templates of size 5×5.

In order to give a thorough performance survey of GKLT tracking, we test the method with different scenes and objects, use different kinds of features, and analyze special cases as well as mass statistics.

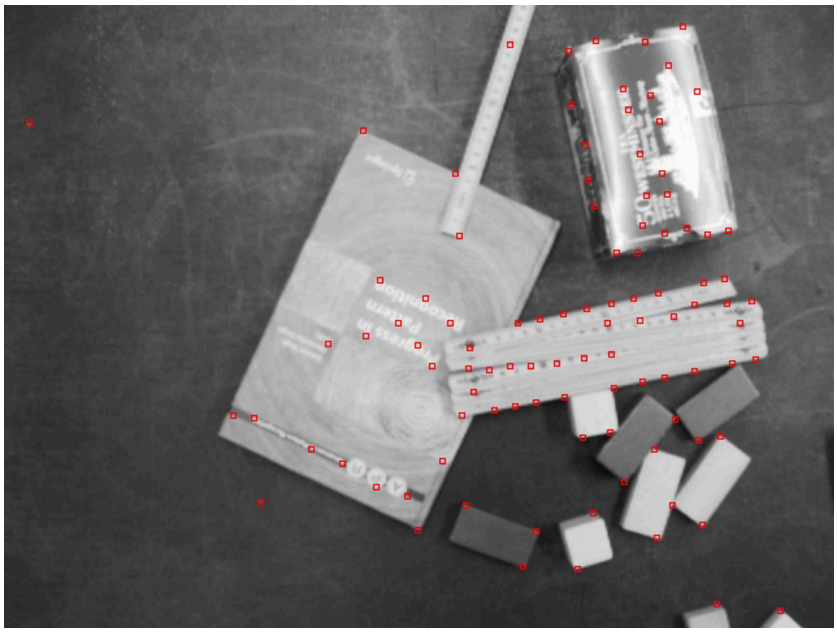


Figure 4.4: Initial frame and 100 features selected for tracking.

4.5.1 Tracking Duration of GKLT$_1$ Using Translational Warping

For the first experiment, we compare the standard KLT and the GKLT$_1$ tracking methods using only translational warping functions as depicted in Appendix A. This test is justified since the main difference between the tracking methods is the modeling of the feature translation. We use a sequence of 21 images, the first one for feature detection and 20 images for tracking. We employ the robot arm shown in Figure 1.1 to move a calibrated, 640 × 480-camera Sony DFW-VL 500 over the scene. Figure 4.4 illustrates the initial frame and 100 features selected for tracking. Figure 4.5 is the ninth frame

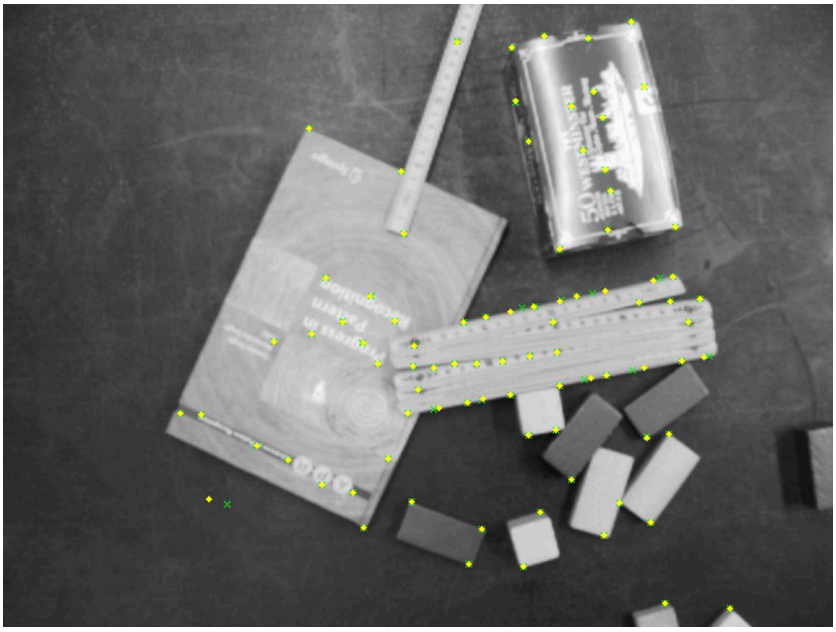


Figure 4.5: Frame 9 of the tracking sequence. Light green crosses mark the feature positions tracked by KLT, yellow diamonds by GKLT_1 using w = 0.9.

Table 4.2: Tracking durations of KLT and GKLT_1 in a sequence of 20 frames for tracking. Further details in Section 4.5.1.

GKLT_1, $w =$	0.5	0.6	0.7	0.8	0.9	0.95	KLT
μ_L (frames)	15.96	16.16	16.18	16.10	16.00	16.04	16.07
σ_L (frames)	5.30	5.19	5.21	5.21	5.27	5.26	5.28

together with the tracked feature positions of the test run with $w = 0.9$ for GKLT_1.

Now, we have a look at the tracking durations achieved by KLT and GKLT_1 with different settings of the epipolar weight w, which is the parameter related to the uncertainty of the known camera parameters. For this setup, we reach the tracking durations listed in Table 4.2. We see only marginal improvements for certain values of the epipolar weight. Nonetheless, we observe differing tracking qualities in Figure 4.5. The relative feature positions, especially along edges, are better preserved using GKLT_1.

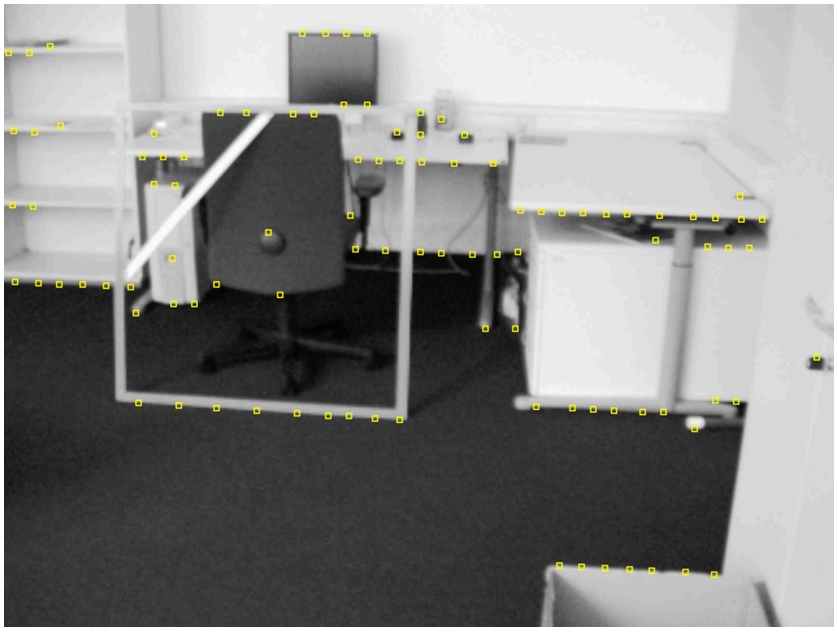


Figure 4.6: First frame and 100 features selected for tracking.

4.5.2 Accuracy of GKLT$_1$ Using Translational Warping

The last test revealed differences between KLT and GKLT$_1$ regarding the preservation of the relative feature positions, which hints at different 2D tracking accuracies. Hence, we now take two images from different positions and calibrate the extrinsic camera parameters using the method presented in [Trummer et al., 2006]. We detect 100 features in the first image of the office scene in Figure 4.6 and manually create the ground-truth positions in the second image, Figure 4.7. As reference for our ground-truth data, we do not use more than the feature positions in the first image. We measure how far (in pixels) the tracked positions deviate from the ground-truth and give the results in Table 4.3.

We observe that GKLT$_1$ produces smaller errors for all allocations of w tested. GKLT$_1$ shows a mean error up to 20% smaller than the KLT error, in which case the standard deviation is also smaller by 20%. The close-up in Figure 4.8 shows that in particular features along edges, i. e. 1D features, are tracked more accurately using additional knowledge about camera parameters in terms of an epipolar constraint. Somewhat surprising is the good performance of GKLT$_1$ at $w = 0.5$. This allocation of w does not favor any

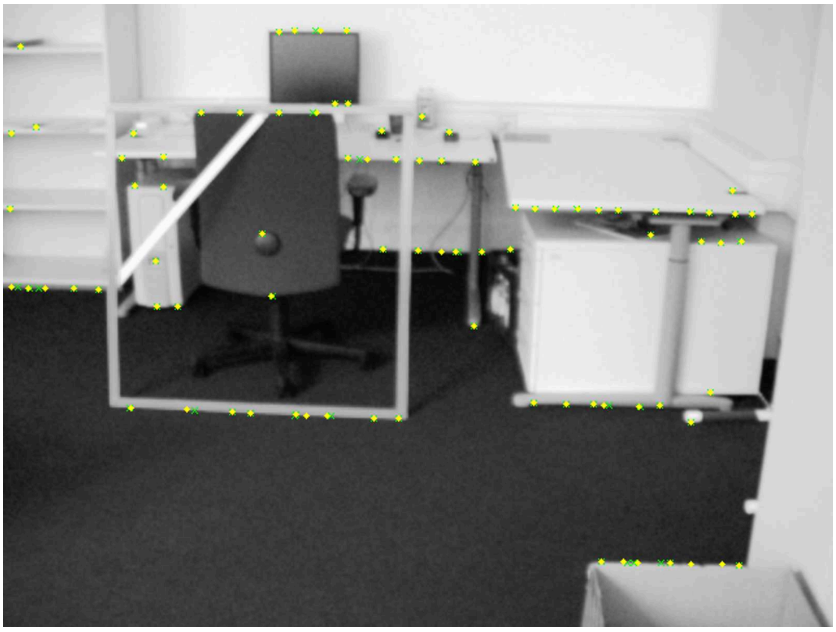
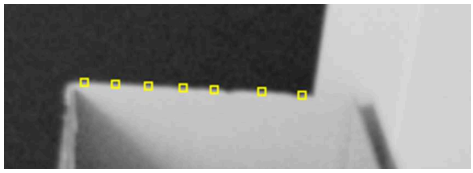


Figure 4.7: Second frame. Light green crosses mark the feature positions tracked by KLT, yellow diamonds by GKLT$_1$ using $w = 0.5$.

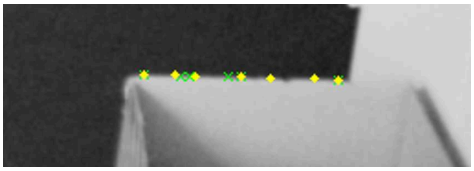
Table 4.3: Tracking accuracies of KLT and GKLT$_1$ in two images. Further details in Section 4.5.2.

GKLT$_1$, $w =$	0.5	0.6	0.7	0.8	0.9	0.95	KLT
μ_E (pixels)	4.78	4.69	4.97	4.89	5.37	5.39	5.84
σ_E (pixels)	5.52	5.67	6.31	6.94	7.23	7.48	7.17

translational direction in the optimization. Only the step size of the parameter change is halved. The better accuracy than KLT for $w = 0.5$ indicates that changing the directions of translation itself is positively affecting the optimization process, if the new directions are related to the true translation. One possible reason for this effect is the approximation of the image gradient using discrete sampling values in a discrete image. An alignment of the sampling positions along the true translation may favor an accurate approximation of the image gradient.



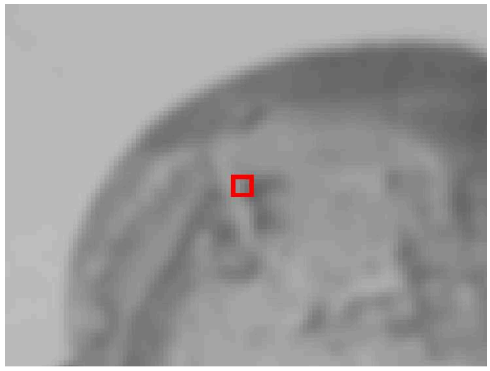
(a) Features initialized along an edge.



(b) Positions tracked by KLT (light green crosses) and $\text{GKLT}_\mathbf{1}$ (yellow diamonds).

Figure 4.8: Close-ups from Figures 4.6 and 4.7. $\text{GKLT}_\mathbf{1}$ tracks all features and preserves point alignment.

(a) Image of Santa Claus figurine **Horschtl** and one feature selected.



(b) Close-up of the selected 2D feature.

Figure 4.9: Case study comparing KLT and $\text{GKLT}_\mathbf{1}$ using one feature.

4.5.3 A Case Study: Comparing Optimization Steps

Apparently, GKLT tracking provides improved tracking results in some cases by employing epipolar feature translation. To understand this effect, we have a closer look at the actual optimization performed for feature tracking. For this, we select a feature as illustrated in Figure 4.9 and try to track it in one frame using KLT and GKLT_1 tracking with pure translational transformation. This setup facilitates an illustration of the respective objective functions and optimization steps in terms of the respective translational parameters. Since we explicitly examine a single case, this experiment, by nature, only supports statements of existence.

Figure 4.10 provides an approximation of the surface of the tracking objective function. The figure further contains blue boxes marking the tracking steps, which lead top-down in the sense of steepest descent with respect to the objective function. Thus, the highest marker, in the sense of the highest level on the objective function, represents the initial state of the transformation. We note a clear minimum of the objective function close to the initial position. The shape of this minimum also indicates a true 2D structure of the tracked feature. Nonetheless, a large baseline between the frames causes the initial position to be much closer to a second close local minimum, which is not the desired one. In fact, the initial position lies inside the basin of convergence of the false local minimum. It is hence not surprising that the KLT tracking descents to the false local optimum and is stuck there without convergence.

In opposite to KLT, GKLT tracking weights the translational directions in a purposive manner. For a proper epipolar weight, optimization steps along the respective epipolar line are larger than steps in the perpendicular direction, as to be seen in Figure 4.11. Changes of the parameter λ_1, which represents translations along the epipolar line, are larger than the ones of λ_2 responsible for the perpendicular direction. The manual allocation $w = 0.9$ corresponds to assumption of a quite accurate epipolar geometry. The combination of the specific image feature, the epipolar geometry, and the proper setting of the epipolar weight lead the optimization out of the wrong basin of convergence and towards the true local optimum for the feature translation. A part of the price to pay for this are the small steps for the final optimization of the epipolar translation parameters. The image data, and hence the objective function, dictates a small deviation from the epipolar line, which is performed in small steps. Thus, we slightly increase the number of steps and computation time. We quantitatively refer to the computation time in the next section. By this experiment, we mainly see that there are cases in which the prior knowledge of camera parameters guides the feature tracking and affords a better tracking result than without using prior knowledge.

4.5.4 GKLT$_1$ and GKLT$_2$ with Translational Warping

The difference between GKLT$_1$ and GKLT$_2$ feature tracking is that GKLT$_2$ utilizes automatic uncertainty estimation as depicted in Section 4.4.2. Now, the question arises how this integrated uncertainty estimation performs in the

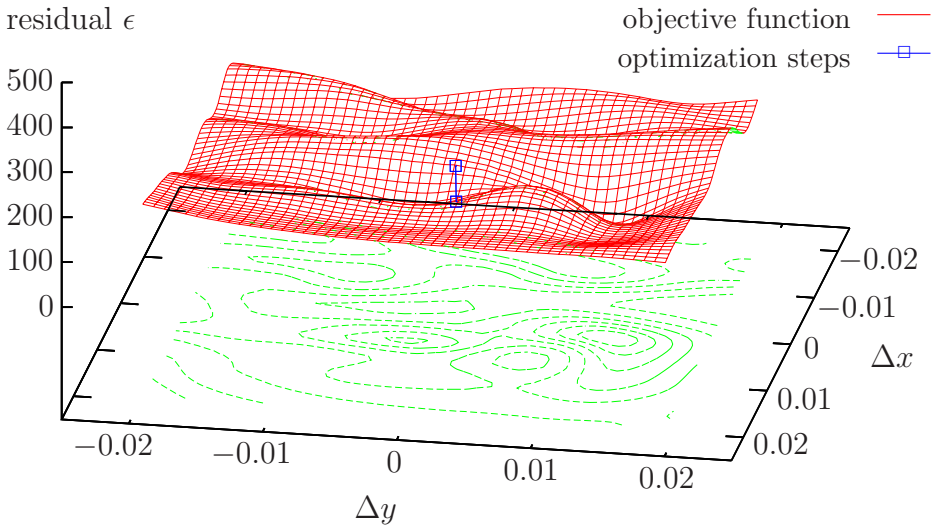


Figure 4.10: KLT objective function and performed optimization steps while tracking the feature shown in Figure 4.9. The initial solution is in the convergence radius of a local, non-global minimum. Optimizing unweightedly along the image axes favors the wrong local minimum, KLT tracking does not reach convergence.

presence of accurate and noisy prior information. Additionally, it would be interesting to compare the performance of GKLT_1 with different allocations of the uncertainty parameter for the same prior information. To this end we utilize the following experiment. We employ pure translational tracking of the 746 features shown in Figure 4.12(a) in 26 frames, where the first frame is reserved for feature detection. The camera path is located on a half sphere centered at the gravity center of *Horschtl*. Each feature tracked in at least five frames is fed to a standard DLT triangulation, which yields the respective 3D point. The resulting 3D points of all features and the registered 3D ground-truth , cf. Figure 4.12(b), provide the basis for an accuracy evaluation of the reconstructed 3D data. As feature tracking methods we compare KLT, GKLT_1 with epipolar weights $w = 0.0, 0.1, \ldots, 1.0$, and GKLT_2 with automatic uncertainty estimation, each of them using only translational warping. We initialize the epipolar weight of GKLT_2 for each feature as $w = 0.5$. Considering the effect of noise prior information, we use two setups. First, we use the camera parameters as given by the controlled environment as described above and, second, we employ completely random camera parameters. In

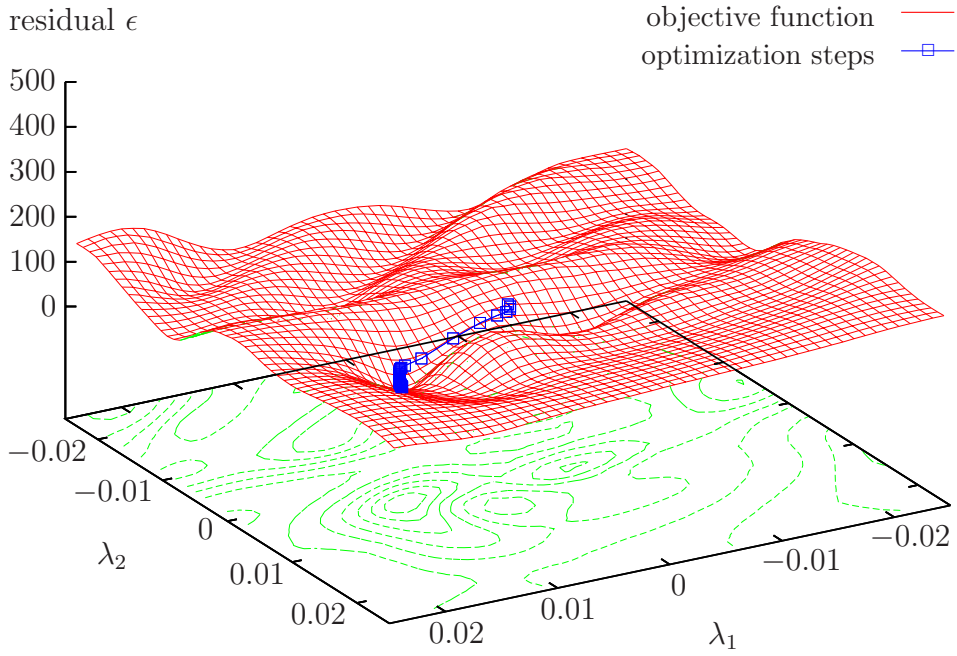
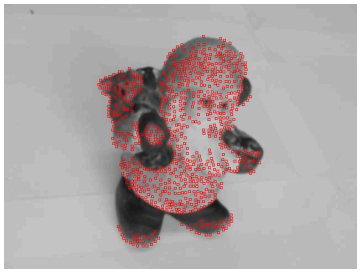


Figure 4.11: GKLT$_1$ (w = 0.9) objective function and optimization steps regarding the feature shown in Figure 4.9. Note that the objective function is flipped with respect to Figure 4.10. GKLT only executes small steps in non-epipolar direction. Large steps along the epipolar line lead the optimization algorithm to the true optimum. GKLT tracking reaches convergence.

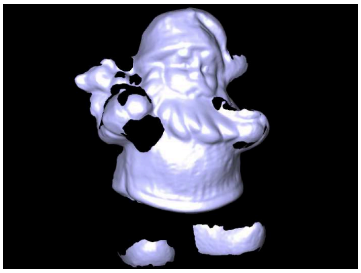
the latter case, the epipolar directions are also random with respect to the truth, which can be interpreted as a maximal amount of additional noise

Table 4.4 lists the achieved tracking accuracies. We additionally provide information about the tracking durations.

With respect to camera parameters without additional noise, the tracking durations of KLT and GKLT$_2$ are quite the same in this experiment. As the main reason for this we see the pure translational warping, which cannot cope with the perspective effects caused by the non-linear camera motion. For GKLT$_1$, we find severe variations of the tracking duration, which compares to KLT for $w = 0.9$, but falls behind up to 50% for other values of w. Likewise, the accuracy of GKLT$_1$ strongly depends on the selected value of w. For $w = 0.9$, GKLT$_1$ produces a mean error that is 9% smaller than the one of KLT, and a standard deviation being 25% smaller. On the other hand, $w = 0.0$ yields a mean error 270% larger than the one of KLT. An optimal value for w is likely to be in $[0.8, 1.0]$, but it is unknown. GKLT$_2$ reduces the



(a) Initial frame of the test sequence with 746 features selected.



(b) View of the set of 3D reference points. The surface mesh is shown for the purpose of illustration.

Figure 4.12: Test image and 3D reference data of the Horschtl figurine.

mean error by 5% and the standard deviation by 9% compared to KLT. This result of $GKLT_2$ is close to the best result of $GKLT_1$ and far better than the worst results of $GKLT_1$ with manual uncertainty control.

Regarding random camera parameters, $GKLT_2$ again shows a tracking duration comparable to KLT, and slightly better accuracy. $GKLT_2$ automatically adapts the uncertainty parameter of each feature individually and hence manages the useless prior information. On the contrary, $GKLT_1$ uses a global value of w for all features in all frames. Again, the performance of $GKLT_1$ is strongly fluctuating for different allocations of w. $GKLT_1$ reaches the performance level of KLT at $w = 0.2$, but gives dramatically worse results for any other value of w.

This experiment gives evidence that GKLT performs better than the standard KLT tracking provided a proper allocation of the uncertainty parameter w. $GKLT_1$ uses manual uncertainty control and is hence depending on additional prior information about the quality of the known camera parameters. $GKLT_2$ utilizes integrated uncertainty estimation and performs better than standard KLT in any case tested, but does not reach the results of $GKLT_1$ with an optimal choice of w. The quality differences of KLT and GKLT are already observable for pure translational warping and a quite short tracking sequence with 25 frames. Further tests have to reveal the performance of GKLT for longer image sequences and the most often used affine warping.

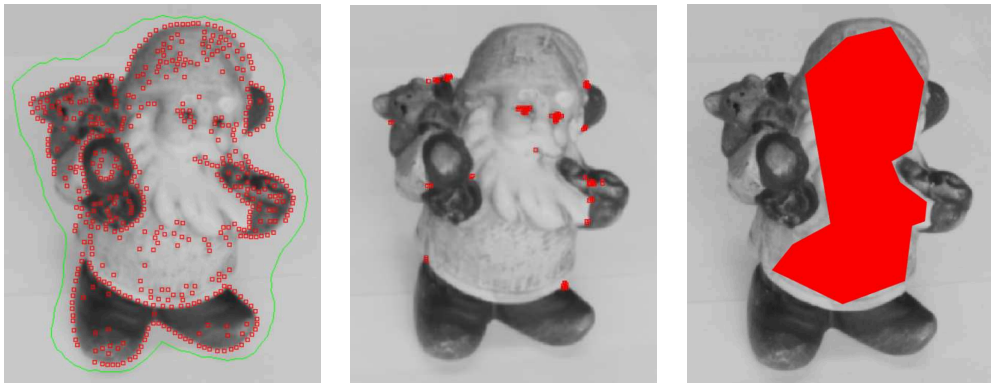
Table 4.4: Accuracy evaluation by mean error distance $\mu_E(\mathrm{mm})$ and standard deviation $\sigma_E(\mathrm{mm})$ for each tracker. $\mathbf{GKLT_1}$ shows accuracy from 9% better to 269% worse than KLT, depending on choice of w relative to respective uncertainty of camera parameters. $\mathbf{GKLT_2}$ performs slightly better than standard KLT in any case tested. Without additional noise, the accuracy of $\mathbf{GKLT_2}$ is 5% better than KLT.

	KLT	**GKLT$_1$**, w equals:											**GKLT$_2$**
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
Using camera parameters without additional noise:													
$\mu_E(\mathrm{mm})$	2.68	9.90	3.52	3.15	2.93	2.77	2.77	2.65	2.62	2.51	2.45	3.90	2.56
$\sigma_E(\mathrm{mm})$	3.70	6.99	4.65	4.08	3.63	3.38	3.63	3.55	3.41	3.17	2.77	5.12	3.36
Using disturbed camera parameters:													
$\mu_E(\mathrm{mm})$	2.68	5.09	2.76	2.68	2.75	2.76	2.77	2.78	2.88	3.05	3.35	7.98	2.66
$\sigma_E(\mathrm{mm})$	3.70	5.60	3.40	3.37	3.60	3.71	3.63	3.50	4.05	4.08	4.30	6.90	3.61

4.5.5 GKLT_2 Using Affine Warping

The previous experiments highlight that translational feature warping benefits from switching to epipolar directions regarding uncertainty. We now want to examine the behavior of GKLT_2 with affine feature warping as formulated in Section A.2. Still, the modifications of GKLT concern only the translational part of the feature warping. It is hence interesting to see how this modification affects the optimization of all warping parameters. Besides, affine warping is a common choice for the feature transformation and by this is an important benchmark for GKLT tracking. From now, we omit the evaluation of GKLT_1 for the sake of compactness. By means of the previous experiment, we have an impression of the relation between GKLT_2 and GKLT_1 with different values of w. Since, in general, the uncertainty of the prior knowledge is unknown, we stick with GKLT_2 that provides automatic uncertainty estimation.

For this test, we use 51 images of *Horschtl* taken from a half sphere above the figurine, and we use the registered 3D ground-truth shown in Figure 4.12. Initiated by the results of previous experiments, we explicitly separate the following tests with respect to different feature types, in particular we separately deal with 1D features along edges, with features providing a true 2D optical structure, and with exhaustively chosen features, cf. Figure 4.13. Additionally, we vary the baseline between consecutive frames by just leaving out every second frame.



(a) 499 1D features, i. e. edge features.

(b) 100 2D features, i. e. corner-like features.

(c) 28018 features, exhaustively chosen.

Figure 4.13: Initial frame of a test sequence. The features used for tracking are separated in terms of the optical structure.

The Tables 4.5 and 4.6 present the results for tracking 1D features with different baselines, while the Tables 4.7 and 4.8 list the respective numbers for tracking 2D features. The results for features that are chosen exhaustively in an image region appear in the Tables 4.9 and 4.10. In general, $GKLT_2$ provides a more accurate tracking and better tracking durations than KLT. The advantages of $GKLT_2$ are especially prominent for the cases of increased baselines and deficient optical feature structures. These results highlight that $GKLT_2$ makes effective use of the epipolar translation constraints to resolve ambiguous motion due to the aperture problem. $GKLT_2$ improves the mean tracking duration by up to 13% and the mean tracking accuracy by up to 41% compared to KLT.

4.5.6 Comparing KLT, $GKLT_2$, and $GKLT_{3D}$

$GKLT_{3D}$ augments the GKLT tracking framework by continuously and robustly estimating the 3D position using tracked feature positions in the images. Thus, $GKLT_{3D}$ infers information on the 3D state and, in reverse, employs this information to detect tracking outliers and to reinitialize lost features in the images. Since $GKLT_2$ already uses prior knowledge of camera parameters to improve tracking, which has been shown by the previous experiments, we implement $GKLT_2$ as the actual image tracking within the

Table 4.5: Tracking the 499 1D features shown in Figure 4.13(a) in 50 frames. GKLT$_2$ provides better tracking duration and better accuracy compared to KLT. The mean trail length is about 6% larger and the mean error about 41% smaller for GKLT$_2$.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	9.30	7.25	11.34	12.26
GKLT$_2$	9.88 +6.22%	8.14 +12.24%	6.68 −41.09%	6.53 −46.74%

Table 4.6: Skipping every second frame, we track 499 1D features in 25 frames, i. e. we increase the baseline between consecutive frames. Again, GKLT$_2$ provides better tracking duration and better accuracy compared to KLT.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	5.23	4.21	6.58	7.89
GKLT$_2$	5.56 +6.40%	4.56 +8.30%	5.98 −9.12%	11.35 +43.85%

Table 4.7: Tracking the 100 2D features shown in Figure 4.13(b) in 50 frames. GKLT$_2$ provides slightly better tracking duration and better accuracy compared to KLT.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	18.64	11.70	2.20	1.72
GKLT$_2$	19.06 +2.25%	11.84 +1.19%	2.19 −0.45%	1.55 −9.88%

Table 4.8: The increased baseline also increases the performance gain: GKLT$_2$ handles the larger baseline clearly better than KLT in terms of tracking duration and accuracy.

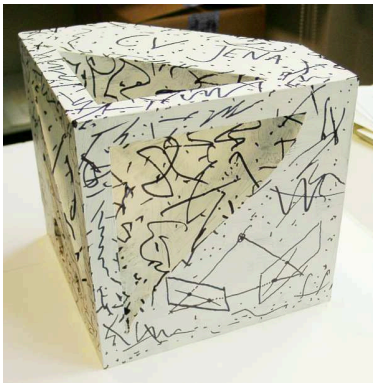
	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	9.18	5.42	2.56	1.82
GKLT$_2$	10.39 +13.18%	6.61 +21.83%	1.91 −25.39%	1.60 −12.09%

Table 4.9: Tracking the 28018 features shown in Figure 4.13(c) in 50 frames. GKLT$_2$ provides slightly better tracking duration and clearly better accuracy compared to KLT.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	19.72	14.33	1.91	5.90
GKLT$_2$	20.12 +2.04%	14.42 +0.61%	1.48 −22.51%	3.11 −47.29%

Table 4.10: Tracking 28018 features in 25 frames with increased baseline. The advantage of GKLT$_2$ over KLT is comparable to the case of smaller baseline.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	11.47	7.79	2.21	6.93
GKLT$_2$	11.69 +1.94%	7.86 +0.92%	1.84 −16.74%	4.78 −31.02%



(a) View of the NBV test object

(b) ... inside a controlled environment.

Figure 4.14: All-aluminum NBV test object proposed in [Munkelt et al., 2007]. An outstanding artistic design provides optical surface structure and hence features for tracking.

$GKLT_{3D}$ framework, cf. Table 4.1. In order to find the effects of these further extensions, we take images of the NBV test object proposed in [Munkelt et al., 2007] utilizing camera positions on a half sphere over the object. Figure 4.15 demonstrates the initial frame of the tracking sequences and the features selected for tracking. Since GKLT is capable of tracking corner-like and edge-like features, these selected features include both types. For accuracy evaluation, we register the 3D CAD model of the object to the measurement space.

First, we examine the performances using a short image sequence with ten frames for tracking. That is why we do not expect great differences regarding the tracking duration. And since a short sequence like this covers only a small overall baseline, this experiment mainly focuses on the tracking accuracy. Table 4.11 presents the tracking results for the short sequence. The mean trail lengths of $GKLT_2$ and $GKLT_{3D}$ are very similar and about 11% larger than the one of KLT. Despite the low number of frames, $GKLT_2$ reduces the mean error by about 55% and the according standard deviation by 75% compared to KLT. $GKLT_{3D}$ even exceeds this performance gain by reaching a mean error 64% better and a standard deviation 89% better than KLT.

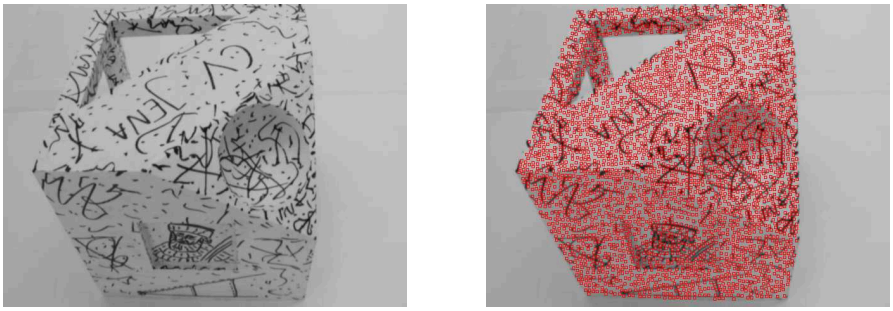
As the second test, we elongate the tracking sequence to 200 frames. The

Table 4.11: Comparison of tracking duration and accuracy for tracking the features from Figure 4.15(b) in a short sequence of 11 frames, one frame for feature detection. GKLT$_{3D}$ offers best accuracy.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	9.56	2.60	7.62	25.27
$GKLT_2$	10.67 +11.61%	1.29 −50.38%	3.46 −54.59%	6.30 −75.07%
$GKLT_{3D}$	10.64 +11.30%	1.36 −47.70%	2.75 −63.91%	2.74 −89.16%

Table 4.12: Comparison in a long sequence of 201 frames, one frame for feature detection. GKLT$_{3D}$ shows, by far, the best tracking duration and reconstruction accuracy.

	μ_L (frames)	σ_L (frames)	μ_E (mm)	σ_E (mm)
KLT	23.47	21.22	9.10	27.26
$GKLT_2$	33.88 +44.35%	21.70 +2.36%	4.34 −52.31%	6.69 −75.46%
$GKLT_{3D}$	91.06 +287.98%	41.90 +97.46%	2.65 −70.88%	2.38 −91.27%



(a) Initial frame.

(b) 3227 features (red boxes) selected.

Figure 4.15: Initial frame and selected features.

first 11 frames of this long sequence are identical to the short sequence used above, so the initialization of Figure 4.15 still holds. This experiment gives particular attention to the tracking stability, which includes duration and accuracy in the following manner. We term tracking unstable if the trail lengths are short, which causes a small baseline and hence an unstable 3D reconstruction. Likewise, tracking is unstable for large trail lengths and low tracking accuracy. In this case, a tracked image position does not provide a good estimation of the true image position of the mapped 3D point, and the 3D reconstruction will also be unstable. Thus, tracking is more stable for the combination of larger duration and higher accuracy. Table 4.12 underlines that GKLT_2 clearly improves the tracking results compared to KLT; the mean tracking duration is 44% larger and the mean error 52% smaller. Nonetheless, considering the results for the short sequence in Table 4.11, the relative accuracy improvement falls behind in this test. Furthermore, the absolute mean accuracy achieved by GKLT_2 is worse for the long sequence, which raises stability issues. On the contrary, $\text{GKLT}_{3\text{D}}$ clearly outperforms the other methods in each sense tested. Likewise, $\text{GKLT}_{3\text{D}}$ together with the long image sequence improves its own absolute results reached with the short sequence. The reinitialization of lost features done by $\text{GKLT}_{3\text{D}}$ dramatically increases the mean trail length. Robustly estimating the 3D position and detecting tracking outliers, at the same time, improves tracking accuracy. Thus, $\text{GKLT}_{3\text{D}}$ offers the most accurate and the most stable tracking and 3D reconstruction of the methods tested.

4.6 Conclusion and Future Work

In this chapter, we present an extended framework for feature tracking based on the KLT tracking method, the Guided KLT tracking. We do so by redefining the translational part of the warping function using prior knowledge of the epipolar geometry. These new directions of translation correspond to a movement following the epipolar constraint and to another one caused by uncertainty, which allows to softly constrain the optimization search space with respect to the present uncertainty. We further present a way to automatically estimate the amount of the generally unknown uncertainty. In the end, we establish a framework augmented by concurrent, robust 3D estimation, reinitialization of lost features, and detection of tracking outliers.

As an aspect of interpretation, we augment the translational directions with meaning, which theoretically reduces the optimization space by one degree of freedom thanks to prior knowledge of camera parameters. Instead of optimizing along the image axes, which are arbitrary directions with respect to a general feature translation, we want to apply the theoretically true translation along the respective epipolar line. Since we explicitly regard uncertainty, this constraining of the search space has to be adapted to the present uncertainty. This approach can be seen as an application of the epipolar band described in [Faugeras and Luong, 2001] page 349. In opposite to known methods describing the uncertainty of the epipolar geometry, we estimate the uncertainty using the observed data directly. Using an optimization scheme and the epipolar constraint, GKLT tracking combines the advantages of feature tracking and stereo matching.

The effects of using GKLT tracking are presented by various experiments. Besides outperforming the known KLT tracking method in terms of tracking duration and accuracy, GKLT also eases the tracking assumptions of small baselines and true 2D optical structure.

Future work meets further challenges to improve GKLT tracking. For instance, we aim at a dilution of the brightness-constancy assumption, which can be done by using a feature descriptor consisting of data other than intensity values, or by employing distance measures that are invariant to certain illumination changes. Another point of interest is facilitating GKLT tracking *without* prior knowledge of camera parameters. For this, the main idea is a mutual optimization of tracking results and estimation of camera parameters. A possible first step consists of feature tracking with standard KLT

in an image sequence and estimating the epipolar geometries between images. Second, GKLT tracking uses the estimated epipolar geometries and improves the tracking results, which in turn allows to better estimate the epipolar geometries, and so on. By this means we could reach a concurrent feature tracking, 3D reconstruction, and self-calibration using image streams from uncalibrated cameras.

Chapter 5

Coarse Registration of 3D Surface Triangulations

From the previous chapter we see a way to do feature tracking using knowledge of camera parameters, which directly leads to accuracy-based view planning as being exposed in Chapter 6. For this chapter, we take an indirect path towards view planning. We describe a novel method for 3D coarse registration using surface triangulations. In Section 5.1 we enlighten this indirect path by pointing out the method's general applications and their meaning for view planning. Section 5.2 outlines challenges of the problem formulation and our way to solve them. Previous work is reviewed in Section 5.3. We present the actual registration method in Section 5.4. Section 5.5 evaluates the performance using surface triangulations of different kinds of natural and technical objects. Finally, we conclude this chapter providing related topics for future research in Section 5.6.

5.1 Applications and Relevance to View Planning

Solving the registration problem for 3D objects is a crucial task in many 3D applications. One important issue in this context is estimating optimal transformations between sets of 3D points, i. e. aligning the two sets. The alignment is necessary, for instance, to fuse partial reconstructions of one object or to compute an error measure with respect to a ground-truth model in

another coordinate frame. As another example, the developing research with photonic mixing devices (PMD), like in [Kähler et al., 2008], requires registration of range data. In the example of aligning two partial reconstructions, the fact of partial overlap of the input point sets is an additional challenge to the registration method. Considering the alignment of a measurement and the ground-truth model, the registration method used has to cope with different point densities and distributions. As the standard solution to the 3D registration problem, the ICP algorithm proposed in [Besl and McKay, 1992] produces results depending strongly on a proper initialization, cf. [Zinßer et al., 2003, Rusinkiewicz and Levoy, 2001], and it is not explicitly designed to handle input sets with only partial overlap. A further possible application is object identification. Given a database of 3D objects, we align a candidate object to each object in the database. We yield the identification by comparison of the respective alignment errors. In similar ways, the applications of 3D object retrieval and classification, data fusion for 3D reconstruction, and calibration tasks may incorporate the proposed method.

With respect to view planning, the method presented in this chapter is a valuable basis for including specific prior knowledge into the planning procedure in the following sense. Imagine we know certain geometric properties of the target object. This prior knowledge may be given, for instance, as a 3D CAD model or by a parametric definition of a certain geometric primitive. A view planning goal can be to perform the 3D reconstruction of the respective primitive as accurate as possible, which involves the identification of the primitive within the currently reconstructed data. Another aspect is offline training for view planning. Given that a certain geometric primitive is part of many objects to be reconstructed, we may find it beneficial to outsource the view planning for this primitive to a separate offline step. At runtime, the planning method tries to identify the geometric primitive and then just triggers the view plan achieved during the offline training. In this way, the prior knowledge of geometric primitives improves the runtime and the reconstruction result.

5.2 Challenges and Selected Approach

The main idea of our approach for 3D registration, see [Trummer et al., 2009d], is based on the method for 2D affine point matching presented in [Voß et al., 2001]. Given two discrete 2D point sets and provided that these two

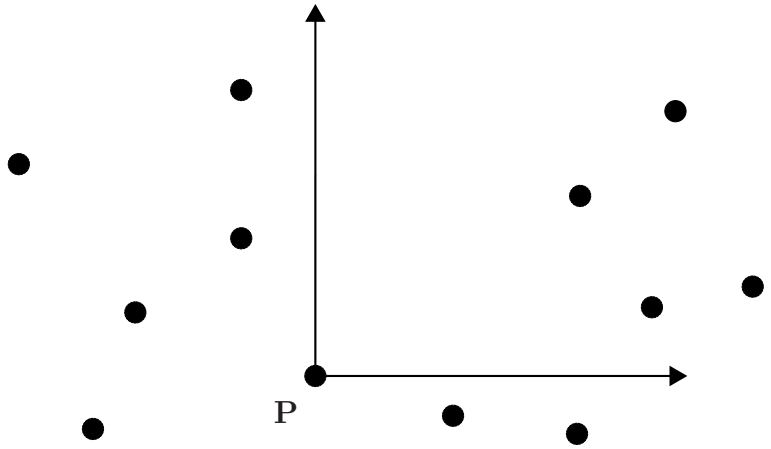
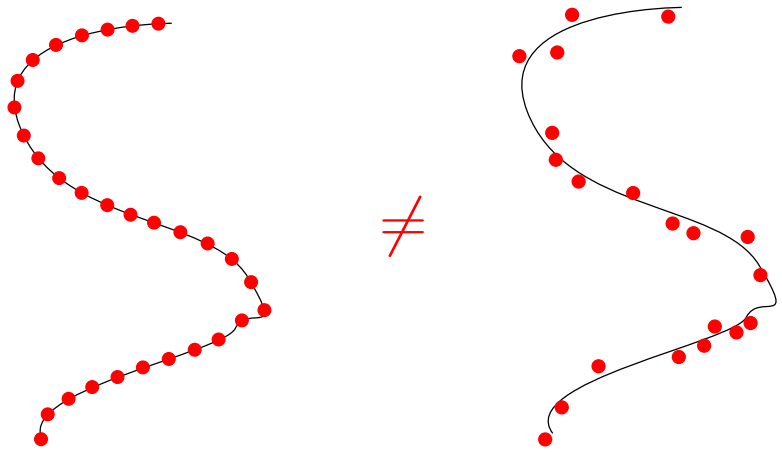


Figure 5.1: In [Voß et al., 2001], the authors present 2D affine point matching using invariants of non-centralized point moments. That is to say, for point P the non-centralized moments of the whole discrete point set endow moment invariants that can be seen as a descriptor for point P in this particular point set.

sets are related by an affine transformation, this work deals with the task to optimally estimate the unknown affine transformation. To this end, Voß and Süße compute non-centralized moments of each point set with respect to each point in it as illustrated in Figure 5.1. Based on these moments, affine moment invariants provide features describing the position of the particular point in the particular point set. Finally, the authors use the Hungarian method, cf. [Kuhn, 1955], to find an optimal assignment of point descriptors, which yields point correspondences that allow to estimate the affine transformation between the two point sets.

We aim at transferring the depicted idea from 2D affine matching of point sets to 3D registration. In doing so, we have to face further difficulties. Different 3D point clouds, for example, different 3D reconstructions of the same real-world surface $\mathfrak{S}$ are not the same point sets, in general. Hence, the preconditions of any matching algorithm assuming transformations of *one* point set are violated. The 3D point sets may feature different point numbers, densities, distributions as shown in Figure 5.2. These characteristics are caused by different discrete representations of the same, non-discrete surface $\mathfrak{S}$. In the end, we try to reasonably align completely different 3D point sets. Computing point moments cannot solve this problem. Another serious



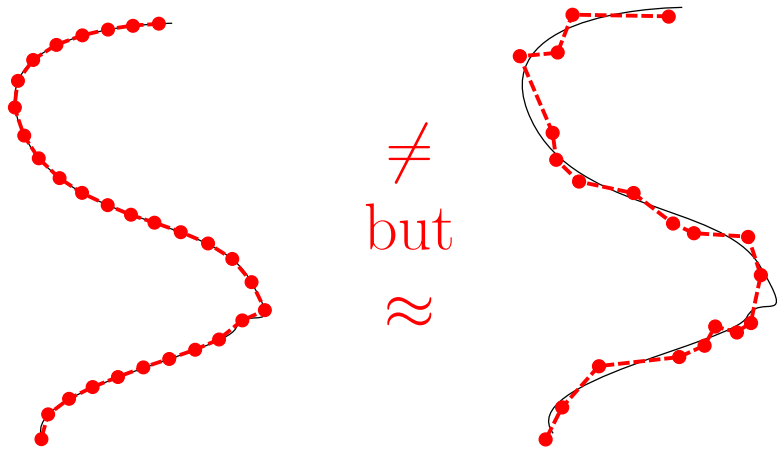
(a) Points derived from a ground-truth model of the surface.

(b) Measured points of the surface.

Figure 5.2: Different sets of points (red) derived from the same real-world surface $\mathfrak{S}$ (black). The point sets are not equal, they differ in terms of cardinality, distribution, density, and accuracy with respect to $\mathfrak{S}$.

challenge is real partial overlap of the two surface regions of $\mathfrak{S}$ represented by the two point sets. The possibility of partial overlap prohibits a pure global alignment and requires to include the principle of locality in some way. As a simple example, the standard solution for the ICP algorithm to reach some robustness against partial overlap is to set a threshold for the next-neighbor search in step 2, Section 3.5.3.1. Thus, the algorithm just ignores points that are too far away from the other point set, which could be beneficial depending on the actual point sets and the currently estimated transformation. We further mention that the result produced by ICP strongly depends on the given initial solution.

An important part of our solution is including non-discrete surface information, see Figure 5.3. Instead of using sets of 3D points, we compute moments of 3D surface triangulations. Surface triangulations are sets of tuples $\mathsf{S} = \{(\mathsf{X}, \mathsf{T})\}$ consisting of a set X of 3D points and a set T of triangles defined by 3-subsets of the point index set according to (1.16). This piecewise-linear surface representation endows invariance with respect to different point numbers, densities, and distributions given that the questionable surface triangulations exactly represent the same surface. This consideration



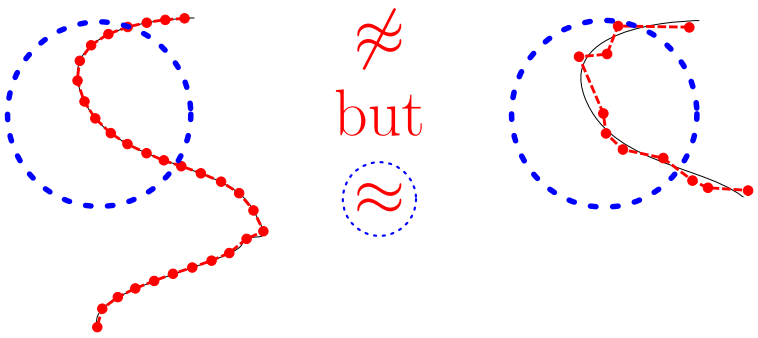
(a) Points derived from a ground-truth model of the surface, additional non-discrete information.

(b) Measured points of the surface, additional non-discrete information.

Figure 5.3: Including non-discrete surface information (red dashed line segments) eases the strict inequality shown in Figure 5.2. We use piecewise-linear approximations of the real-world surface. For the particular case of 3D registration, we employ 3D surface triangulations.

requires to explicitly separate the discrete 3D points in X, the corner points or *vertices*, from general surface points that are elements of any triangle of the surface triangulation; vertices are also general surface points, but not vice versa. For instance, different, accurate triangulations inside a plane surface region probably provide different vertices, but the triangles are all located in the same plane yielding identical surface points. This example easily illustrates the mentioned invariances given surface triangulations that exactly represent a real-world surface $\mathfrak{S}$. Additionally, surface triangulations provide well-conditioned behavior in the sense that small position errors of the vertices cause small changes of the overall surface structure. So, we tackle the problems of different vertices in terms of numbers, local densities, distributions, and accuracies by means of including surface triangulations. This additional input to our registration method may be part of the 3D reconstruction or originate from Delaunay triangulation, cf. [Delaunay, 1934], implemented in `Meshlab` and `Matlab`, for example.

We already mentioned that possible partial overlap requires a non-global



(a) Real-world surface $\mathfrak{S}$ (black) and surface triangulation S (red).

(b) Partial surface $\mathfrak{S}' \subset \mathfrak{S}$ and according surface triangulation S'.

Figure 5.4: Since $\mathfrak{S}' \neq \mathfrak{S}$, neither the surfaces nor the surface triangulations are equal or similar, globally. Yet, regarding local regions (blue) yields local similarity.

treatment in some way. This is a clear fact since two surface representations with only partial overlap are simply not the same, globally. Figure 5.4 displays a real-world surface $\mathfrak{S}$ in Figure 5.4(a) and a partial surface $\mathfrak{S}' \subset \mathfrak{S}$ in Figure 5.4(b) as black lines. The red points and dashed lines indicate the respective surface triangulations. As $\mathfrak{S}' \neq \mathfrak{S}$, the according surface triangulations S' and S are also not equal. Our solution to this particular challenge is to regard local regions, which are shown as blue dashed circles in Figure 5.4. The surfaces $\mathfrak{S}$ and $\mathfrak{S}'$ are equal with respect to the shown local regions. Hence, the above statements of identity and similarity of the respective surface triangulations apply to these local regions.

In comprehension of our approach, we compute local, invariant features of 3D surface triangulations based on non-centralized surface moments for each vertex of each surface triangulation. Without sampling or projection, we derive the features directly from the surface in closed form. We use the Hungarian method for a globally optimal assignment of the resulting descriptors. This yields point correspondences that allow to optimally estimate a 3D transformation, or alignment, of the input surface triangulations. The method is non-iterative and does not require any initial solution. Our ex-

periments with real data show that the proposed method can serve as an automatic initialization of the ICP algorithm and, hence, extends the field of applications for this standard registration method.

We note that the proposed method is also valid for Euclidean transformations of 3D point sets and 3D volumes. So, for instance, given that two 3D point sets are related by a Euclidean transformation, we are able to compute moment invariants based on the 3D point moments and apply the described alignment procedure.

The next section provides a survey of relevant literature.

5.3 Previous Work

Deriving algebraic moment invariants is originally the work of Hu in [Hu, 1962]. In the following years, Flusser increases the set of known moment invariants up to arbitrary orders, [Flusser, 2000]. Providing invariant features of points, lines and other shapes and objects, moment invariants find broad application to solve classification, identification and matching tasks, cf. [Paquet et al., 2000]. For a survey of the development and the application of moment invariants see [Flusser, 2006]. Voss and Suesse [Voß et al., 2001] use invariants of 2D point moments to perform matching of affinely transformed 2D point sets. They establish affine invariants based on non-centralized point moments. By this means and an optimal point assignment using the Hungarian method proposed in [Kuhn, 1955] they present a way to perform matching of a 2D point set and an affine transformation of that same point set. Lo and Don [Lo and Don, 1989] algebraically derive a set of 3D volume moment invariants with respect to Euclidean and similarity transformations. The application of 3D moment invariants with parametric surfaces is shown by Xu and Li in [Xu and Li, 2006], but in this work the authors use sampling of surfaces instead of calculating the exact surface moments. While moments are extensively used for 2D and 3D classification and recognition tasks, cf. [Flusser, 2006], the application of moment invariants to the field of 3D surface registration is only little explored.

The ICP algorithm is established by Besl and McKay in [Besl and McKay, 1992]. Lots of the extensions to this now standard 3D registration method are compared in [Rusinkiewicz and Levoy, 2001]. The application of moment invariants from 3D points to the ICP algorithm is shown in [Sharp et al., 2002]. Compared to our work, in [Sharp et al., 2002] the usage of moment

invariants is incorporated directly into the iterative estimation procedure of ICP, which still needs an initial solution.

Finding suitable ways of computing a coarse registration of 3D data sets is an actual research topic, cf. [Xiao et al., 2007], which is formerly reviewed in [Audette et al., 2000]. Some methods aim at assigning features from salient points or surface regions, for instance, [Makadia et al., 2006,Wyngaerd and van Gool, 2002, Schoen and Haeusler, 2006], whereat the features are constructed from geometrical, [Makadia et al., 2006,Wyngaerd and van Gool, 2002], or information-theoretic, [Schoen and Haeusler, 2006], considerations. Other methods concentrate on effective strategies for searching the whole 3D input data, for example, [Chen et al., 1999,Xiao et al., 2007,Winkelbach et al., 2004], applying iterative RANSAC matching schemes, [Chen et al., 1999, Winkelbach et al., 2004], or by using a volumetric data representation as in [Xiao et al., 2007]. The registration method in [Mian et al., 2006] only applies to range data. Compared to these methods, we use a non-iterative scheme to optimally assign invariant features that we calculate directly from the 3D surface without local fitting, matching, or projection. We compute exact 3D surface moments allowing the application of algebraically derived moment invariants, cf. [Lo and Don, 1989], as invariant features. The application of the denoted invariants of volume moments is valid for points and surfaces regarding Euclidean transformations.

(a) Surface triangulation S_1.

(b) Surface triangulation S_2.

(c) Real-world surface $\mathfrak{S}$.

Figure 5.5: The task is to find a Euclidean transformation that aligns S_1 and S_2 optimally in terms of representing the real-world surface $\mathfrak{S}$ of Dino Detlef.

5.4 3D Coarse Registration

In the following we derive a new, direct method to register general 3D surface triangulations $\mathsf{S}_1 = \{(\mathsf{X}_1, \mathsf{T}_1)\}$ and $\mathsf{S}_2 = \{(\mathsf{X}_2, \mathsf{T}_2)\}$ that represent at least partially overlapping regions of the same real-world surface $\mathfrak{S}$, and that are related by an unknown Euclidean or similarity transformation. At the current state of the development, we focus on invariance with respect to Euclidean transformations. Figure 5.5 illustrates a practical example of the task at hand. Our solution uses a direct and efficient way to compute moments of surface triangulations that allow to establish algebraically derived moment invariants, which serve as features. Since any feature-based approach has to respect different feature qualities, we describe our current statistical solution to filter out non-distinctive features. Finally, we use the Hungarian method to minimize the global costs of assigning features of vertices and, thus, the points itself. These point correspondences endow the data basis to estimate the underlying 3D transformation.

5.4.1 Computing Local, Invariant Features of 3D Surface Triangulations

As the first step of our approach, we determine exact moments of 3D surface triangulations.

Computing surface moments of 3D surface triangulations. We consider a 3D surface triangulation $\mathsf{S} = \{(\mathsf{X}, \mathsf{T})\}$ consisting of a set X of P vertices and a set T of N triangles. A triangle $\mathsf{T}_j \in \mathsf{T}$ is defined by three indices of vertices,

$$\mathsf{T}_j = \{i_{j1}, i_{j2}, i_{j3}\} \tag{5.1}$$

with $1 \leq j \leq N$, $1 \leq i_{jc} \leq P$, and $1 \leq c \leq 3$. Thus, the three vertices of triangle j are $\mathbf{X}_{i_{jc}} = (X_{i_{jc}}, Y_{i_{jc}}, Z_{i_{jc}})^\mathsf{T}$, $1 \leq c \leq 3$. The $(k+l+n)^{\text{th}}$-order 3D surface moments M_{kln} of S are the accumulated surface moments m^j_{kln} of the associated triangles T_j,

$$M_{kln} = \sum_{j=1}^{N} m^j_{kln}. \tag{5.2}$$

For a general triangle T, the surface moments are

$$m_{kln} = \iint_{\mathsf{T}} X^k Y^l Z^n \rho(X, Y, Z)\, \mathrm{d}s, \tag{5.3}$$

where we employ a surface density function $\rho(X, Y, Z) \equiv 1$. Using a suitable parameterization

$$P_{\mathsf{T}}(u, v) = \begin{pmatrix} X_{\mathsf{T}}(u, v) \\ Y_{\mathsf{T}}(u, v) \\ Z_{\mathsf{T}}(u, v) \end{pmatrix}, \tag{5.4}$$

$u, v \in \mathsf{D} \subset \mathbb{R}^2$, of triangle T, the surface moments of the 3D triangle appear as

$$m_{kln} = \iint_{\mathsf{D}} X_{\mathsf{T}}^k(u, v) Y_{\mathsf{T}}^l(u, v) Z_{\mathsf{T}}^n(u, v) \sqrt{EG - F^2}\, \mathrm{d}u\, \mathrm{d}v \tag{5.5}$$

with the coefficients of the first fundamental form $E = X_u^2 + Y_u^2 + Z_u^2$, $G = X_v^2 + Y_v^2 + Z_v^2$, and $F = X_u X_v + Y_u Y_v + Z_u Z_v$, where $X_u = \frac{\partial X_{\mathsf{T}}(u,v)}{\partial u}$ and so on. The set D is the domain of the parameters u and v. The expression in (5.5) was already used in [Xu and Li, 2006] to uniformly sample the 3D triangles. We specify (5.5) for exact computation of the triangle area moments. Applying the parameterization $P_{\mathsf{T}}(u, v)$, we reduce the computation of the surface moments of a triangle T to the computation of the area moments m'_{pq} of D, the definition domain of u, v. We will show these area moments m'_{pq} to be constant for all $P_{\mathsf{T}_j}(u, v)$. In the first step we find a proper parameterization for the general triangle T. The second step shows that the computation of the surface moments of the parameterized triangle can be put down to the computation of the constant area moments of D.

First, we describe the triangle T as a simplex $\mathsf{Q} \subset \mathbb{R}^3$,

$$\mathsf{Q} = \alpha \mathbf{X}_1 + \beta \mathbf{X}_2 + \gamma \mathbf{X}_3 \quad \text{with} \quad \alpha, \beta, \gamma \geq 0 \quad \text{and} \quad \alpha + \beta + \gamma = 1, \tag{5.6}$$

where $\mathbf{X}_c = (X_c, Y_c, Z_c)^{\mathsf{T}}$, $1 \leq c \leq 3$, are the vertices of triangle T. It follows that $\gamma = 1 - \alpha - \beta$. By rearranging we yield

$$\mathsf{Q} = \alpha(\mathbf{X}_1 - \mathbf{X}_3) + \beta(\mathbf{X}_2 - \mathbf{X}_3) + \mathbf{X}_3 \quad \text{with} \quad \alpha + \beta \leq 1. \tag{5.7}$$

Setting $u = \alpha$ and $v = \beta$ in (5.7) yields the parameterization

$$P_{\mathsf{T}}(u, v) = u \begin{pmatrix} X_1 - X_3 \\ Y_1 - Y_3 \\ Z_1 - Z_3 \end{pmatrix} + v \begin{pmatrix} X_2 - X_3 \\ Y_2 - Y_3 \\ Z_2 - Z_3 \end{pmatrix} + \begin{pmatrix} X_3 \\ Y_3 \\ Z_3 \end{pmatrix} \tag{5.8}$$

of the triangle T. From (5.7) we see that α and β and hence u and v do not depend on the triangle corners. Therefore, the domain is

$$\mathsf{D} = \{(u, v) : u, v \geq 0, u + v \leq 1\} \tag{5.9}$$

for the parameterizations $P_{\mathsf{T}_j}(u, v)$ of each triangle T_j. We calculate the area moments m'_{pq} of D as

$$m'_{pq} = \iint_{\mathsf{D}} u^p v^q \, \mathrm{d}u \, \mathrm{d}v, \tag{5.10}$$

The relevant area moments of D up to third order are

$$\begin{aligned} m'_{00} &= \frac{1}{2}, \\ m'_{01} = m'_{10} &= \frac{1}{6}, \\ m'_{02} = m'_{20} &= \frac{1}{12}, \\ m'_{11} &= \frac{1}{24}, \\ m'_{03} = m'_{30} &= \frac{1}{20}, \\ m'_{12} = m'_{21} &= \frac{1}{60}. \end{aligned} \tag{5.11}$$

Second, we show that computing the 3D surface moments m_{kln} can be reduced to the calculation of the 2D area moments m'_{pq}. Starting with (5.5)

and setting $C = \sqrt{EG - F^2}$ we get

$$
\begin{aligned}
m_{kln} &= \iint_{\mathsf{D}} X_{\mathsf{T}}^k(u,v) Y_{\mathsf{T}}^l(u,v) Z_{\mathsf{T}}^n(u,v) \sqrt{EG - F^2}\,\mathrm{d}u\,\mathrm{d}v \\
&= C \iint_{\mathsf{D}} \big[(u(X_1 - X_3) + v(X_2 - X_3) + X_3)^k (u(Y_1 - Y_3) + v(Y_2 - Y_3) + Y_3)^l \\
&\qquad (u(Z_1 - Z_3) + v(Z_2 - Z_3) + Z_3)^n\big]\,\mathrm{d}u\,\mathrm{d}v \qquad (5.12) \\
&= C \iint_{\mathsf{D}} \left[u^{k+l+n}(X_1 - X_3)^k (Y_1 - Y_3)^l (Z_1 - Z_3)^n + \ldots + X_3^k Y_3^l Z_3^n\right]\,\mathrm{d}u\,\mathrm{d}v \\
&\qquad (5.13) \\
&= C \left[(X_1 - X_3)^k (Y_1 - Y_3)^l (Z_1 - Z_3)^n \iint_{\mathsf{D}} u^{k+l+n}\,\mathrm{d}u\,\mathrm{d}v + \right. \\
&\qquad \left. \ldots + X_3^k Y_3^l Z_3^n \iint_{\mathsf{D}} \mathrm{d}u\,\mathrm{d}v\right] \qquad (5.14) \\
&= C \left[(X_1 - X_3)^k (Y_1 - Y_3)^l (Z_1 - Z_3)^n m'_{(k+l+n)0} + \ldots + X_3^k Y_3^l Z_3^n m'_{00}\right]. \\
&\qquad (5.15)
\end{aligned}
$$

So, we start by applying the parameterization (5.8) and yield (5.12). Then, we expand the expressions in parentheses and reach the coefficients of $u^i v^j$, $i, j \in \mathbb{N}$ in (5.13). Equation (5.14) phrases the integral of sums as a sum of integrals, which (5.15) identifies as the 2D area moments m'_{pq}.

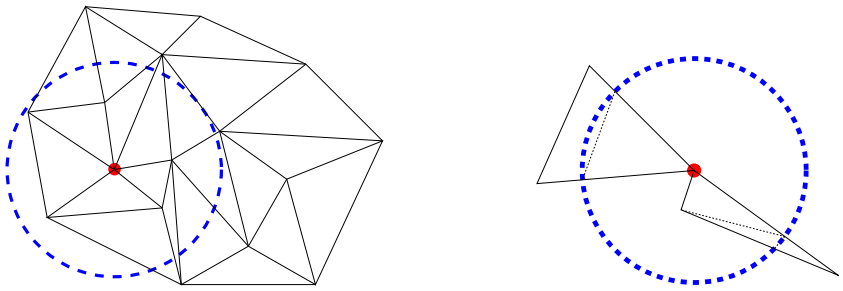
By this means we find an expression for the 3D surface moments of a triangle depending only on the corner points of the triangle and on the constant 2D area moments of the parameterization domain of the triangle. This yields exact 3D surface moments of 3D triangles, and hence of 3D surface triangulations, in an efficient and direct manner. We provide the surface moments used in this work in Appendix B.

Deriving moment invariants. Using the 3D surface moments M_{kln}, we finally compute the 11 3D Euclidean moment invariants I_{22}^2, I_{222}^2, …, I_{1113}^3 proposed by Lo and Don in [Lo and Don, 1989]. These invariants are derived algebraically and include moments up to third order. For the sake of readability, we shift the listing of the moment invariants to Appendix C.

By choice of the user, these invariants are established with respect to Euclidean or similarity transformations, respectively. Since we concentrate on Euclidean invariants for this work and use non-centralized moments, we are able to apply further invariants. Trivially, the area $M_{000} = I_0$ is invariant with respect to Euclidean transformations. Likewise, the 2D moment invariant with respect to rotations $m_{10}^2 + m_{01}^2$ as noted in [Süße and Ortmann, 2003] can easily be transferred to 3D yielding

$$I_1 = M_{100}^2 + M_{010}^2 + M_{001}^2. \tag{5.16}$$

In the case of centralized moments, $I_1 \equiv 0$ is still invariant, but pointless. This fact emphasizes the special nature of invariants based on non-centralized moments. In conclusion, we apply the 13-vector of Euclidean moment invariants $(I_0, I_1, I_{22}^2, \ldots, I_{1113}^3)^\mathsf{T}$ as the descriptor of the respective vertex.



(a) Local surface region (blue dashed) around a point (red).

(b) Approximation of triangles that have one or two corner points inside the local region. New edges (black dashed) endow new triangles.

Figure 5.6: Spherical local regions (blue dashed) confine the triangles to use for the calculation of moment invariants for the respective point (red). The resulting feature describes the local surface around the point.

Regarding a local region. Given a set of 3D triangles T and the according corner points X, we are now able to compute surface moments and moment invariants. As pointed out in Section 5.2, the possibility of partial overlap

requires to apply a local treatment in some way. Descriptors using global information gain maximal distinctiveness and robustness to noise. Yet, this requires the input surface triangulations to represent exactly the same real-world surface. For partial overlap, one way out is to calculate surface moments from infinitesimal small regions, which causes high sensitivity to noise. Figure 5.6 displays the way implemented in this work, which is an intermediate way between global and infinitesimal small regions. We compute the local surface moment invariants around point $\mathbf{X}$ by regarding only the triangles that are located inside a sphere $S(\mathbf{X}, r)$ centered at $\mathbf{X}$ with radius r,

$$M_{kln}(\mathbf{X}) = \sum_{\mathsf{T} \text{ in } S(\mathbf{X},r)} m_{kln}^{(\mathsf{T})}. \tag{5.17}$$

Different densities and distribution of the vertices may cause problems if we strictly accept or decline only entire triangles. Especially for large triangles, this treatment would take into account triangles that are larger than the local region, or decline complete triangles that exceed this local region. Instead, we approximate certain triangles by using a subset of them as follows. If a triangle has one or two corners inside the sphere, we approximate the part of the triangle inside the local region using the intersection points of sphere and triangle, cf. Figure 5.6. In this way, the method assumes identity of the surfaces only inside the local regions. On the one hand, this step reduces the information used to compute the descriptor of point $\mathbf{X}$, which inhibits feature distinctiveness and robustness against noise. On the other hand, this local treatment handles partial overlap of the input surfaces in terms of ignoring missing surface parts outside the local region. Our experiments demonstrate the robustness of the proposed registration method with respect to different choices of the sphere radius.

5.4.2 Feature Selection

The local surface restriction for the calculation of descriptors causes the need to filter out weak descriptors, as also stated in [Suikerbuik et al., 2004]. An obvious example of such a weak descriptor is a point within a planar region. This descriptor can be assigned to any other descriptor from a planar region without raising an assignment error. As a consequence, structural information within the local surface region is necessary to achieve distinctive point descriptors. This is another instance of the *aperture problem*. Our current

solution to this problem is a statistical analysis of each input surface as described in the following. First, we calculate the maximal Euclidean distance between descriptors $d_{\max}$ of vertices on the surface. Second, we compute a local measure of distinctiveness for the considered point. For a point $\mathbf{X}$ and the local surface region inside the sphere with radius r, we calculate the mean distance $\overline{d}_{\text{loc}}$ between the descriptor of $\mathbf{X}$ and the descriptors of all other vertices inside the sphere. By comparing $\overline{d}_{\text{loc}}$ to $d_{\max}$ and thresholding we decide if the point is used for assignment.

5.4.3 Registration by Optimal Point Assignment

Having two surface triangulations S_1, S_2 as input data, we compute the 3D moment invariants for each vertex of each surface. Specifically, the origin is set into the respective point and moments are non-centralized. This yields *3D moment invariants as point descriptors* characterizing the position of the point within the considered surface or, in other words, characterizing the surface around the considered point. The descriptors are calculated from the surface itself without any fitting or sampling, and the property of invariance is derived algebraically.

We construct the cost matrix necessary for the Hungarian method using the Euclidean distances between point descriptors. The Hungarian method proposed in [Kuhn, 1955] assures a point assignment with minimal global cost and also works for point sets of unequal size. The cost function implements the cost of assigning two descriptors simply as their Euclidean distance. While the complexity of the Hungarian method was in $O(P^4)$ using P points, originally, proper data structures endow a complexity in $O(P^3)$ as shown in [Luo and Hancock, 2003]. We thus register the surface triangulations S_1, S_2 by assigning point descriptors – and hence the points – of the respective surfaces to each other without necessarily assigning all descriptors. The 3D point correspondences allow to use the method proposed in [Walker et al., 1991], which estimates a globally optimal 3D Euclidean transformation

The assignment of surface points raises the discretization problem: We want to assign vertices of S_1 to vertices of S_2 that were sampled at the same position of the *real-world surface* $\mathfrak{S}$. Therefore, we have to assume structure-preserving sampling by S_1, S_2 and at least one S_j dense enough to roughly cover the sampling positions of the other surface triangulation. The practical meaning of this statement is that a human observer should be able to identify both surfaces referencing the same object, which is a natural condition for a

solvable registration task. For the difficult case of two very roughly sampled surfaces, future work may deal with suitable resampling of the input surface triangulations.

5.5 Experimental Evaluation

In the following we present quantitative and qualitative evaluations of the method described in the previous sections. We show the benefits of our registration approach with respect to different point densities of the surface triangulations, various noise levels, and overlap. Additionally, we highlight possible applications to classification and identification tasks. In conclusion, we like to emphasize the method's capabilities as a preprocessing to the ICP algorithm and the broad application of the derived invariants as surface descriptors.

Figure 5.5 illustrates one of the test data sets used. We obtain a high quality 3D reconstruction of *Dino Detlef* using a fringe-projection measurement system. All surface triangulations are established by standard Delaunay triangulation and a surface simplification algorithm, see [Garland and Heckbert, 1997], included in `Meshlab`, `http://meshlab.sourceforge.net/`. By means of this software, we also achieve 3D models of the same object with different point densities and different point distributions. It is known that the challenges for a registration method stated by technical objects differ from natural objects. This due to local self-similarities and symmetries that are very likely for technical objects. Hence, we also apply the proposed registration method to 3D surfaces of the NBV test object presented in Section 6.1. The used surfaces originate from the 3D CAD model and from fringe-projection measurements.

Throughout the experiments we use 3D moment invariants with respect to Euclidean transformations. For the quantitative evaluation of the alignment quality we determine the error $e_{\text{coarse,V}}$ as the mean vertex-to-closest-vertex distance between the input surfaces and the error $e_{\text{coarse,S}}$ as the mean vertex-to-closest-surface-point distance. These measures describe the alignment accuracy of our coarse registration. After the registration step, we perform an ICP alignment using an own implementation. By visual inspection we check for ICP convergence to the true solution and note the resulting alignment errors $e_{\text{ICP,V}}$ and $e_{\text{ICP,S}}$, which are the mean vertex-to-vertex and vertex-to-surface distances. We assume the fact to be known that the ICP

algorithm needs an initial solution inside its basin of convergence to succeed, cf. [Rusinkiewicz and Levoy, 2001]. Thus, we omit tests on ICP alone and only give the ICP results using our coarse registration as the initial step. The noted runtimes (system: Core2 Duo, 2.4 GHz, 4 GB RAM) comprise all steps of the whole registration method, in particular calculating the surface moments, calculating the invariants, and running the Hungarian method. For the identification experiments, the respective sections describe the specific test data and criteria.

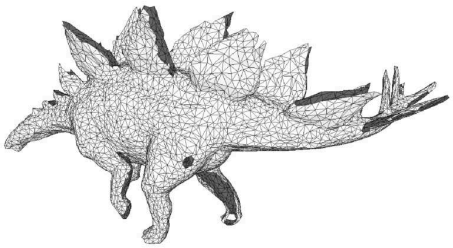
5.5.1 The Influence of Different Point Densities on the 3D Registration

For the first experiment, we reduce the density of the initial 3D model from Figure 5.5(b) to yield surfaces S_j with V_j vertices and F_j faces (triangles), $j = 1, 2$, using the surface simplification algorithm presented in [Garland and Heckbert, 1997] that is implemented in `Meshlab 1.1.1`. Due to the complexity of the dinosaur, see Figure 5.5, each reduction of the point density comes with a reduction of the preserved surface information. The registration results are listed in Table 5.1. We note the vertex/face numbers of each surface, which indicate the loss of information about the surface structure. Runtimes, error values, and ICP convergence information are also given. In order to obtain results that are meaningful with respect to different point densities, we compute global features using the whole surface and, thus, eliminate effects arising from the size of a local region, which we will evaluate later on.

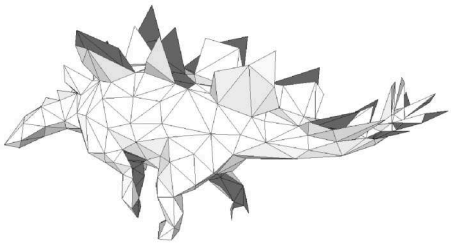
Table 5.1: Evaluating registration of surface triangulations with different point densities. Our method proofs to be robust. Alignment convergence is reached for strongly differing densities.

V_1/F_1	V_2/F_2	runtime (s)	$e_{\text{coarse,V}}$ (mm)	$e_{\text{coarse,S}}$ (mm)	ICP conv.	$e_{\text{ICP,V}}$ (mm)	$e_{\text{ICP,S}}$ (mm)
5274/10000	5274/10000	87.53	0.00	0.00	`yes`	0.00	0.00
5274/10000	2670/5000	52.17	0.51	0.18	`yes`	0.51	0.18
5274/10000	558/1000	43.59	1.19	0.71	`yes`	1.00	0.46
5274/10000	290/500	38.48	3.37	1.31	`yes`	1.11	0.58
5274/10000	58/100	52.48	9.82	2.74	`no`	8.60	2.71

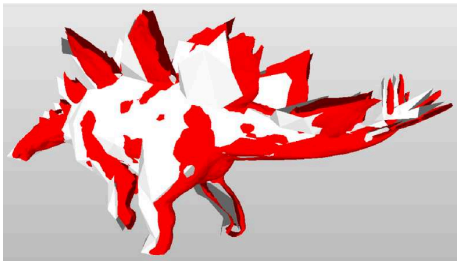
As seen from Table 5.1 and Figure 5.7, our method produces suitable



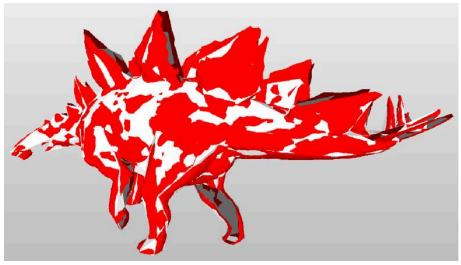
(a) Surface triangulation S_1 with 10000 triangles.



(b) Surface triangulation S_2 with 500 triangles.



(c) Coarse Registration of S_1 (red) and S_2 (white) achieved by the proposed method.



(d) Registration refinement by ICP, initialized by coarse registration.

Figure 5.7: Test data and results for Section 5.5.1. In spite of strongly differing sampling resolutions, we yield a 3D coarse registration valuable as an ICP initialization. Numerical results are given in Table 5.1.

initial solutions for the ICP algorithm even with extremely different point densities (10000 $\leftrightarrow$ 500 triangles). The method fails when applied to a pair of surfaces with 5274 vertices / 10000 faces and 58 vertices / 100 faces. The reason is the large loss in information when representing the complex dinosaur model with only 58 points. In case of such a strong simplification, whole object parts like the plates of the corselet vanish. This loss of semantic object parts makes it hard to identify a reasonable alignment even for a human observer.

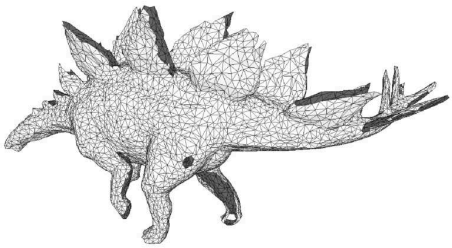
Table 5.2: Evaluating registration of surface triangulations (10000 ↔ 1000 faces) with Gaussian noise of different levels. For single cases, the coarse alignment still works with $\sigma = 4$mm, i. e. 99.73% of all vertex X/Y/Z-coordinates are disturbed up to 12mm in each dimension within the 3D model of length 200mm.

σ (mm)	$\overline{e}_{\text{coarse,V}}$ (mm)	$\overline{e}_{\text{coarse,S}}$ (mm)	ICP convergence		$\overline{e}_{\text{ICP,V}}$ (mm)	$\overline{e}_{\text{ICP,S}}$ (mm)
			yes	no		
0.0	1.19	0.71	10x	0x	1.00	0.46
0.1	1.24	0.75	10x	0x	1.00	0.47
0.5	1.50	0.83	10x	0x	1.12	0.64
1.0	1.57	0.92	10x	0x	1.39	0.86
2.0	3.15	1.50	10x	0x	2.03	1.16
3.0	5.71	2.19	9x	1x	2.65	1.40
4.0	23.01	3.74	2x	8x	7.88	2.50

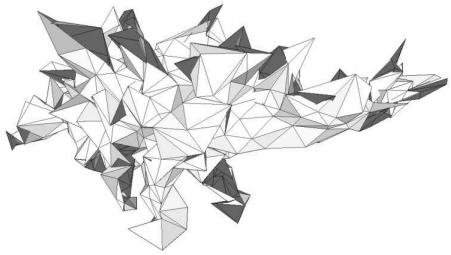
5.5.2 The Influence of Additional Noise on the 3D Registration

The robustness with respect to perturbations is crucial for registration methods that are used with real data from practical measurements. Therefore, we evaluate our coarse registration method using noisy input data. We register surfaces S_1 (5274 vertices / 10000 faces) and S_2 (558 vertices / 1000 faces), where each vertex coordinate of S_2 is disturbed by Gaussian noise $\mathcal{N}(0, \sigma^2)$. This means that each of the $X/Y/Z$-coordinates of each vertex is independently disturbed by Gaussian noise with standard deviation σ. The registration results are listed in Table 5.2. All error values $\overline{e}_{\cdot,\cdot}$ are mean values out of ten independent cycles of disturbance and registration. The runtime was about 44 s for each cycle.

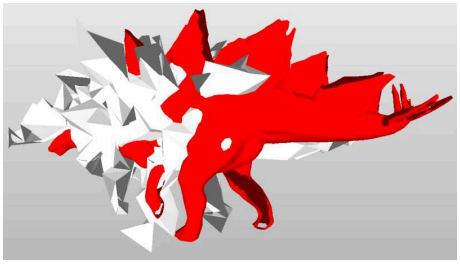
Table 5.2 shows that our registration method provides valuable coarse alignments with noise levels up to $\sigma = 3$mm. That means the 3D model of *Dino Detlef* of length 200mm is disturbed in such a way that 99.73% of all vertex $X/Y/Z$-coordinates are displaced up to 9mm in each dimension, and the following ICP converges in presence of such severe surface deformations. For single cases with $\sigma = 4$mm as in Figure 5.8, the global surface information does still allow to produce a proper coarse registration.



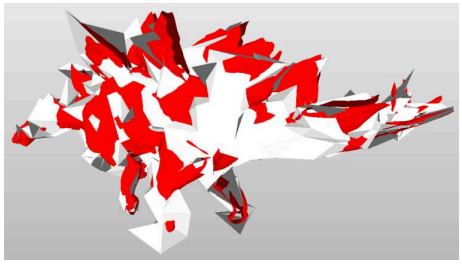
(a) Surface triangulation S_1 with 10000 triangles, length about 200mm.



(b) Surface triangulation S_2 with 1000 triangles, Gaussian noise with $\sigma = 4$mm.



(c) Coarse Registration of S_1 (red) and S_2 (white) achieved by the proposed method.



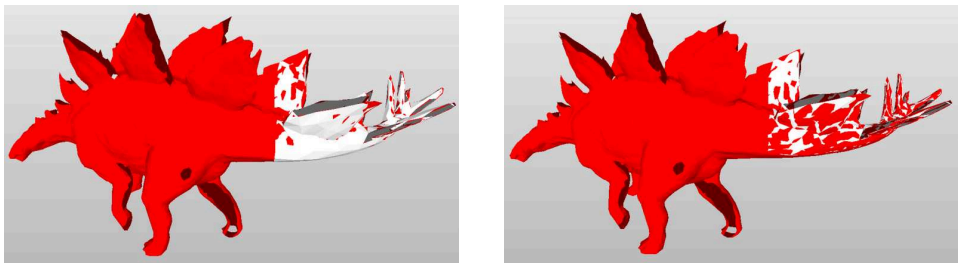
(d) Registration refinement by ICP, initialized by coarse registration.

Figure 5.8: Test data and results for Section 5.5.2. For single cases, the proposed method handles Gaussian noise up to $\sigma = 4$mm applied to the vertex coordinates. Numerical results are given in Table 5.2.

5.5.3 The Influence of Real Partial Overlap on the 3D Registration

A further practical issue is real partial overlap of the surface triangulations S_1 and S_2. This is to say that the given surface triangulations do not represent the same real-world surface $\mathfrak{S}$. At least one surface triangulation covers a part of the real-world surface that is not covered by the other surface triangulation. The partial overlap criterion is evaluated using the 3D surfaces seen in Figure 5.5. Surface S_1 is the whole 3D model of the dinosaur with 5274 vertices / 10000 faces, S_2 is the tail (length about 110mm) with 290 vertices / 500 faces from a less dense version of the 3D model. The moment invariants of each vertex are calculated with respect to the local surface region

around the vertex which lies inside a sphere of radius r around the vertex, cf. Figure 5.6.



(a) Coarse Alignment of the whole dino model (red) with 10000 triangles and the tail part (white) with 500 triangles.

(b) Using the coarse alignment, ICP reaches a refinement.

Figure 5.9: Test result for Section 5.5.3 and r = 25mm. The partially overlapping input surfaces are illustrated in Figures 5.5(a) and 5.5(b). Numerical results are listed in Table 5.3.

First, Table 5.3 shows that our registration method can handle real partial overlap effectively. Figure 5.9 gives an example. There is, compared to the size of the 3D models, a large range of valid values for the sphere radius r to achieve a suitable initial solution for the ICP algorithm. Valid values of r are between 10mm and 55mm. Smaller local regions do not contain enough significant surface structure. If the radius is greater than 55mm, then an increasing part of the local regions covers the whole tail part that is about 110mm long. In this case, the majority of the local regions around vertices are in touch with the front border of the tail part, which contains different surface information in the whole dinosaur model and yields hence different surface features. Second, we note that ICP does not reach convergence for all values of r. Although the configuration of r seems to be non-critical and may be performed by selecting r about one third of the largest extension of the smaller surface, this remains an open point requiring future work.

5.5.4 Registration Results Using Technical Surfaces with Real Partial Overlap

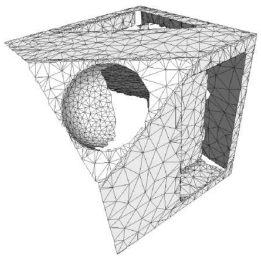
Considering the technical NBV test object developed in Section 6.1 and the non-technical 3D model of *Dino Detlef*, we immediately notice severely dif-

Table 5.3: Evaluating registration in the presence of partial overlap. ICP reaches convergence for a large range of values for radius r.

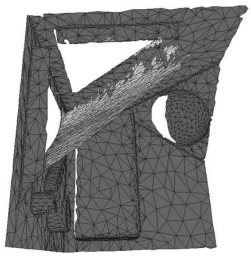
r (mm)	runtime (s)	$e_{\text{coarse,V}}$ (mm)	$e_{\text{coarse,S}}$ (mm)	ICP conv.	$e_{\text{ICP,V}}$ (mm)	$e_{\text{ICP,S}}$ (mm)
5	6.29	5.91	2.19	no	5.92	2.19
10	10.19	0.95	0.52	yes	0.86	0.34
15	15.41	0.90	0.43	yes	0.86	0.34
20	22.54	0.92	0.45	yes	0.87	0.35
25	31.43	0.90	0.42	yes	0.86	0.33
30	42.50	0.90	0.43	yes	0.87	0.35
35	54.69	0.94	0.46	yes	0.86	0.34
40	69.09	1.11	0.63	yes	1.09	0.60
45	84.94	0.94	0.47	yes	0.90	0.41
50	101.87	1.01	0.54	yes	0.86	0.33
55	117.69	0.92	0.45	yes	0.87	0.35
60	132.62	10.78	2.43	no	6.48	2.29
65	151.78	6.89	2.27	no	6.90	2.27

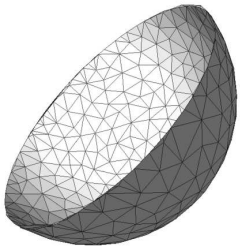
fering surface properties. These differences regard significant local surface structure, local self-similarities, and dominant edge directions. Technical objects tend to feature planar local regions that do not provide any distinctiveness, corners and edges that are locally identical to other regions on the same object, and few dominant edge directions. On the contrary, non-technical objects are likely to fulfill the exact opposites. While the specific characteristics of each object type favors certain approaches for 3D registration, it is really hard to successfully tackle both types of object surfaces with the same method. The above tests give evidence that the proposed registration method effectively deals with the non-technical dinosaur figurine. Now, we probe the method's performance utilizing the surface triangulations illustrated in Figure 5.10. In addition to the difficulties mentioned above, the input surfaces provide partial overlap in a way that none of the surfaces is covered completely by the other one. Hence, only local features are useful in this case, and we evaluate registration results with respect to different radii of the local region around each vertex. For the sake of completeness, we also present the alignment errors as in the previous sections. Nonetheless, we clarify the strictly confined value of the these numbers, since useful distances

require the respective corresponding points of both surfaces.

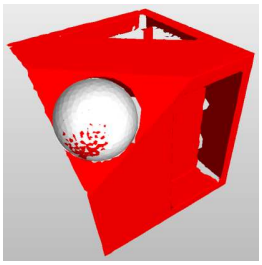


(a) Different views of S_1, which is a partial scan of the NBV test object with 5000 triangles. The edge length of the basis cube is 160mm.

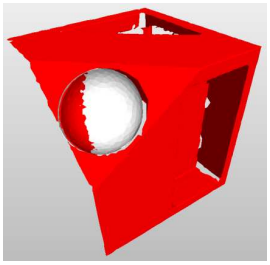




(b) Half sphere S_2 with 650 triangles extracted from a 3D CAD model of the NBV test object.



(c) Coarse alignment using $\mathbf{r = 10mm}$.



(d) ICP refinement based on the coarse registration.

Figure 5.10: Test data and result for Section 5.5.4. The partially overlapping, technical surfaces pose a special challenge for the proposed registration method. Numerical results are listed in Table 5.4.

The illustration in Figure 5.10 shows a successful registration of the input surface triangulations using $r = 10$mm. Compared to the radius of the half sphere, which is 32.5mm, the proposed registration method converges for a large range of radii reaching from 5mm to 20mm. This is evidence that our coarse registration can deal with technical and non-technical surfaces, alike. Furthermore, this experiment evolves a way to identify a given 3D CAD model within a reconstructed 3D surface triangulation.

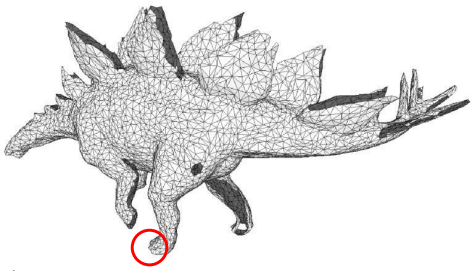
Table 5.4: Evaluating registration of technical surfaces shown in Figure 5.10 in the presence of partial overlap. ICP reaches convergence for a large range of values for radius r, compared to the 32.5mm radius of the half sphere in Figure 5.10(b).

r (mm)	runtime (s)	$e_{\text{coarse,V}}$ (mm)	$e_{\text{coarse,S}}$ (mm)	ICP conv.	$e_{\text{ICP,V}}$ (mm)	$e_{\text{ICP,S}}$ (mm)
2	1.45	15.83	3.72	no	15.58	3.69
5	2.75	11.74	2.75	yes	11.41	2.58
10	3.84	6.20	1.87	yes	4.97	1.71
15	4.33	12.74	3.23	yes	10.58	2.58
20	5.07	11.70	2.55	yes	11.33	2.63
25	5.93	10.63	2.76	no	9.03	2.54
30	7.14	9.24	2.60	no	9.01	2.56

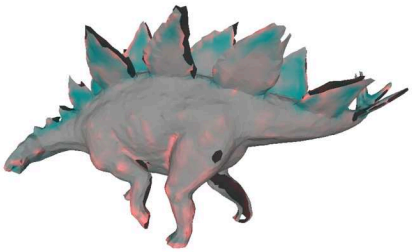
5.5.5 Assessing Surface Similarity and Clustering

Having seen the noteworthy registration performance in the previous sections, we find it valuable to have a closer look at how much the actual feature quality contributes to these results. To this end we demonstrate a visualization of feature similarities. The experimental setup uses one surface triangulation and computes local features inside a spherical region around each vertex. Given a reference vertex on the same surface triangulation, we compare each feature vector to the one of the reference vertex. This yields a sorted list of all feature vectors that we use to colorize the surface vertices in accordance to the respective position in the sorted list. Besides assessing local self-similarities of the surface triangulation, this experiment demonstrates the use of the proposed surface features for tasks of retrieval, identification, and classification.

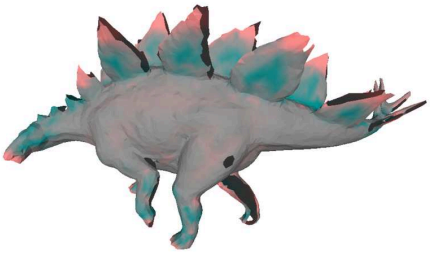
The Figures 5.11 and 5.12 show self-similarities with respect to different sizes of the spherical local region used for the feature computation. Please note that the colors refer to the rank of the respective vertex feature in the list sorted with respect to the differences to the reference feature. This is to say that the color differences do not depend on the feature differences but only on their rank. The results vary for different sizes of the local region. For $r = 5$mm, i. e. the local regions are spheres with a diameter of 10mm, we observe a relatively noisy surface coloring. This effect is reasonable since



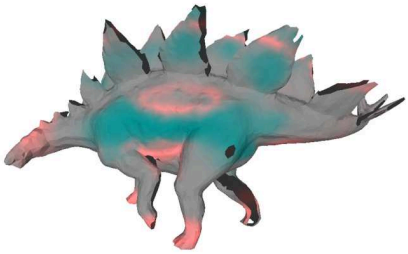
(a) 3D dino surface. The red circle marks the reference vertex.



(b) Local self-similarities using spherical local regions around each vertex with radius $r = 5$mm.



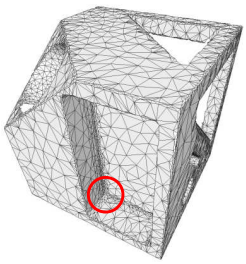
(c) Using $r = 10$mm.



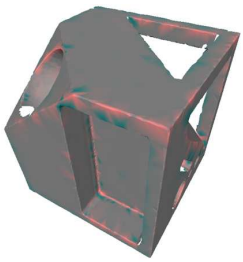
(d) Using $r = 40$mm.

Figure 5.11: Comparing local self-similarities as depicted in Section 5.5.5 using the 3D reconstruction of the dinosaur, the length of which is about 200mm. We select the middle toe on the left back foot of the model as the reference vertex. The colors indicate feature similarity of the respective vertex regarding the feature of the reference point, reaching from red (most similar) over gray to blue. Note that the colors depend on the similarity rank.

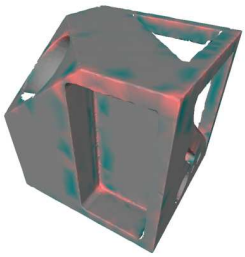
the scope of the feature computation is relatively small. The larger the local region, the more smooth gets the coloring. While these are trivial observations, they are still important to the application and the input surface at hand. For instance, Figure 5.11 shows that the choice $r = 10$mm allows to easily identify protruding elements of the dinosaur model similar to the back left foot, like the head and the corselet plates. On the other hand, $r = 5$mm provides identification of smaller pleats and edges on the surface similar to the closer region of the reference vertex. With $r = 40mm$, the local region is able to comprise nearly half of the dinosaur model. This does not seem to be a good choice aiming to identify a small region like the back left foot, if this is the task given. The same reasonable effects appear when using



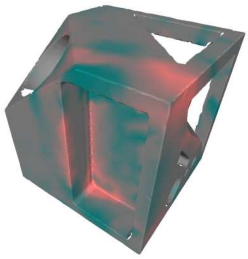
(a) 3D surface of the NBV object. The red circle marks the reference vertex.



(b) Local self-similarities using spherical local regions around each vertex with radius $r = 5$mm.



(c) Using $\mathsf{r} = 10$mm.

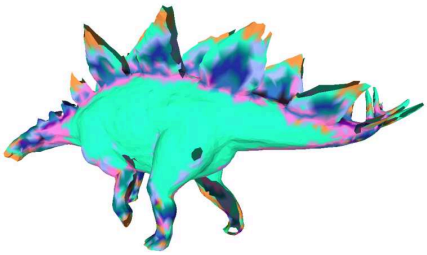


(d) Using $\mathsf{r} = 40$mm.

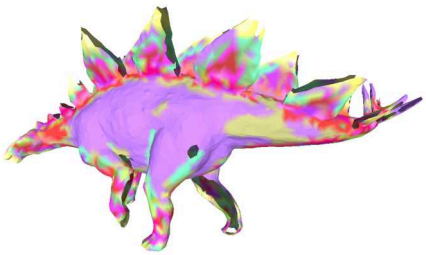
Figure 5.12: Comparing local self-similarities as depicted in Section 5.5.5 using the 3D reconstruction of the NBV object, the edge length of which is 160mm. We select a corner of the notch as the reference vertex. The colors indicate feature similarity of the respective vertex regarding the feature of the reference point, reaching from red (most similar) over gray to blue. Note that the colors depend on the similarity rank.

the 3D reconstruction of the NBV object as in Figure 5.12. The measured similarity is visually reasonable with respect to the selected radius of the local region. For $r = 5$mm the features at corners and along edges provide obvious relative similarities to the reference vertex in the corner of the notch. This examination visually confirms that it is possible to easily find suitable values for the local region in order to solve tasks of identification and classification using 3D surface triangulations.

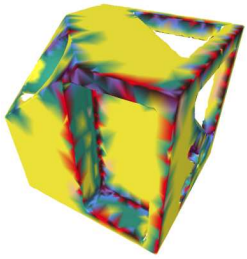
One application exploiting feature similarities is clustering of surface triangulations. As a feasibility study, we perform k-means clustering, cf. [Lloyd, 1982], of the local features of surface triangulations and illustrate the result by identically coloring the vertices, the features of which belong to the



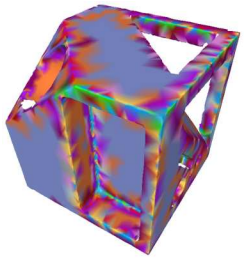
(a) Clustering the features of the dinosaur surface with 10 clusters.



(b) Clustering the features of the dinosaur surface with 25 clusters.



(c) Clustering the features of the NBV surface with 10 clusters.



(d) Clustering the features of the NBV surface with 25 clusters.

Figure 5.13: Clustering of local surface features (r = 10mm) according to Section 5.5.5. Each random color represents a cluster. The vertices of the surface triangulation are assigned to the cluster of their respective surface feature.

same cluster. Since we randomly pick features to initialize the cluster means, we have to apply further steps to favor a reasonable result. First, we start with an over-segmentation by initializing twice as much clusters as finally wanted. Second, we alternate computational steps performing robust reinitialization of cluster means and unification of the two clusters having means closest to each other. The robust reinitialization of cluster means recomputes each cluster mean with respect to the best 0.5-quantile of the cluster elements, which is followed by k-means clustering using the changed means. In this part, it is possible to reduce the number of clusters depending on the data and the current clusters. The unification of closest clusters simply identifies the pair of means providing the minimal Euclidean distance over all of these pairs and merges these clusters, which always reduces the cluster number by one. Starting with 20 and 50 clusters and employing steps of

reinitialization and unification, we finally create segmentations with 10 and 25 clusters, respectively.

We apply the above clustering method to the reconstructed surface triangulations of *Dino Detlef* and the NBV object. The computation of local features uses a radius of 10mm. Figure 5.13 shows that the resulting clusters according to the proposed surface features provide a kind of semantic segmentation of the surface, as dealt with in [Fröhlich et al., 2010]. In Figure 5.13(a), protruding parts of the dinosaur share the same, light-brown colored cluster, as well as bend surface regions sharing the cluster colored pink. Using 25 clusters instead of 10, Figure 5.13(b) illustrates a finer segmentation of the respective surface parts. Likewise, the Figures 5.13(c) and 5.13(d) demonstrate the semantic clustering of the NBV object. We are able to identify clusters representing plane regions, bend regions, corners, and edges. Again, we emphasize two facts concerning the radius of the local region used for the feature computation. First, it is obvious that the result strongly depends on the chosen radius since a feature can only describe what it has seen. Second, determining a proper radius is very intuitive, if done by the user. Otherwise, we propose to use more than one value for the radius and finally take the result that is the best with respect to some well-defined criterion. Another possible way is a preprocessing that yields the optimal radius for the task and the input surface at hand.

5.6 Conclusion and Future Work

We conclude this chapter and highlight the main aspects of the proposed method for the registration of 3D surface triangulations. We explore a feature-based approach performing a globally optimal assignment of vertex features using the Hungarian method without any initial solution. As features we use algebraically derived moment invariants known from the literature. To form these invariants, we compute exact moments of 3D surface triangulations in a direct and efficient way. The surface moments are non-centralized and use the respective vertex as the origin of the coordinate frame. In this way, the surface moments and, hence, the moment invariants characterize the, possibly local, surface around the focused vertex. This kind of feature computation of surface triangulations provides well-conditioned descriptors that are also valid for tasks of identification, classification, and retrieval. With respect to view planning, the methods presented in this chapter allow

to identify certain object features and treat them in a purposive manner depending on the planning goals.

The experimental evaluation assesses the performance of the proposed registration method and the quality of the proposed features in different concerns. We demonstrate that the registration method effectively handles surface triangulations featuring different vertex densities and distributions, noise, and partial overlap. These results apply to the tested technical and non-technical objects, alike. The additional application of the ICP algorithm refines the registration result. Thus, the proposed method further appears as a proper initialization step for an ICP alignment. Additional fields of application are provided by the invariant surface features alone. Well-conditioned descriptors facilitate solving classification and related tasks, which is underlined by experiments performing similarity analysis and vertex clustering. By this means, the presented surface features touch the active research area of semantic segmentation.

As various the benefits and applications are, so are the possibilities to improve and extend the registration method in future research. Up to now, we focus on moment invariants with respect to Euclidean transformations. Considering the broad field of 3D applications, it is valuable to extend the invariants to similarity transformations, which is supported by the method. In doing so, we can identify any half sphere within the input surface using one specific 3D CAD model, for instance. While this is already possible for global features, the problems come with the necessary definition of a local region with respect to similarity transformations. Imagine two related surfaces with unknown global scalings and partial overlap. Then the sizes of the local regions should be chosen such that the ratio of object scalings equals the ratio of the sizes of the local regions. This requires the ratio to be known, which would allow to scale one surface accordingly and to use Euclidean invariants. In other words, a proper way to define local regions in presence of similarity transformations is not yet found. Another point worth examining is including further neighborhood information into the registration procedure. The benefit is getting obvious if you think about technical objects featuring many local ambiguities and symmetries, for instance, a corner of a cube. While a local surface region comprising enough unique data may be too large regarding partial overlap, there may be ways to include further information about the respective neighborhoods into the registration method, which may resolve ambiguities. Considering the additional challenge of two different surface triangulations being sampled very roughly, a proper resampling of the

input surfaces appears as a valuable goal. This preprocessing redistributes the vertices and hence eases the assignment of the respective vertex descriptors. The statistical feature selection presented above is our first approach to filter out features of non-distinctive surface regions. In future research, we like to investigate methods to constructively detect 3D surface features, which is similar to the feature detection in image data. At last, we suggest to examine ways to compute a suitable radius for the local surface regions automatically. Despite the robustness of the presented registration method with respect to different choices for the radius, which has been shown in the experiments, an automatic computation relieves the user of this burden.

Chapter 6

A Modular Approach to Online Next-best-view Planning

The above chapters dealing with GKLT tracking and 3D coarse registration are closely related to actual view planning. As we will show in this chapter, GKLT feature tracking directly leads to view planning for accuracy optimization using a passive camera. This is based on the 3D point estimation and covariance data provided by GKLT tracking. The depicted coarse registration and surface features allow a special treatment of geometric primitives and build a bridge towards model-based view planning within this data-driven approach. As a starting point, however, we find it most appealing to first think about quality criteria applying to view planning in Section 6.1. In the following Section 6.2, we outline our understanding of a generic, modular, online planning system using a passive camera. The modules of the proposed planning system get the focus in the consecutive sections. In Section 6.3 we present accuracy optimization based on uncertainty analysis. The probabilistic surface estimation proposed in Section 6.4 affords visibility analysis, Section 6.5, and the special treatment of geometric primitives, Section 6.6. By this means we construct a modular system capable of addressing most various planning goals and limitations. To conclude our theoretical examinations, we illuminate an additional aspect of view planning in Section 6.7, which is view planning with respect to the reconstruction method of factorization.

6.1 Benchmarking 3D Reconstructions from Next-best-view Planning

Before dealing with actual view planning using a passive camera, we consider it useful to figure out how to assess different general view planning methods, which include the ones using a passive camera. We base our statements on our presentations in [Munkelt et al., 2007]. In comparing view planning methods and results, two major problems of all past activities can be observed. First, researchers use their own test objects. Second, the reconstruction quality is measured by different criteria or, if these are denoted the same, the criteria may be formalized in different ways. Because of these shortcomings in standardization, the studied methods cannot be compared and the state of the art in this research area cannot be reviewed without further considerable effort, since the exact relative performance of the methods is unknown. A solution to this problem is a common reference test object featuring challenges for all kinds of active vision systems, like different scanner hardware and different planning methods. Further, a reference benchmark judges the quality of the reconstructed test object and, hence, judges the performance of the whole NBV planning system. In doing so, we settle for the way to assess a view planning method using the reconstruction result it produces, which is augmented by the number of views reflecting the necessary costs. Thus, the desired benchmark has to overcome the difficulty of being independent of the scanner hardware used, for example, IR/laser scanners, intensity cameras with/without fringe projection, and independent of the actual planning method.

6.1.1 Literature Review

Before we present a benchmarking system comprising a generic test object and quality criteria, we review previous approaches to measure the quality of a view planning system applied by other authors.

One particular aspect in benchmarking view planning methods concerns the test object. Researchers use most different objects to assess the specific performance of their respective method. Roberts and Marshall [Roberts and Marshall, 1998] employ an L-shaped block and a wedge. In contrast to these geometric primitives, Pito [Pito, 1999] uses a cup, and in [Li et al., 2005] the authors apply their proposed method to a dug figurine. Some attempts

have been made to construct specific test objects, either as a model of buildings in [Maver and Bajcsy, 1993] or as simulation models in [Banta et al., 2000,Munkelt et al., 2006]. However, from a benchmarking perspective, most of those approaches lack certain features that are needed in a view planning framework. With a few exceptions, the complexity of most test objects is rather low, especially in the sense that there are only small regions with self-occlusions. This leads to nearly 99% completeness within few views taken. While this might be adequate for a proof-of-concept, this is not a satisfying starting position for a detailed comparison between planning algorithms. Seitz et al. [Seitz et al., 2006] state, for instance, that the completeness numbers were not very discriminative. Therefore an adapted test object should contain a suitably large amount of concavities and occluded surface areas.

The second major benchmarking aspect is providing meaningful measures. While many authors, as in [Scott et al., 2003], propose the grazing angle as a subjective quality measure, we consider it crucial to evaluate the quality of the resulting 3D reconstruction including surface coverage. Seitz et al. [Seitz et al., 2006] perform an evaluation by computing the error that 90%, which is a user threshold, of all reconstructed points do not exceed. In [Girod et al., 2000], the authors compute the difference volume of surface meshes. Many authors consider the number of views and completeness, while the latter often is not formalized. Only few authors, cf. [Seitz et al., 2006,Wenhardt et al., 2006], take accuracy into account.

The above observations state a lack of standardization in terms of test objects and evaluation criteria. To overcome this situation in which different planning methods are hard to compare, we suggest a reference test object and reference quality criteria. Although, it is clear that a common minimal set of object details posing special challenges is hard to find. In the next sections, we present generic geometric details that a test object should consist of together with a discussion of an associated formal benchmark regarding reconstruction accuracy and completeness.

6.1.2 The NBV Test Object

In this section we want to discuss desirable attributes of an NBV test object. We then show how we realize those attributes using certain object details by presenting the specific reference object prototype shown in Figure 6.1. In the text, numbers in [brackets] refer to the respective object elements. A test object should not be symmetric, see the details [1], [2], and [4], which often

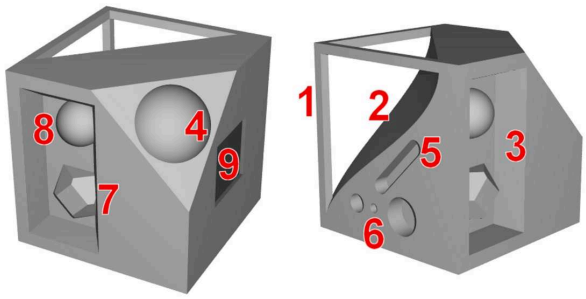


Figure 6.1: Model views of the proposed reference test object for view planning. The numbered elements are referred to in Section 6.1.2. The Figures 4.14 and 4.15 present views of the manufactured aluminum object.

leads to simplified, regularly spaced view plans. Furthermore, self occlusions are needed to challenge the planner, cf. [1], [3], [4], [7], and [8]. Additionally, when using active fringe projection systems, shadows should be cast onto the object as done by [1]. Curved surfaces, see [2], [4], [8], as well as sharp edges pose difficulties to different reconstruction approaches. To test the incorporation of the sensor model into the planner we need details that require special sensor alignment. In particular, the access to holes is a difficult planning issue and has to be tested, cf. [5], [6]. Optionally, length errors can be tested using [7] or [8]. Finally, a scanner resolution estimate should be provided as with [9].

The proposed test object has the overall shape of a cube, but it is not symmetric. It combines a set of object details covering the discussed requirements. By constraining its size to the bounding cube, the object satisfies the often implicit demand for complete enclosure into the measurement volume. To satisfy the need for a fine grained surface texture, laser labeling can optionally be applied. A common texture pattern for passive lighting techniques is available at `http://www.inf-cv.uni-jena.de/index.php?id=nbvbench`. Adaptation to different complexity levels can be achieved through a plug-in architecture of several details, which can be added as needed. The test object has five faces of interest that can be scanned from the upper half sphere around it. While normally standing on its bottom side, it can be placed on any of its sides. The test object features the following details.

- *Basic object setup.* This setup challenges basic planning and recon-

struction capabilities.

[1] *Tripod:* Three small rectangular elements that occlude and shade parts of detail [2]. Both the accuracy and completeness in the junctions are demanding.

[2] *Sinusoidal face:* Yields an asymmetric overall shape of the test object. The smooth, curved surface provides varying surface normals that complicate the reconstruction.

[3] *Notch:* Used as a cavity for two details of the full benchmark. Its side faces need to be scanned from appropriate positions.

[4] *Negative half sphere:* Constrained visibility and shadows challenge the reconstruction of its interior.

- *Full object setup.* In combination, all details pose a challenge to more sophisticated scanning and planning methods. The full test object extends the basic one by the following, additional details.

  [5] *Slotted hole:* Scanning this concavity requires a certain alignment of stereo systems. Concealable.

  [6] *Drill holes:* Three holes with proportions of diameter to depth from 2:1 through 1:1 to 1:2 act as prototypes for concavities of interest with increasing difficulty. Concealable.

  [7] *Frustrum of pyramid:* Common test detail with hard to scan side faces. Optionally, one could compare both the planarity of its five upper faces and the length deviation of their resulting intersections to their corresponding ground truth values. Pluggable.

  [8] *Positive half sphere:* Test detail with calibrated radius. Base of the detail is hard to scan. Pluggable.

  [9] *Riffle plate:* Plate with a 2D array of miniature frustrums of pyramids. The size of their top face decreases. Can be used to determine scanning resolution in object space. Pluggable.

These details, surely, do not represent all objects in the world, but as a union of abstract challenges they cover a wide range of real-world objects. The whole test object is not intended to evaluate a scanners physical resolution. However, to reasonably rate the reconstruction precision, a relative accuracy of 30μm for smoothness as well as length deviation is targeted for.

6.1.3 Evaluating the Planning Result

The test object suggested in the previous section poses a part of the input to a view planning method. Together with well-defined planning goals and constraints, the planner actively adjusts the available parameters and establishes an according 3D reconstruction of the test object. We propose a formalized benchmark that measures the performance of the planning method by evaluating the number of views and the quality of the resulting 3D reconstruction in terms of accuracy and completeness. Performing the benchmark requires the 3D reconstruction and the reference CAD model being aligned in the same coordinate frame.

6.1.3.1 Benchmarking Accuracy

As a measure of accuracy we use a modified Hausdorff metric according to [Dubuisson and Jain, 1994]. Practically, we assume a surface triangulation or a dense point cover of the reference model aligned to the 3D reconstruction. For each reconstructed 3D point we compute the distance to the closest reference point. Regarding all these distances, we present the mean error μ_e as the reconstruction accuracy. This does not prohibit to provide further information, for instance, in terms of a box-whisker plot, i. e. to provide the 0.025-, 0.250-, 0.500-, 0.750-, 0.975-quantiles and the outlying elements outside of this range. However, the compact accuracy benchmark is given by μ_e.

6.1.3.2 Benchmarking Completeness

Benchmarking completeness seems to be an ill-posed task if we consider reconstructed sets of 3D points. Incompleteness may be understood as gaps in the 3D reconstruction, but a finite, discrete set of 3D points features gaps everywhere. Avoiding this difficulty by applying non-discrete surfaces shifts the paradigm; in doing so, we yield a complete reconstruction, by definition. This kind of evaluation produces a 100%-completeness and an unrated accuracy regarding each surface point, which favors the difference volume as in [Girod et al., 2000] as a measure of accuracy. By this means we are forced to accept the quantitative evaluation as a measure of accuracy, but still miss a statement regarding completeness. Since non-discrete approaches lead to a shift of paradigms, as outlined above, we stick to the discrete 3D reconstruction in terms of finite, discrete sets of 3D points. For methods that actually

recover a non-discrete surface, we derive a point cover from the resulting surface. Now, we try to find a numerical representation that is meaningful with respect to reconstruction completeness.

What is completeness in the context of comparing a 3D CAD model and a finite, discrete set of 3D points? In a dual manner, a reasonable answer is that reconstruction incompleteness may be seen as gaps between 3D points that are larger than the other gaps. In the end, this assumption says that a complete 3D reconstruction provides homogeneously distributed 3D points. However, this seems not to be a suitable model. 3D points may obey a homogeneous distribution and, at the same time, be sparse, which corresponds to equally sized but large gaps between the points and is considered as incomplete. Therefore, we suggest to augment the statement concerning distributional homogeneity by the mean distance μ_d between a 3D point and its next neighbor. In this way we provide two numerical expressions. First, the coverage $c \in [0, 1]$ is a measure independent of the resolution of the 3D reconstruction and states how well the CAD model is covered by reconstructed points in the sense of homogeneity. Second, the mean distance μ_d between neighboring reconstructed points describes the resolution of the 3D reconstruction. Using these criteria, we find the simple abstract expression: *completeness* = [*coverage, resolution*]. One 3D reconstruction is more complete than another one if it features better coverage and higher resolution.

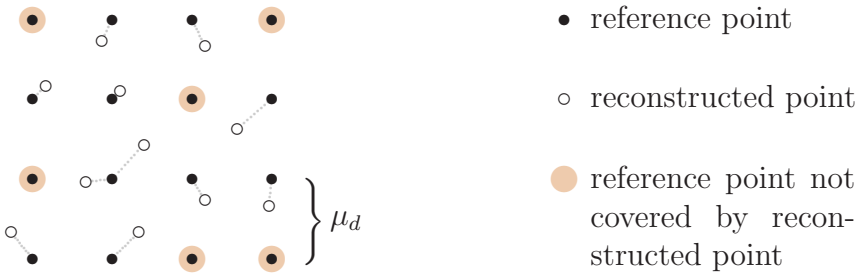


Figure 6.2: Coverage analysis using a homogeneous point cover of the CAD model as reference points. Some reference points are covered by reconstructed points, some are not. For the depicted example, 10 of 16 reference points are covered, hence c = 62.5%. Further explanations are given in Section 6.1.3.2.

The actual computation of the coverage c makes use of the mean distance

μ_d of the reconstructed point set, which we determine first. Employing the reference CAD model, we derive a homogeneous point cover from the CAD surface featuring the same mean distance μ_d, cf. Figure 6.2. Then, we refer to the set of reconstructed 3D points and, for each of these points, mark the next neighbor within the homogeneous point cover of the reference model. This is to say we mark reference points that are *covered* by at least one reconstructed point. We get the coverage c as the ratio of marked reference points related to all reference points.

The presented completeness benchmark raises lots of what-if question, at least since the mathematical foundation inherently is a statistical one for the reasons listed above. We first want to comprehend the main aspects and then answer some likely questions in order to justify this kind of benchmarking and underline its reasonability. We use aligned, finite, discrete point sets to compute two numerals expressing completeness, the coverage c and mean distance μ_d of neighboring reconstructed points. We do not use any user threshold. The homogeneously distributed point cover of the reference CAD model is provided by statistical means that are implemented, for example, in *Geomagic*. Using this reference point cover, the coverage c penalizes gaps within the reconstruction that are large compared to μ_d. In the same way, the coverage yields low values for reconstructions featuring dense clusters of points, since these lead to gaps at other positions. The coverage thus measures the distributional homogeneity of reconstructed points with respect to the reference CAD surface of the NBV test object. This measure is independent of the number of reconstructed points and the point density, respectively. The density, or resolution, of the reconstruction is given in terms of the mean distance μ_d between nearest neighbors within the set of reconstructed 3D points. Now, the reader may state the following questions.

- *What if my reconstruction features lots of 3D points that are concentrated along corners and edges of the technical NBV test object? We should consider this complete since the surface between the points is mostly planar.*

  The benchmark assumes a general reconstruction method that does not state assumptions about the object shape. Hence, the surface regions between edges and corners are evaluated as unknown to the reconstruction method, i. e. gaps. The described 3D reconstruction will get a low coverage rating.

- *What if I cause the reconstruction method to produce many points in a specific surface region that provides accurate 3D points? Would that not push the accuracy rating?*

  Yes, it would. However, this necessarily decreases the coverage rating.

- *What if the reconstruction procedure produces a non-discrete representation like a surface triangulation?*

  In this specific case we suggest to use the corner points, i. e. the vertices of the surface triangulation, for the evaluation.

- *And what about continuous, parametric surfaces?*

  We can derive arbitrary discrete 3D surface points as a basis for the benchmark. However, this arbitrary number P of points does *not* entail an arbitrary high overall benchmark rating. Given a continuous, parametric surface, we define the number P as necessary user input. Applying the same method as for the coverage analysis, we derive P discrete surface points from the reconstruction and apply the described benchmark to these reconstructed points. For an increasing number P of reconstructed points, differences between the reference and reconstructed surfaces cause decreasing coverage and decreasing accuracy, so this is not a way to cheat.

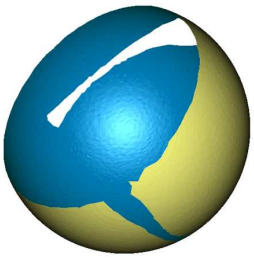
6.1.3.3 The Overall Benchmark and an Example of Application

The overall benchmarking procedure consists of applying the candidate planning method to the NBV test object and evaluating the 3D reconstruction in the way presented above. Finally, we yield the benchmark

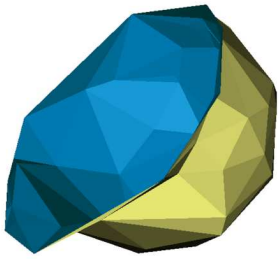
$$\mathcal{B} = [c, \mu_d, \mu_e, v] \tag{6.1}$$

providing the reached values of the coverage c, the mean distance μ_d of point and its next neighbor within the set of reconstructed points, the mean error μ_e, and the number of views v. This can be done for the whole NBV test object as well as for the particular object details.

As an example of application, we apply the proposed benchmark scheme to the planning results of two different methods. In [Munkelt et al., 2006], the authors employ a high-accuracy fringe-projection system and perform model-based view planning. On the other hand, the authors of [Wenhardt



(a) 3D reconstruction computed by the planning method [Munkelt et al., 2006].



(b) 3D reconstruction yielded by the planning method [Wenhardt et al., 2006].

Figure 6.3: Different reconstructions of the negative half sphere, which is a detail of the NBV test object. The surface meshes are shown for better visualization only, while the benchmark uses not more than the 3D points.

et al., 2006] implement view planning for accuracy optimization by means of a passive camera. Hence, the methods strongly differ with respect to the reconstruction technology and methodology. Figure 6.3 illustrates the negative half spheres produced by either methods. We provide the results of the benchmark using the basic setup of the NBV object in Table 6.1. The pictures of the reconstructions show that the fringe-projection system achieves a strongly larger number of points and better accuracy. These facts are reflected by the numbers in Table 6.1. The coverage reached by the fringe-projection system may seem a little low. While Figure 6.3(a) shows only a small gap, it is not visible that the scans are overlapping. Inside these areas of overlap, the point density is higher than outside, which is caused by shifted sampling positions. These inhomogeneities together with the visible gap produce the reasonable coverage value of 71% for the fringe-projection system. Still, this coverage is higher than the one of 53% achieved by the system using a passive camera. The sampling resolutions within both 3D reconstructions are represented by the mean distances between a reconstructed point and its next neighbor in the reconstruction, which are 0.39mm for the fringe-projection system and 7.98mm for the passive system. This severe difference is mainly caused by the used hardware and also qualitatively visible in Figure 6.3. The reconstruction accuracy is reflected by the mean

Table 6.1: Benchmark according to (6.1). For a particular object detail, the specific number of views is hard to determine, so we omit it. The basic benchmark is given by taking the average of the whole basic object and the average of the details. The numbers reflect the quality differences shown in Figure 6.3.

object detail	planning according to [Munkelt et al., 2006]	planning according to [Wenhardt et al., 2006]
tripod	[69%, 0.34mm, 0.10mm]	[58%, 6.59mm, 1.43mm]
sinusiodal face	[73%, 0.41mm, 0.08mm]	[47%, 7.18mm, 1.58mm]
notch	[75%, 0.51mm, 0.18mm]	[51%, 12.70mm, 5.62mm]
negative half sphere	[77%, 0.55mm, 0.12mm]	[57%, 7.34mm, 2.42mm]
whole basic object	[70%, 0.31mm, 0.11mm, 8]	[52%, 7.52mm, 1.50mm, 10]
basic benchmark	[71%, 0.39mm, 0.12mm, 8]	[53%, 7.98mm, 2.13mm, 10]

point errors of 0.12mm for the fringe-projection system and 2.13mm for the passive system. Finally, the benchmark tells us that the fringe-projection system used eight views to yield the reconstruction, while the passive system recorded ten views.

The illustrated example yields clear and reasonable statements on the different reconstruction qualities, since the measured quantities are all dominated by the fringe-projection system. Other questions come from concerning the cost factor and applicability of such a system, which are not reflected by the benchmark. On a fixed system, the benchmark allows to assess the results of different planning methods. For systems and methods that achieve qualities closer to each other, the benchmark helps to focus on particular quality criteria that are important to the task at hand.

6.2 The Modular Online Planning System

Now that we are capable of assessing the performances of different planning methods quantitatively, we find this a reasonable starting position to think about actual view planning. In doing so, we are faced with several theoretical and practical challenges, which we partially formulated in the previous chapters. In Section 1.1.2 we presented a formalization of the view planning problem revealing various aspects of this topic. Likewise, the literature review in Chapter 2 underlines the heterogeneous nature of view planning and the particular planning goals. Additionally, we argued that it is one reason-

able way to focus on using a passive camera in Section 1.1. In consequence, we recognize the following main challenges.

(a) *View planning is not a fixed, static problem.* As we outlined, view planning is a high-level approach to solve specific, user-defined tasks. While solving all specific view planning formulations by means of one algorithm appears to be impossible, a proper view planning method still should provide generic means to address multiple instances of the view planning problem.

(b) *Neither the view planning environment nor the hardware are fixed.* Just as the actual view planning task is defined by the user, so the used hardware and the application environment are not fixed for all view planning tasks. This raises the claim that the methodology should be invariant with respect to these variables.

(c) *Using a passive camera constrains the planning procedure.* For this work we focus on, but do not restrict the presented methods to the sensor type of a passive camera. Assuming that we employ feature tracking to solve the correspondence problem, the camera motion is bound to the assumption of small baselines. Hence, we cannot jump to arbitrary positions and directly use the resulting images for reconstruction and planning.

(d) *Using a passive camera constrains completeness.* All kinds of sensors are given technical boundaries that constrain the quality of the 3D reconstruction. However, a passive camera additionally requires optical surface structure to solve the correspondence problem. This claim restricts the maximal number of points and their positions on the surface. Keeping these facts in mind, completeness issues are hard to address by means of this sensor type.

(e) *Some challenges are contradictory.* On the one side, we want to use each kind of prior knowledge to improve the reconstruction, which includes, for instance, the physics of the scanner hardware. On the other side, a generic method cannot use knowledge of one specific sensor type. We have to resolve this contradiction by a clear focus.

(f) *In terms of application, we like to use a fire-and-forget solution.* The most comfortable way to use a system is to push a button and get

results after some time. Likewise, this kind of execution seems most appealing for view planning.

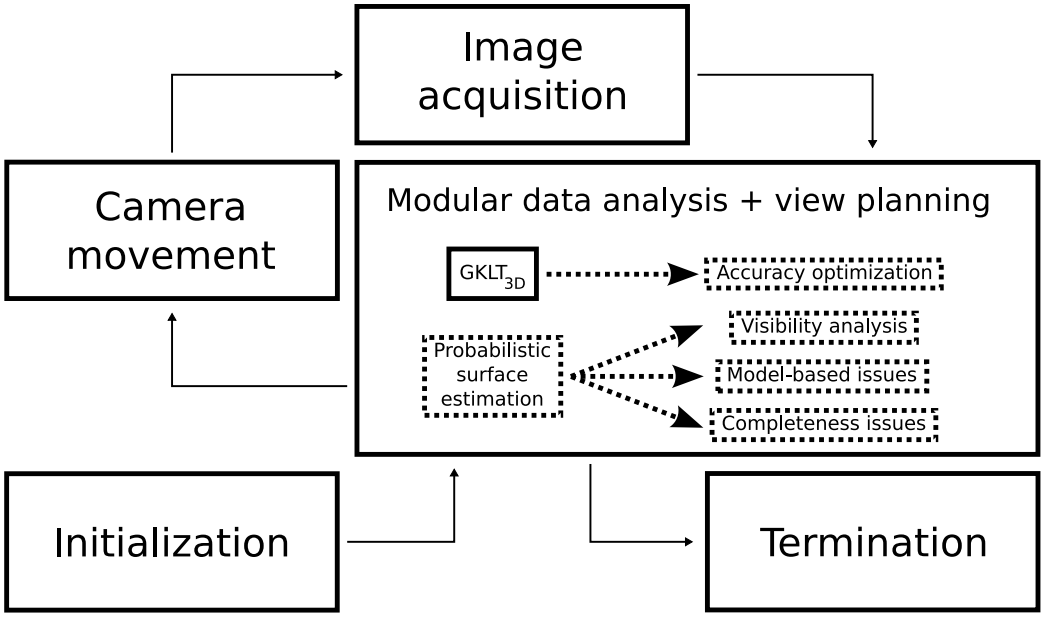


Figure 6.4: A specification of the proposed modular approach to view planning. The module of $GKLT_{3D}$ tracking is the only mandatory one, since there is no 3D reconstruction without this module. It enables accuracy optimization if wanted. Depending on the planning goals and constraints, the algorithm may employ further modules for surface estimation, visibility analysis, model-based tasks, and completeness issues.

We answer the formulated challenges by presenting a modular approach to online next-best-view planning using a passive camera as illustrated in Figure 6.4. The basic cycle of the planning algorithm originates from feature tracking for 3D reconstruction, which we perform in terms of $GKLT_{3D}$ tracking as presented in Chapter 4. So, one planning cycle consists of taking an image, running GKLT tracking and the further modules needed to address the particular planning goals and constraints, and adapting the camera parameters accordingly. At some well-defined point depending on the planning goals and constraints, the planning algorithm terminates. We invoke the planning modules after each single image taken, hence each view is a

planned view, and there are no simulations of large movements by simple step-wise interpolation of camera parameters. This is to say we respect the assumption of small baselines within our planning approach. In comparison to Figure 1.4, we picture a combined complex for data analysis and planning in Figure 6.4 in order to clarify the relations between the modules. Still, we separate the modules for data analysis and the ones for view planning in the sense that the former provide the data basis for the latter.

Now, we reply to the above challenges in detail.

(A) Despite the various formulations of specific view planning tasks, we identify reconstruction accuracy and completeness as the two main aspects that view planning shall optimize. We present planning modules treating both of these issues. $\text{GKLT}_{3\text{D}}$ tracking provides 3D point estimates together with 3D covariances, which endow the data basis for the accuracy optimization presented in Section 6.3. With respect to completeness issues, we employ the probabilistic surface estimation proposed in Section 6.4. The resulting non-discrete surface, given as a 3D surface triangulation, allows to assess the completeness of a 3D reconstruction, to perform visibility analysis as in Section 6.5, and to bridge the gap towards model-based view planning as shown in Section 6.6. The module of $\text{GKLT}_{3\text{D}}$ tracking is the only mandatory one, since it establishes the basic reconstruction of 3D points. Hence, we introduce a set of modules for data analysis and view planning that allow a plug-in architecture of the whole view planning method. By this means we address a large variety of specific view planning tasks.

(B) The proposed planning modules employ transferable methodologies. $\text{GKLT}_{3\text{D}}$ tracking does not depend on a specific passive camera. Likewise, we present an accuracy optimization that has been shown to be transferable to a fringe-projection system in [Munkelt et al., 2010]. Further, the 3D coarse registration and 3D surface features presented in Chapter 5, which are the basis for detecting geometric primitives and other model-based tasks, are robust with respect to most differing surface qualities produced by different scanning systems.

(C) Using a passive camera, we accept that we are not able to directly jump to arbitrary camera positions. On the contrary, we include the assumption of small baselines into the planning cycle as shown for the respective modules.

(D) Using a passive camera, the reachable completeness is bound by the optical surface structure of the target object. Therefore, we do not deal with pure completeness optimization in the context of this work. Nevertheless, the probabilistic surface estimation offers the data basis to address related tasks. For instance, another method like [Dey and Goswami, 2003] may use the surface estimation to produce a water-tight model.

(E) To avoid contradictions, we focus on view planning using a passive camera without explicitly modeling specific sensor physics. We hence propose methodologies that are transferable to environments other than a passive camera mounted on a robotic arm, as done in [Munkelt et al., 2010] for accuracy optimization. However, our approach does not forbid to be further extended with respect to specific claims.

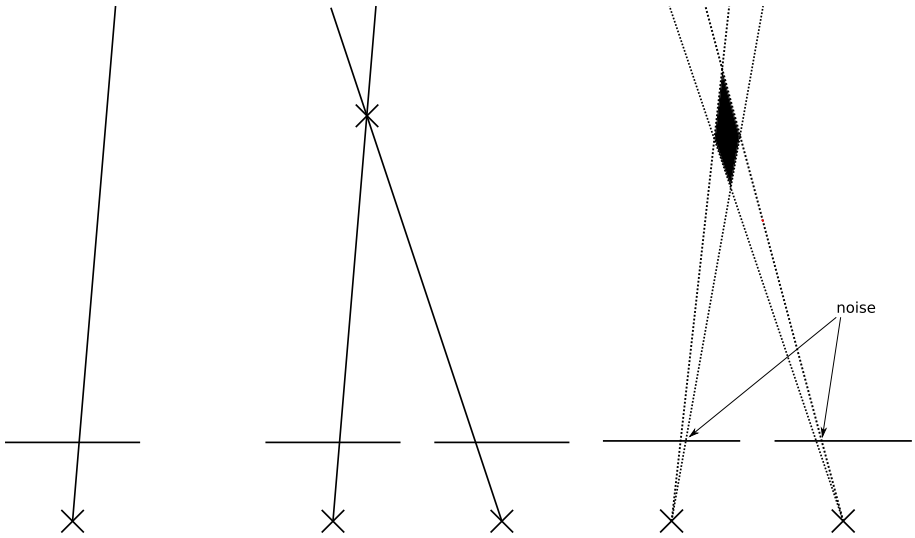
(F) To push a button once and yield the wanted results after some time is a noble ideal of software engineering. However, this simplification is already too strong for most of the all-day practical applications, if we just think about user input. Further, this work is not about software engineering, but develops methodologies to tackle view planning as a whole as well as specific view planning tasks. Nonetheless, future work can use the modules presented in this chapter to derive a user-friendly, practical software solution for view planning.

6.3 Accuracy Optimization Using an Extended E-criterion

The essence of accuracy optimization is the reduction of the reconstruction uncertainty, which is a dual proceeding being better to handle. For this module we use the 3D covariance data of a 3D point estimate and derive a closed-form, optimal solution for the camera motion, cf. [Trummer et al., 2010a]. First, we investigate the reconstruction uncertainty using triangulation-based 3D point reconstruction, and we derive the extended E-criterion to minimize this uncertainty. Next, we describe the optimal camera motion with respect to accuracy optimization and the extended E-criterion.

6.3.1 Assessing Triangulation Uncertainty

In the first place, we want to characterize the 3D uncertainty of a point estimate achieved on the basis of triangulation using rays of sight. The Figures 6.5 and 6.6 illustrate the underlying setup of passive cameras mapping the observed point to image points, which are afflicted by noise. We explicitly refer to this underlying setup, and not to the DLT algorithm used in this work, cf. Section 4.4.3. The motivation of this approach lies in the fact that the DLT algorithm minimizes an algebraic error in the projective space, which has no geometric meaning and makes it hard to infer on the quality of a camera position.

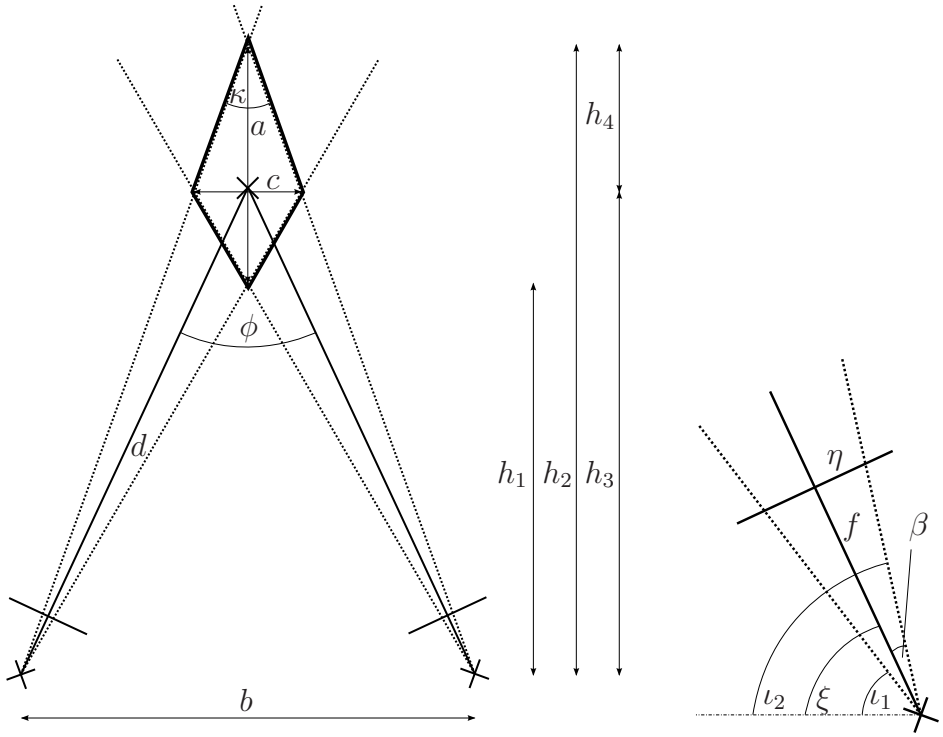


(a) A single passive camera and the ray of sight of an image point. 3D uncertainty is infinite along the ray of sight.

(b) Theoretical setup of triangulation as in Figure 3.4. The two rays of sight belong to the same world point, but originate from different cameras. Uncertainty is not regarded.

(c) Triangulation setup regarding uncertainty. Uncertainty of the image points causes uncertainty of the 3D estimate (black region).

Figure 6.5: An illustration of the uncertainty of a 3D point estimate using passive cameras and triangulation. The noise assumption is a disturbance of the 2D image points.



(a) The basic setup of two identical cameras observing one point. Noise of the image points causes a kite-shaped uncertainty region regarding the observed point. The utmost noisy rays of sight (dashed lines) deviate from the true ones (solid lines).

(b) A close-up of one camera and the angles relating baseline, true ray of sight, and noise bounds.

Figure 6.6: A simple 2D geometric model investigating triangulation uncertainty, and the symbols used in Section 6.3.1. Based on noisy image points, the position of the observed point is uncertain inside the characterized region. Specification of Figure 6.5(c).

Goal. Assessing triangulation uncertainty by probabilistic means yields expressions that are hard to handle. For instance, assuming Gaussian noise on the positions of image points in a fixed camera setup provides a probability density function of the 3D point position that is difficult and non-Gaussian. Instead, we utilize the simple 2D geometric model in Figure 6.6 that does not refer to distributional aspects. We model extremal noise effects and thus deal with a geometric worst-case scenario. In this sense, we aim to characterize the worst case for the point reconstruction and to provide an upper bound for this worst case. In detail, we want to quantify the maximal diameter of the region, cf. Figure 6.6(a), in which the observed point lies. Surely, this is

a confined, theoretical consideration, but the practical relevance will be outlined in the following, and we are going to reinforce an old photogrammetric fact, finally.

Assumptions.

- We do not regard distributional aspects but render a worst-case scenario of noisy image points. This is to say that we refer to the worst deviations of noisy image points with respect to the true image points.

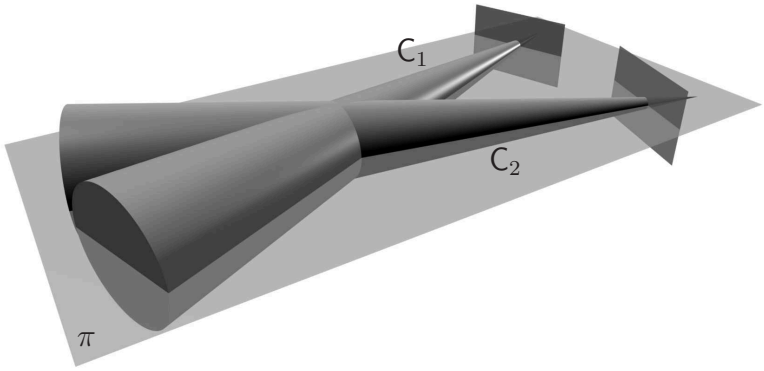
- The geometric setup in Figure 6.6 is placed in the 2D plane π containing the camera centers and the true 3D point being observed. While this is a severe simplification, we argue by means of Figure 6.7 that this plane also contains the maximal diameter of the 3D uncertainty region given the specific camera setup and a reasonable noise ratio.

- The camera centers are at distance b to each other. Each camera center features the same distance d to the true position of the observed point. The noise is given as the worst case in terms of a maximal deviation η of the image point regarding its true position, given that both cameras have the same focal length f and the image planes are perpendicular to the respective rays of sight. So, without loss of generality, each camera maps the observed point along its respective optical axis. Another camera rotation and focal length would just call for an adaptation of the noise description. The noise η and the focal length f determine the angle β.

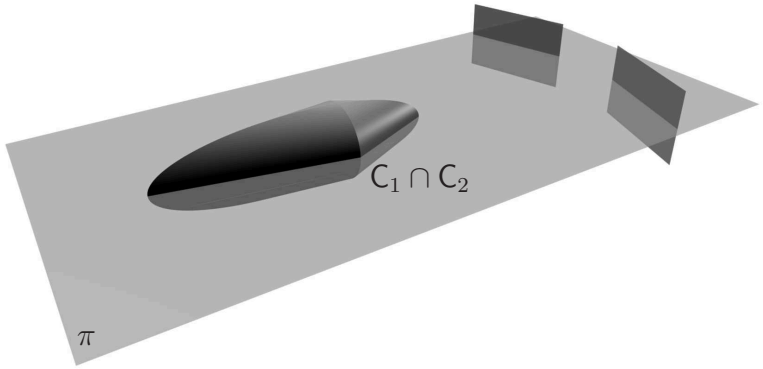
- The angle ϕ describes the inclination of the true rays of sight. Given the identical distances d between the observed point and the camera centers, ϕ codes the same information as the baseline distance b.

- In this specific setup, the 2D region of uncertainty for the observed point is a kite, the diagonals of which have lengths a and c. The maximum of a and c is the maximal diameter of the uncertainty region for the reconstruction in this worst-case scenario. Hence, the maximum of a and c is the smallest upper bound for the error distance of a reconstruction of the observed point, and it is the quantity we want to characterize.

- We denote supportive symbols. These are the angles κ as an internal angle of the uncertainty kite, ξ as the angle between a ray of sight and the connection of the camera centers, and $\iota_{1,2} = \xi \mp \beta$. We further use the heights h_1 to h_4.



(a) Radial noise bounds for the image points lead to infinite 3D cone volumes C_1 and C_2 describing the 3D uncertainty volumes of each back-projection. Plane π contains the camera centers and the true 3D point being observed.



(b) The 3D volume intersection of C_1 and C_2 represents possible results of the noisy 3D estimate, i. e. its 3D uncertainty volume. Given the particular setup, the maximal diameter of this uncertainty volume lies in plane π, particularly in $\pi \cap \mathsf{C}_1 \cap \mathsf{C}_2$, which is the uncertainty kite in Figure 6.6(a).

Figure 6.7: A visual fortification of the assumption that the maximal diameter of the 3D uncertainty volume equals the maximal diameter of the 2D uncertainty kite in Figure 6.6(a).

Parity of kite diagonals a **and** c**.** Given the parametric camera setup and noise model depicted above, we want to analyze the maximal diameter of the

uncertainty region, which is the maximum of a and c. The maximal diameter depends on the noise level and the relative camera pose. The noise level is expressed in terms of the ratio of η and f, which is comprehended by the angle

$$\beta = \arctan\left(\frac{\eta}{f}\right). \tag{6.2}$$

For the relative camera pose, a change of the camera-point distance d finds, ceteris paribus, a monotone expression reaching from small diameter for small values of d towards an infinite diameter for d at infinity. This parameter is thus not interesting, and we keep it fixed in the following. Another parameter change that also influences the baseline b is adapting the angle ϕ between the rays of sight. For the general case of a constant $\beta > 0$ and a constant $d > f$, we utilize Figure 6.6(a) and find that different values of ϕ have severe influence on the kite diagonals a and c. There even exists a lower breakdown value of ϕ in the sense that a will be infinitely large. If we consider the whole range $\phi \in [0, 2\pi]$, we find symmetric configurations for $\phi \in [0, \pi]$ and $\phi \in]\pi, 2\pi]$. We further observe that for small ϕ $a > c$, and $a < c$ for ϕ close to π. Hence, the ratio $\frac{a}{c}$ crosses 1 for $\phi \in [0, \pi]$. Our first step will be to compute the angle ϕ_p that causes the diagonals a and c to be at parity. Next, we characterize ϕ_p with respect to the noise level β. Finally, we show that for ϕ_p the equal diagonals provide a smallest upper bound of the maximal diameter of the uncertainty region.

We start by preparing expressions of the symbols in Figure 6.6 just using

given parameters. We find

$$b = 2d \sin\left(\frac{\phi}{2}\right) \tag{6.3}$$

$$\xi = \frac{\pi - \phi}{2} \tag{6.4}$$

$$\iota_{1,2} = \frac{\pi - \phi}{2} \mp \beta \tag{6.5}$$

$$\kappa = \pi - 2\iota_2 = \phi - 2\beta \tag{6.6}$$

$$h_1 = \frac{b}{2} \tan \iota_1 = d \sin\left(\frac{\phi}{2}\right) \tan\left(\frac{\pi - \phi}{2} - \beta\right) \tag{6.7}$$

$$h_2 = \frac{b}{2} \tan \iota_2 = d \sin\left(\frac{\phi}{2}\right) \tan\left(\frac{\pi - \phi}{2} + \beta\right) \tag{6.8}$$

$$h_3 = \frac{b}{2} \tan \xi = d \sin\left(\frac{\phi}{2}\right) \tan\left(\frac{\pi - \phi}{2}\right) \tag{6.9}$$

$$h_4 = h_2 - h_3 = d \sin\left(\frac{\phi}{2}\right) \left(\tan\left(\frac{\pi - \phi}{2} + \beta\right) - \tan\left(\frac{\pi - \phi}{2}\right)\right). \tag{6.10}$$

We aim at ϕ_p such that $a(\phi_p) = c(\phi_p)$, if we consider the lengths of the diagonals a and c as functions depending on ϕ. The functional expressions are

$$\begin{aligned} a(\phi) &= h_2(\phi) - h_1(\phi) \\ &= d \sin\left(\frac{\phi}{2}\right) \left(\tan\left(\frac{\pi - \phi}{2} + \beta\right) - \tan\left(\frac{\pi - \phi}{2} - \beta\right)\right) \end{aligned} \tag{6.11}$$

and

$$\begin{aligned} c(\phi) &= 2h_4(\phi) \tan\left(\frac{\kappa(\phi)}{2}\right) \\ &= 2d \sin\left(\frac{\phi}{2}\right) \tan\left(\frac{\phi}{2} - \beta\right) \left(\tan\left(\frac{\pi - \phi}{2} + \beta\right) - \tan\left(\frac{\pi - \phi}{2}\right)\right). \end{aligned} \tag{6.12}$$

Noting that ϕ occurs as argument of the tan-function in various forms, we introduce

$$x = \tan\left(\frac{\pi - \phi}{2} + \beta\right) \tag{6.13}$$

in order to abbreviate notation and to express the dependencies between the various occurrences. For the further occurrences of $\tan(z\phi + \ldots)$, $z \in \mathbb{R}$, we find

$$\tan\left(\frac{\pi-\phi}{2}-\beta\right) = \tan\left(\frac{\pi-\phi}{2}+\beta-2\beta\right) = \frac{x-\tan(2\beta)}{1+x\tan(2\beta)}, \tag{6.14}$$

$$\tan\left(\frac{\pi-\phi}{2}\right) = \tan\left(\frac{\pi-\phi}{2}+\beta-\beta\right) = \frac{x-\tan\beta}{1+x\tan\beta}, \tag{6.15}$$

$$\begin{aligned}\tan\left(\frac{\phi}{2}-\beta\right) &= -\tan\left(-\frac{\phi}{2}+\beta\right) = -\tan\left(\pi-\frac{\phi}{2}+\beta-\pi\right)\\ &= -\frac{\sin\left(\pi-\frac{\phi}{2}+\beta-\pi\right)}{\cos\left(\pi-\frac{\phi}{2}+\beta-\pi\right)} = -\frac{\cos\left(\frac{\pi}{2}-\frac{\phi}{2}+\beta-\pi\right)}{-\sin\left(\frac{\pi}{2}-\frac{\phi}{2}+\beta-\pi\right)}\\ &= \frac{1}{\tan\left(\frac{\pi}{2}-\frac{\phi}{2}+\beta-\pi\right)} = \frac{1}{\frac{x-\tan\pi}{1+x\tan\pi}} = \frac{1}{x}.\end{aligned} \tag{6.16}$$

Now, we are prepared to set

$$a(\phi) = c(\phi). \tag{6.17}$$

Using (6.11) and (6.12), we reduce $d\sin\left(\frac{\phi}{2}\right)$, apply the substitutions (6.13) to (6.16), and yield

$$\frac{2}{x} = \frac{x-\frac{x-\tan(2\beta)}{1+x\tan(2\beta)}}{x-\frac{x-\tan\beta}{1+x\tan\beta}} = \frac{\tan(2\beta(1+x\tan\beta))}{\tan\beta(1+x\tan(2\beta))}, \tag{6.18}$$

$$0 = x^2 + \frac{\tan(2\beta)-2\tan\beta\tan(2\beta)}{\tan\beta\tan(2\beta)}x - \frac{2}{\tan(2\beta)} \tag{6.19}$$

$$= x^2 + \left(\frac{1}{\tan\beta}-2\right)x - \frac{2}{\tan(2\beta)}, \tag{6.20}$$

$$x_{1,2} = 1 - \frac{1}{2\tan\beta} \pm \sqrt{\left(\frac{1}{2\tan\beta}-1\right)^2 + \frac{2}{\tan(2\beta)}}. \tag{6.21}$$

Employing quadrant relations we conclude that

$$\phi_p = \phi_1 = 2\left(\frac{\pi}{2} - \arctan(x_1) + \beta\right), \tag{6.22}$$

while ϕ_2 describes the mirror-symmetric solution. Hence, we have determined the angle ϕ_p such that the respective relative camera pose and the given noise ratio β cause the kite diagonals to be equal,

$$a(\phi_p) = c(\phi_p) = l. \tag{6.23}$$

Characterizing ϕ_p. An interesting fact is that ϕ_p has the only parameter β, the distance d has no influence. This means that the angle at which a and c are equal does only depend on the noise level. What else can we say about ϕ_p? Example calculations show that ϕ_p is close to $\frac{\pi}{2}$, if the noise level β is within limits that are practically relevant. And what exactly happens for small values of β? To investigate this question, we consider β approaching zero, i. e. we consider

$$\phi_p^* = \lim_{\beta \to 0} \phi_p. \tag{6.24}$$

Referring to (6.22), we examine

$$\lim_{\beta \to 0} x_1 = \lim_{\beta \to 0} 1 + A \tag{6.25}$$

using

$$A = -\frac{1}{2\tan\beta} + \sqrt{\left(\frac{1}{2\tan\beta} - 1\right)^2 + \frac{2}{\tan(2\beta)}} \tag{6.26}$$

$$= \frac{-1 + \sqrt{1 + 4\tan^2\beta - 4\tan^3\beta}}{2\tan\beta}. \tag{6.27}$$

Using $m = \tan\beta$ and $A = A(m)$, we aim to find

$$\lim_{m \to 0} A(m) = \lim_{m \to 0} \frac{g_1(m)}{g_2(m)}. \tag{6.28}$$

Since $A(0)$ yields the irregular expression $\frac{0}{0}$, we apply the rule of l'Hospital,

$$\lim_{m \to 0} \frac{g_1(m)}{g_2(m)} = \lim_{m \to 0} \frac{g_1'(m)}{g_2'(m)} = \lim_{m \to 0} \frac{\frac{1}{2}\left(1 + 4m^2 - 4m^3\right)^{-\frac{1}{2}}\left(8m - 12m^2\right)}{2} \tag{6.29}$$

$$= \lim_{m \to 0} \frac{8m - 12m^2}{\sqrt{1 + 4m^2 - 4m^3}} = \frac{0}{1} = 0. \tag{6.30}$$

Tracing back the whole chain of derivation, we see that A approaches zero if m and hence also if β does so, thus

$$\lim_{\beta \to 0} x_1 = 1. \tag{6.31}$$

Applying this result to (6.22), we finally achieve

$$\begin{aligned} \phi_p^* &= \lim_{\beta \to 0} 2 \left(\frac{\pi}{2} - \arctan(x_1) + \beta \right) = 2 \left(\frac{\pi}{2} - \arctan(1) + 0 \right) \\ &= 2 \left(\frac{\pi}{2} - \frac{\pi}{4} \right) \\ &= \frac{\pi}{2}. \end{aligned} \tag{6.32}$$

What we have shown here is the asymptotic behavior of ϕ_p for small values of β. For the noise level β approaching zero, the angle ϕ_p between rays of sight that causes equal kite diagonals of the uncertainty region approaches 90°.

The maximal diagonal with respect to ϕ and β. Up to now, we are able to compute an angle ϕ_p, given the noise level β, such that accordingly placed cameras produce an uncertainty region that is a kite with equal diagonal lengths $a = c = l$. We further know that ϕ_p is close to $\frac{\pi}{2}$ for small values of β. What we actually want to characterize is the maximal diagonal with respect to ϕ and β. In the following we sketch the proof that the camera configuration featuring ϕ_p also features the smallest maximum of a and c. In other words, if we aim to minimize the maximal diameter of the uncertainty region, which minimizes the reconstruction error, then we have to place the camera centers such that $\phi = \phi_p$. Hence, we set the assertion that

$$\forall \phi, \phi \in [0, \pi], \phi \neq \phi_p : a(\phi) > l \vee c(\phi) > l \tag{6.33}$$

using the configuration in Figure 6.6. By definition is $a(\phi_p) = c(\phi_p) = l$. An equivalent assertion is that

$$\forall \phi, \phi \in [0, \pi], \phi \neq \phi_p : a(\phi) + l \vee c(\phi) > 2l. \tag{6.34}$$

We use the latter expression, define a function

$$g(\phi) = a(\phi) + c(\phi), \tag{6.35}$$

and argue that $g(\phi)$ has exactly one minimum in $[0, \pi]$ that lies at ϕ_p. Looking at the expression for $a(\phi)$ in (6.11) and for $c(\phi)$ in (6.12) conveys an impression of the effort necessary to derive an analytic solution for this basic task. With the help of `Mathematica` we confirm the described properties of $g(\phi)$, and we find that (6.33) is actually true. This provides us with the following information. There is a an angle ϕ_p, which corresponds to a baseline b_p, such that the kite diagonals a and c are equal. The value of ϕ_p depends on the noise level β and is close to $\frac{\pi}{2}$. If $\phi < \phi_p$, then $a > c$ and $a > l$. If $\phi > \phi_p$, then $a < c$ and $c > l$. Hence, the angle ϕ_p between the rays of sight features the minimum of the maximal diagonal length with respect to ϕ, which is $a(\phi_p) = c(\phi_p) = l$.

Conclusion. Employing a simple 2D geometric uncertainty model, we describe the worst reconstruction error based on the worst error affecting the image points in a triangulation scenario. We analyze the upper bound of the reconstruction error in terms of the maximal diagonal of the uncertainty region. Minimizing the upper bound with respect to the angle between the rays of sight, i. e. with respect to the camera baseline, yields that the noted angle should be close to 90° depending on the noise level. To give an example, $\beta = 0.573°$, which comes from $f = 0.1$ and $\eta = 0.001 = \frac{1}{100} f$, produces an optimal $\phi_p = 90.582°$. For smaller values of β, ϕ_p approaches 90°. In the following we boldly say that the optimal ϕ_p equals 90°, whereat we abbreviate the derived facts and accept an inaccuracy that seems to be meaningless in practice.

For practical application, the statement derived above suggests to establish perpendicular rays of sight in order to minimize reconstruction uncertainty. While this is really intuitive and also well-known from practical photogrammetry, we underlined this fact by a basic theoretical foundation.

6.3.2 Deriving an Extended E-criterion and Implementing a Closed-form, Optimal Solution

The previous section deals with an optimal adjustment of relative camera positions in order to minimize the reconstruction uncertainty. As the main aspect, the angle between the rays of sight should be 90° to reach this goal. Regarding the view planning cycle in Figure 6.4, we have to take into account that there are more than two camera positions and that the baseline is

bound by the needs of feature tracking. We transfer the knowledge derived in Section 6.3.1 by taking a more abstract look at Figure 6.5. For one camera, the ray of sight of an image point describes a 3D estimate featuring an infinite uncertainty along the same ray. Exactly this 3D uncertainty provides the connection to a more general reconstruction setup. Given that we have available an actual 3D estimate of the observed point together with the dominant direction of the 3D uncertainty for this estimate, we apply the result of the previous section and adjust the next camera position in a way that establishes a right angle between the dominant uncertainty direction and the new ray of sight. The above derivation tells us that, respecting the stated assumptions, we thus minimize the uncertainty of the 3D point estimate.

Returning to the view planning approach presented in this work, we again emphasize the application of GKLT feature tracking presented in Chapter 4. One property of the final GKLT tracking is the concurrent, robust 3D estimation of a tracked feature. The method provides a robust estimate $\widehat{\mathbf{P}}_s$ of the 3D feature position $\mathbf{P}$ after each tracking step $s > 0$, where $s = 0$ is the index of the feature initialization frame. For a sequence of observations $\langle \widehat{\mathbf{P}}_s \rangle_{s=1,2,\dots,n}$ with $n > 1$, we determine the covariance matrix of the 3D position of feature $\mathbf{P}$,

$$\mathbf{\Sigma}_{\widehat{\mathbf{P}}_n} = \left(\sigma_{ij}^{\widehat{\mathbf{P}}_n} \right)_{1 \leq i,j \leq 3} \tag{6.36}$$

with the covariances $\sigma_{ij}^{\widehat{\mathbf{P}}_n}$ of the 3D coordinates. Let

$$\mathbf{\Sigma}_{\widehat{\mathbf{P}}_n} \mathbf{V}_{\widehat{\mathbf{P}}_n} = \mathbf{\Lambda}_{\widehat{\mathbf{P}}_n} \mathbf{V}_{\widehat{\mathbf{P}}_n} = \begin{pmatrix} \lambda_1^{\widehat{\mathbf{P}}_n} & 0 & 0 \\ 0 & \lambda_2^{\widehat{\mathbf{P}}_n} & 0 \\ 0 & 0 & \lambda_3^{\widehat{\mathbf{P}}_n} \end{pmatrix} \left(\mathbf{v}_1^{\widehat{\mathbf{P}}_n} \mathbf{v}_2^{\widehat{\mathbf{P}}_n} \mathbf{v}_3^{\widehat{\mathbf{P}}_n} \right) \tag{6.37}$$

denote the eigen decomposition of the matrix $\mathbf{\Sigma}_{\widehat{\mathbf{P}}_n}$ with eigenvalues $\lambda_1^{\widehat{\mathbf{P}}_n} \geq \lambda_2^{\widehat{\mathbf{P}}_n} \geq \lambda_3^{\widehat{\mathbf{P}}_n}$ and corresponding eigenvectors $\mathbf{v}_1^{\widehat{\mathbf{P}}_n}$. From this point, we omit the point index $\widehat{\mathbf{P}}_n$, if there is no question about what point estimation is addressed. Now, having a 3D point estimate $\widehat{\mathbf{P}}$ and a corresponding covariance matrix $\mathbf{\Sigma}$, we visualize directional uncertainty by assuming normal distribution. In this case, $\mathbf{\Sigma}$ as in (6.37) defines an equiprobable curve as an ellipsoid with semiaxes along $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. The lengths of the semiaxes are $\lambda_1 \geq \lambda_2 \geq \lambda_3$ up to global scale. By means of the eigenvalues λ_i we are able to identify the main direction of uncertainty, which is $\mathbf{v}_1$. Using the terms at hand, we are able to establish an extended E-criterion for accuracy optimization.

Extended E-Criterion (EEC) Using the notation above, we formulate the extended E-criterion as sensing perpendicular to $\mathbf{v}_1$, which is the direction of the largest uncertainty. As the known E-criterion from statistics [Pukelsheim, 1993], this formulation minimizes the largest eigenvalue λ_1, but additionally incorporates the corresponding eigenvector $\mathbf{v}_1$.

Applying the view planning formalism from Section 1.1.3, the accuracy optimization of this section is similar to the example A in the former section. As in (1.7) to (1.9), the reconstruction data R at least holds the reconstructed 3D points and the according covariance matrices. The goal (1.10) is adjusted to regard the maximal eigenvalue,

$$g : \max_k \widehat{\lambda}_{1k}^{(v+1)} \leq b \tag{6.38}$$

using a target threshold b, and the objective function (1.15) is changed accordingly.

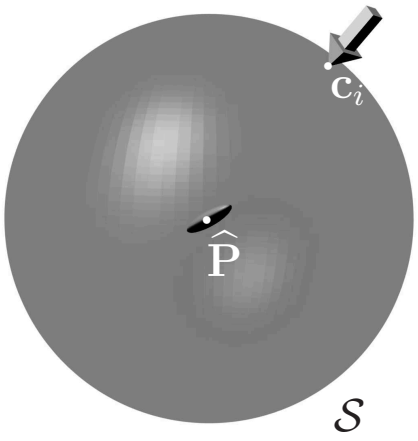


Figure 6.8: Camera center $\mathbf{c}_i$ lies on sphere $\mathcal{S}$. Point estimate $\widehat{\mathbf{P}} = \mathbf{C}$ is shown together with its covariance ellipsoid.

We implement accuracy optimization with respect to the EEC using a spherical motion model for the camera, i. e. we move the camera center on a sphere at which the camera's optical axis is pointed to. For the static motion model, we keep the sphere position and radius constant and place the camera center on the sphere surface by two parameters. For the dynamic motion model, we also vary the sphere position and radius, which provides

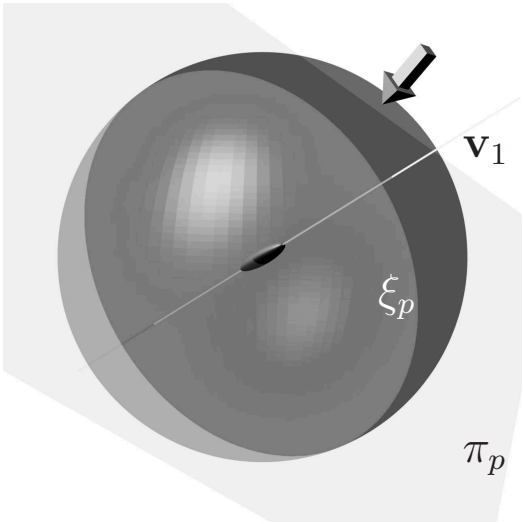


Figure 6.9: Vector $\mathbf{v}_1$ indicates the direction of largest uncertainty of $\widehat{\mathbf{P}}$. Plane π_p through $\widehat{\mathbf{P}} = \mathbf{C}$ is perpendicular to $\mathbf{v}_1$. The intersection of π_p and $\mathcal{S}$ yields the great circle ξ_p on $\mathcal{S}$.

five independent parameters in total. The one parameter missing is fixed by a normalized roll angle of the camera. Hence, the dynamic spherical motion model is practically a general one.

First, let us assume a static motion sphere $\mathcal{S}$ with fixed size and fixed center $\mathbf{C} = \widehat{\mathbf{P}}$. This means that we want to adapt only the two position parameters of $\mathbf{c}_i$ in Figure 6.8 such that the camera takes the shortest path to a position, from where its optical axis is perpendicular to the direction $\mathbf{v}_1$ of the largest uncertainty of $\widehat{\mathbf{P}}$. All camera positions on the sphere fulfilling this condition are situated on a great circle ξ_p of the sphere surface, see Figure 6.9. The great circle ξ_p is given by intersecting the sphere with the plane $\pi_p(\mathbf{X}) : \mathbf{v}_1^{\mathrm{T}}(\mathbf{X} - \widehat{\mathbf{P}}) = 0$, i. e. the plane through $\widehat{\mathbf{P}}$ with normal vector $\mathbf{v}_1$. The shortest way to reach ξ_p, starting at the current camera position $\mathbf{c}_i$, leads along another great circle ξ_s that is perpendicular to ξ_p. The great circle ξ_s lies on the plane $\pi_s(\mathbf{X}) : \mathbf{n}_s^{\mathrm{T}}(\mathbf{X} - \widehat{\mathbf{P}}) = 0$ defined by the two points $\widehat{\mathbf{P}}$ and $\mathbf{c}_i$ and the normal direction perpendicular to $\mathbf{v}_1$. Intersecting the great circles ξ_p and ξ_s yields two intersection points $\mathbf{I}_1$, $\mathbf{I}_2$. Without loss of generality, let $\mathbf{I}_1$ be the intersection point being closer to $\mathbf{c}_i$. Since we use feature tracking and cannot jump directly to $\mathbf{I}_1$, we apply a predefined step size to optimally move the camera on ξ_s towards $\mathbf{I}_1$ as illustrated in Figure 6.10. By this means we achieve a direct, closed-form solution for the point I_1 that indicates the optimal direction for the camera motion with respect to the EEC.

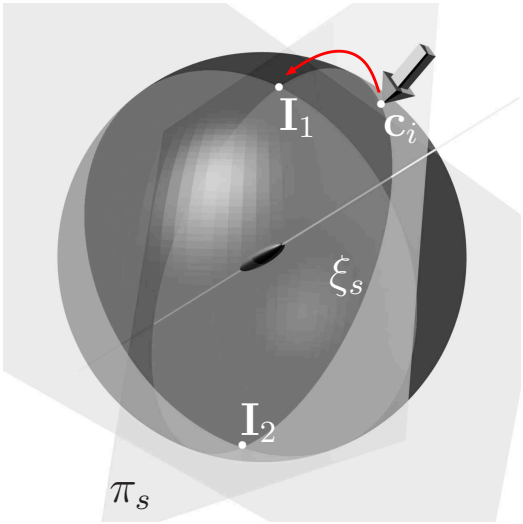


Figure 6.10: The shortest way from $\mathbf{c}_i$ to ξ_p leads along the great circle ξ_s on π_s perpendicular to ξ_p. Intersecting the great circles ξ_s and ξ_p provides points $\mathbf{I}_1$, $\mathbf{I}_2$. Point $\mathbf{I}_1$ is closer to $\mathbf{c}_i$. The camera is moved on ξ_s towards $\mathbf{I}_1$ (red arrow).

Second, we consider the dynamical sphere model for camera motion which provides variable sphere position $\mathbf{C}$ and radius. Further we drop the assumption that the point estimate $\widehat{\mathbf{P}}$ is situated in $\mathbf{C}$. It is important to notice that these positional aspects do not influence the optimal viewing direction. As a consequence, the optimal camera movement on the sphere described above still holds. The only additional matter of the dynamic motion model is to move the camera center $\mathbf{C}$ towards $\widehat{\mathbf{P}}$. Again, the step size of this movement is constrained by the feature tracker.

To conclude this section, we again outline the main aspects of accuracy optimization as performed within this planning module. We start by a theoretical foundation of the well-known photogrammetric advice to have perpendicular rays of sight. A generalization of this idea regarding the main direction of uncertainty allows to formulate a criterion for view planning. We present an extension of the statistical E-criterion and a closed-form, optimal solution for an according camera motion. The main idea for this camera motion is to use the shortest way towards a position such that the new ray of sight is perpendicular to the current main direction of uncertainty. Regarding this kind of accuracy optimization in terms of a planning module, we have a clear interface. The module takes a 3D position and a 3D vector indicating the main direction of uncertainty, and it yields the optimal direction with respect to the EEC to move the camera to. This abstract module interface allows to feed in any 3D position and 3D direction, which enables the user to optimize the accuracy of a 3D point together with covariance data, a set

of points, or any other entity that provides the required input.

6.4 Probabilistic Surface Estimation

In Section 6.2 we identify reconstruction accuracy and completeness as the main quality criteria of a 3D reconstruction. While the previous section deals with accuracy issues, we now tackle reconstruction completeness. As argued above, reconstruction completeness is constrained by the optical surface structure when using a passive camera. Hence, the means to improve completeness are also constrained. Due to these constraints we restrict this module to serve as a supporting module producing a probabilistic, non-discrete surface estimation. By this means we enable further methods to improve surface completeness, for example, to create a water-tight surface model as in [Dey and Goswami, 2003]. Like the robust 3D reconstruction of points, the surface estimation is run concurrently during the planning procedure.

Similar to the proceeding in [Salman and Yvinec, 2010], we establish a non-discrete surface estimation pursuing the following steps, which are also illustrated in Figures 6.11 to 6.13.

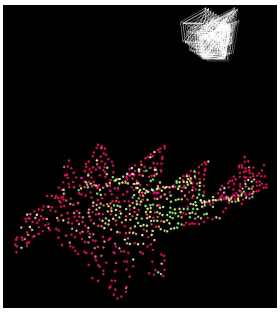
1. Using the concurrently reconstructed 3D points exemplified in Figure 6.11(b), we create a *complete 3D Delaunay triangulation* shown in Figure 6.11(c). The resulting set of tetrahedrons fulfills the Delaunay condition, which we refer to later on, and occupies the convex hull of the input points. Our special interest is directed towards the side faces of the Delaunay tetrahedrons. A part of these 3D triangles is assumed to locally approximate the 3D surface to reconstruct.

2. We apply *visibility constraints* to eliminate impossible 3D triangles that are part of the complete Delaunay triangulation, see Figure 6.11(d).

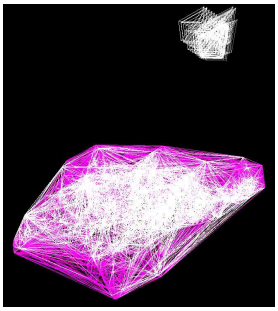
3. For each of the 3D triangles left, we establish the *probability* that the respective triangle is actually a part of the surface to reconstruct, see Figures 6.12 and 6.13.

Complete 3D Delaunay triangulation. We aim to establish a non-discrete surface representation given a finite, discrete set of points, an image sequence, and additional information concerning tracking and camera parameters. To

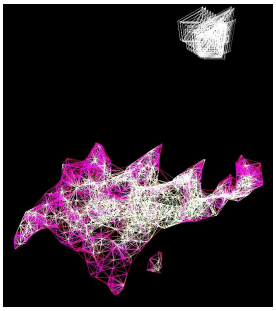
(a) Example image of the input sequence.



(b) Concurrently reconstructed 3D points.



(c) Wireframe model created by complete Delaunay triangulation indicating 3D triangles.



(d) Wireframe model reduced by visibility constraints.

Figure 6.11: Establishing and reducing 3D triangles based on Delaunay triangulation and visibility constraints. Colors (red to green) originate from uncertainty ratings of the 3D points and from overlay in the viewport (towards white). White pyramids indicate camera poses.

this end we can imagine to fit any kind of parametric surface to the given data. However, parametric fitting mostly comes with a complex optimization procedure. Furthermore, certain types of parametric surfaces often favor very specific data only. While we do not exclude parametric fitting in further steps, we stick with simple piecewise linear approximation of the surface, in the first place. As very simple planar elements we use 3D triangles. Now, if we try to create a set of candidate triangles, we have to apply some criterion of reasonability that avoids triangles leading straight through the whole set of input points, for instance. Such a criterion for reasonable triangles is found in terms of the empty sphere condition of 3D Delaunay triangulation. The 3D

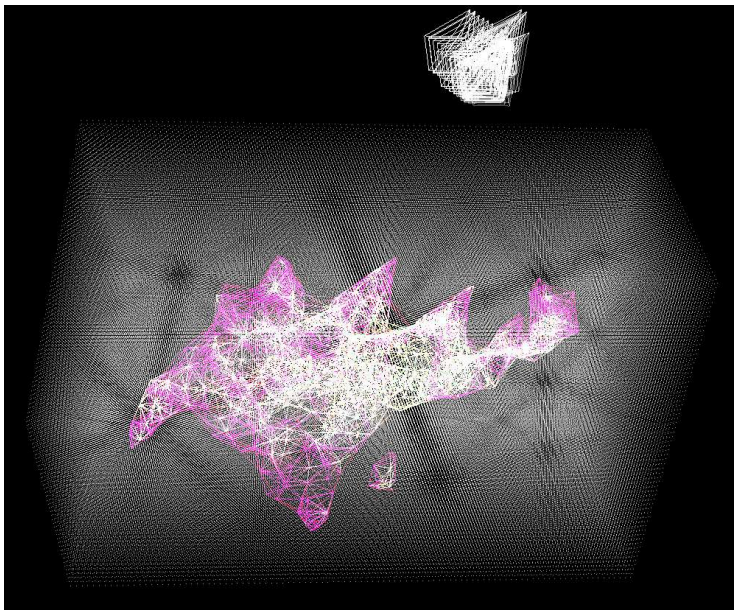


Figure 6.12: Candidate triangles and static voxel space. Each voxel (white points) receives a probability to belong to the object surface. Based on the voxel probabilities, we establish the respective probability for each triangle.

Delaunay triangulation takes a set of 3D points and establishes tetrahedrons in way that each point is corner of at least one tetrahedron. Additionally, the empty sphere condition, or Delaunay condition, says that the circumscribing sphere of each tetrahedron does not contain a fifth point, except for cases of cospherical points. In doing so, we create a set of tetrahedrons, the outer faces of which occupy the convex hull of the input points, cf. Figure 6.11(c). Additionally, the side faces of the tetrahedrons are reasonable triangles in the sense that they induce a useful neighborhood relation in the input set of points. Thus, we assume the resulting set of 3D triangles to contain triangles that locally approximate the surface to be reconstructed.

Eliminating impossible triangles by visibility constraints. This work is not restricted to convex objects, and Figure 6.11(c) illustrates an example of a convex hull featuring triangles that are obviously impossible. In the sense considered here, a triangle is impossible if and only if it crosses the ray of sight of an observed point between the camera center and the point. In other words, if we observe a certain point in a certain image and a candidate

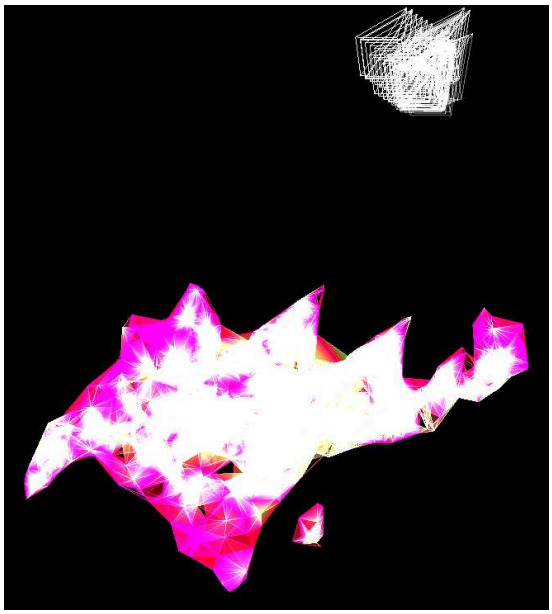


Figure 6.13: Concurrent probabilistic surface estimation for the dinosaur figurine. Transparency indicates probability for each 3D triangle. Colors (red to green) originate from uncertainty ratings of the 3D points and from overlay in the viewport (towards white).

triangle occludes the corresponding 3D point, then the triangle is not a part of the surface to be reconstructed and is deleted. The checking procedure is achieved by simple geometric calculations. Based on the reconstruction uncertainty of the 3D points, we further delete triangles featuring an uncertain corner point.

Assessing the surface probability for the remaining triangles. For the verification of the remaining triangles, we use additional image information. The approach is inspired by the idea of space carving as depicted in [Kutulakos and Seitz, 2000]. The idea is that only voxels on a surface theoretically feature a constant projection intensity in different images. Of course, the theoretically constant appearance is not constant in practice, and the sensor system has to be calibrated using, for instance, a homogeneous scene. Following the concept of space carving, we place a static voxel space in the scene, cf. Figure 6.12, and employ a consistency function that evaluates if a voxel is part of the object surface. More precisely, we use the consistency

function to compute the probability of the respective voxel being part of the object surface, which we shortly call surface probability. Based on the surface probabilities of voxels, we compute the surface probability of a triangle using the probabilities of neighboring voxels. While a static voxel space may seem to be overstated effort, it offers the great advantage that we can handle a changing 3D estimation. This becomes obvious if we consider to sample the actual candidate triangles and to check the image consistencies of these sample points on the triangles. Since the candidate triangles are established using the reconstructed 3D points, the triangles change during the planning process and do not provide constant sampling positions. Instead, we perform static voxel sampling as follows.

For each static voxel $\mathcal{V}$ we project the voxel center to an image point $\mathbf{x}_\mathcal{V}$ and yield an intensity value $I(\mathbf{x}_\mathcal{V})$. We do so in each image and hence achieve a variance $\sigma_\mathcal{V}^2$ that represents the appearance volatility of voxel $\mathcal{V}$ in the image sequence. Calibrating the planning system with respect to constant appearance yields a system parameter σ_c^2 that describes the variance of a surface voxel caused by sensor noise. Using the calibration σ_c^2, we establish a logistic distribution $P(\mathcal{V} \in \mathcal{S}|\sigma_\mathcal{V}^2, \sigma_c^2)$ expressing the probability that voxel $\mathcal{V}$ is an element of the surface $\mathcal{S}$. Hence, each voxel of the static voxel space is augmented by a probability to belong to the surface to be reconstructed. Inside this voxel space there are the 3D point estimates and the 3D triangles as shown in Figure 6.12. Now, for each triangle T we evaluate the probabilities of the neighboring voxels. In this sense, neighboring voxels of the triangle are those voxels, the centers of which feature a distance to the triangle that is smaller than or equal to the half of a voxel diagonal. A triangle T thus defines a set

$$\mathsf{V}(\mathsf{T}) = \{\mathcal{V} : \mathcal{V} \text{ is a neighbor of } \mathsf{T}\} \tag{6.39}$$

of neighboring voxels. Regarding the neighboring voxels, we define the probability of triangle T to belong to surface $\mathcal{S}$ as

$$P(\mathsf{T} \in \mathcal{S}) = \begin{cases} 0, & z \geq z_t \\ \mathbb{E}\left(P\left(\mathcal{V} \in \mathcal{S}|\sigma_\mathcal{V}^2, \sigma_c^2\right)\right), \mathcal{V} \in \mathsf{V}(\mathsf{T}), & \text{else} \end{cases} \tag{6.40}$$

using

$$z = \frac{\#\{\mathcal{V} : \mathcal{V} \in \mathsf{V}(\mathsf{T}) \wedge P\left(\mathcal{V} \in \mathcal{S}|\sigma_\mathcal{V}^2, \sigma_c^2\right) = 0\}}{\#\{\mathcal{V} : \mathcal{V} \in \mathsf{V}(\mathsf{T})\}} \tag{6.41}$$

and a threshold $z_t \in [0, 1]$. For instance, $z_t = 0.5$ provokes that the surface probability of a triangle is zero if at least 50% of the neighboring voxels

feature a surface probability of zero. If less than 50% of the neighboring voxels fulfill this condition, then (6.40) computes the surface probability of the triangle as the mean surface probability of the neighboring voxels. Regarding the assumption of a voxel belonging to the object surface, the according image consistency is expressed in terms of the projection intensity variance as described above. This voxel consistency allows to derive a consistency estimation for surface triangles using (6.40).

One drawback of the basic idea of space carving is that it is valid only for convex objects. So, if the surface probability of a voxel is low, this may be caused by a self-occlusion of the object. However, we can trust that a high surface probability of a voxel in fact indicates what is wanted. Based on this fact, we refer to highly probable surface triangles and do not update the variance $\sigma_{\mathcal{V}}^2$ if voxel $\mathcal{V}$ is covered by such a triangle. Thus, we only evaluate the image consistencies of unoccluded voxels and by this means handle the surface estimation for general, possibly non-convex objects. Still, problems are raised by a homogeneous background, since voxels in front of such feature a high consistency. This problem is inherent to the usage of a passive camera and can only be resolved by realizing large baselines.

Conclusion. We present a planning module performing concurrent, probabilistic surface estimation based on 3D Delaunay triangulation and image consistency checking. As the result we get a set of 3D triangles, each of which is associated with a probability to belong to the object surface to be reconstructed. The triangles use the concurrently reconstructed 3D points as corner points. This module uses additional image data to enrich the information carried by the 3D reconstruction, which regards a non-discrete representation of the reconstructed surface. Thereby, this planning module enables the application of further methods for view planning and surface reconstruction.

6.5 Including Visibility Constraints

One procedural planning aspect is the visibility of a point regarding a certain camera pose. Procedural means that this visibility is not a part of the planning goals, but it has to be respected during the planning procedure in order to achieve the planning goals efficiently. Exploiting the probabilistic surface estimation presented in Section 6.4, we first compute the probability

that a point is occluded in a certain image. Then, we discuss strategies to minimize this occlusion probability within this module.

Aiming to find triangles of the surface estimation that cover a point regarding a certain camera pose, we analyze intersections of surface triangles and the ray of sight. For a point estimate $\widehat{\mathbf{P}}$ that is in the viewport of a camera with center $\mathbf{C}$, the 3D line segment

$$\mathbf{s} = \overline{\widehat{\mathbf{P}}\mathbf{C}} \tag{6.42}$$

is used to check the visibility of $\widehat{\mathbf{P}}$. We define a set $\mathsf{I}_\mathbf{s}$ containing all triangles that intersect the line segment $\mathbf{s}$,

$$\mathsf{I}_\mathbf{s} = \{\mathsf{T} : \mathbf{s} \text{ and } \mathsf{T} \text{ intersect}\}. \tag{6.43}$$

Applying these terms and (6.40), we can formulate the probability

$$P_o\left(\widehat{\mathbf{P}}|\mathbf{C}\right) = \begin{cases} 0, & \#\mathsf{I}_\mathbf{s} = 0 \\ \max\limits_{\mathsf{T}\in\mathsf{I}_\mathbf{s}} P(\mathsf{T} \in \mathcal{S}), & \text{else} \end{cases} \tag{6.44}$$

that the point estimate $\widehat{\mathbf{P}}$ is occluded by the object surface $\mathcal{S}$ when being observed by a camera with center $\mathbf{C}$. The occlusion probability is zero, if no surface triangle crosses the ray of sight. Otherwise, the occlusion probability equals the maximal surface probability of the crossing triangles.

Now that we know how to assess the occlusion probability for a certain point estimate, we want it to be minimal, in general. The concrete formulation of how to include visibility constraints strongly depends on the actual planning goals. For a simple instance, if we aim to optimize the 3D estimate of one certain point, it is obvious that we want to minimize the occlusion probability for this specific point during the planning procedure. Assuming an accuracy optimization as in Section 6.3, we have to adapt the camera motion, if necessary, and find the optimal unoccluded view of the wanted point. We suggest to keep the occlusion probability at zero and find an optimal dodge movement that is the closest to the one given by evaluating the EEC. One way to do this is a Monte-Carlo analysis minimizing the distance between the executed motion and the optimal camera motion while keeping the occlusion probability at zero. For other planning goals, we have to incorporate the visibility analysis in an appropriate manner. Seen as a planning module, the visibility analysis requires the output of the probabilistic surface estimation and yields the occlusion probability for a certain entity and camera pose.

6.6 Model-based Treatment of Geometric Primitives

The probabilistic surface estimation shown in Section 6.4 codes more information than a finite set of 3D points. In particular, this enables to approach the wide field of model-based view planning, i. e. view planning using a known 3D CAD model, or an equivalent, of the object to be reconstructed. Since model-based view planning is extensive and not in the focus of this work, we present just one specific idea to use knowledge of the object for view planning within our approach.

Imagine that we reconstruct an object using view planning and that we have a special interest in geometric primitives such as the negative half sphere in Figure 6.1. This is to say that if we detect such a surface element during the reconstruction, then we want to perform specific actions that promise certain reconstruction qualities, for example, a high accuracy and a high point density for the particular surface element. To this end we propose the following proceeding.

1. Initialize the reconstruction and surface estimation by data-driven view planning as presented in this work.

2. Having a suitable surface estimation, try to detect the geometric primitive using the surface features shown in Chapter 5.

3. For a successful detection, invoke a stored view plan that has been trained offline.

The procedure requires an offline training step yielding view plans for a set of geometric primitives. For instance, we may be interested in surface elements like partial spheres, corners, or edges. For each of these primitives, we prepare a view plan that favors defined reconstruction qualities. One way to find a suitable view plan is reinforcement learning [Sutton and Barto, 1998]. Storing learned view plans for certain primitives and criteria concludes the offline training step. We then start the actual online view planning in a data-driven manner. Once we find the quality of the point and surface estimates sufficient, we apply methods proposed in Chapter 5 to identify possible appearances of the wanted geometric primitives. The experiments in Section 5.5.5 support the assumption that the presented features of 3D

surface triangulations are proper means to detect surface elements within a probabilistic 3D surface triangulation. We suggest the following procedure to detect specific surface elements. Regarding each geometric surface element, we select a central reference point and radius for the computation of local surface features. Applying the same radius, we also compute the local surface features for each point of the current 3D reconstruction. Using the best match within the current 3D reconstruction, we perform the presented 3D registration of the wanted element locally around the detected match. Based on the registration error we decide on the appearance of the geometric primitive within the current 3D reconstruction. In the case of appearance, we invoke the according learned view plan and then continue the online view planning. In this way, we suggest to include the knowledge of optimal view plans for given geometric primitives into the planning procedure.

6.7 View Planning with Respect to the Reconstruction Method of Factorization

The previous sections evolved the components of a modular system for online view planning. This system endows a plug-in architecture and enables to address a variety of planning goals and constraints. All of the presented modules have in common that they provide view planning regarding the observed data, exclusively. In this section we go one step further and examine view planning with respect to the observed data *and* the specific reconstruction method of factorization. For this investigation we explicitly leave the generic, modular, online planning system and refer to the planning goal of accuracy optimization for a set of points using the reconstruction method of factorization.

The motivation of investigating view planning with respect to both data and method is given by one particular promising property of the factorization method. Unlike methods that reconstruct one 3D point, factorization yields a commonly optimal solution for all points and all views. It is thus a valuable goal to investigate interdependencies between specific camera poses and the reconstruction quality achieved by factorization. We may as well assume to have influence on the selection of a subset of scene points to reconstruct. As a result, view planning can optimize the setup in order to further improve the reconstruction accuracy.

In the first place, we give a short review of factorization methods for 3D reconstruction and outline the one used in this work. Afterwards, we describe a simulation system and various simulations that allow to analyze the behavior of the used factorization method. Finally, we draw conclusions on view planning aspects.

6.7.1 Reviewing Factorization

In general, the factorization method takes the image mappings of a set of 3D points in all views, and it yields an estimation of the original 3D points and the camera poses. In [Tomasi and Kanade, 1992], Tomasi and Kanade describe the factorization method under the assumption of orthographic projection. In the context of computer vision, they are the first to outline the algebraically optimal self-calibration performed by factorization. Later on, Poelman and Kanade [Poelman and Kanade, 1997] extend the factorization method for weak-perspective and paraperspective projections using given intrinsic camera parameters. An iterative scheme presented in [Han and Kanade, 1999] even allows Euclidean reconstruction assuming the perspective camera model. Without knowledge of the intrinsic camera parameters and without further information, all we can achieve is a projective reconstruction. Sturm and Triggs [Sturm and Triggs, 1996] introduce an according factorization procedure using the perspective camera model. In this work, the authors iteratively refine the projective depths and the scene estimation to finally compute a projective scene reconstruction.

Since we are interested in a Euclidean reconstruction assuming the perspective camera model, we concentrate on the iterative procedure depicted in [Han and Kanade, 1999]. The authors propose iterations of weak-perspective factorizations in order to approximate the perspective model.

The weak-perspective factorization, which is part of each iteration, is described in [Poelman and Kanade, 1997]. For the weak-perspective projection model see Section 3.1.1.2 in this work. The method takes image points, constructs a measurement matrix $\mathbf{W}^*$, and decomposes $\mathbf{W}^*$ into matrices $\mathbf{M}$ and $\mathbf{S}$ representing motion and structure using singular value decomposition. By means of the image points in F frames, factorization yields P 3D points $\mathbf{s}_p$, $1 \leq p \leq P$. Each of the F cameras is characterized by a translation vector $\mathbf{t}_f$ and a rotation matrix $\mathbf{R}_f$ consisting of row vectors $\mathbf{I}_f$, $\mathbf{J}_f$, and $\mathbf{K}_f$, $1 \leq f \leq F$. Assuming weak-perspective projection and known intrinsic

camera parameters, the normalized coordinates of the image points are

$$u_{fp} = \frac{\mathbf{I}_f(\mathbf{s}_p - \mathbf{t}_f)}{-\mathbf{K}_f\mathbf{t}_f} = \frac{\mathbf{I}_f\mathbf{s}_p - \mathbf{I}_f\mathbf{t}_f}{z_f} = \mathbf{m}_f\mathbf{s}_p + x_f, \tag{6.45}$$

$$v_{fp} = \frac{\mathbf{J}_f(\mathbf{s}_p - \mathbf{t}_f)}{-\mathbf{K}_f\mathbf{t}_f} = \frac{\mathbf{J}_f\mathbf{s}_p - \mathbf{J}_f\mathbf{t}_f}{z_f} = \mathbf{n}_f\mathbf{s}_p + y_f. \tag{6.46}$$

Stacked as matrices, these equations give

$$\begin{pmatrix} u_{11} & \dots & u_{1P} \\ \vdots & & \vdots \\ u_{F1} & \dots & u_{FP} \\ v_{11} & \dots & v_{1P} \\ \vdots & & \vdots \\ v_{F1} & \dots & v_{FP} \end{pmatrix} = \begin{pmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_F \\ \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_F \end{pmatrix} \begin{pmatrix} \mathbf{s}_1 & \dots & \mathbf{s}_P \end{pmatrix} + \begin{pmatrix} x_1 & \dots & x_1 \\ \vdots & & \vdots \\ x_F & \dots & x_F \\ y_1 & \dots & y_1 \\ \vdots & & \vdots \\ y_F & \dots & y_F \end{pmatrix}, \tag{6.47}$$

$$\mathbf{W} = \mathbf{MS} + \mathbf{T}. \tag{6.48}$$

Without loss of generality, factorization assumes that the gravity center of the points to reconstruct is located in the origin of the coordinate frame,

$$\sum_{p=1}^{P} \mathbf{s}_p = \mathbf{0}. \tag{6.49}$$

It follows that

$$\sum_{p=1}^{P} u_{fp} = \sum_{p=1}^{P} (\mathbf{m}_f\mathbf{s}_p + x_f) = \sum_{p=1}^{P} x_f = Px_f \tag{6.50}$$

and

$$x_f = \frac{1}{P}\sum_{p=1}^{P} u_{fp}. \tag{6.51}$$

The same derivation applies to y_f and v_{fp}. Hence, the assumption (6.49) enables to compute the entries of matrix $\mathbf{T}$ from (6.48) and to create the modified measurement matrix

$$\mathbf{W}^* = \mathbf{W} - \mathbf{T} = \mathbf{MS}. \tag{6.52}$$

Since both $\mathbf{M}$ and $\mathbf{S}$ theoretically feature a maximal rank of three, this is also the maximal rank of $\mathbf{W}^*$. Using singular value decomposition yields a factorization

$$\mathbf{W}^* = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T} = \sum_i \mathbf{u}_i \sigma_i \mathbf{v}_i^\mathsf{T}. \tag{6.53}$$

In theory, $\sigma_i = 0$ for $i > 3$. At least due to noise, this is unlikely for practical application. That is why this constraint is enforced,

$$\begin{aligned}
\mathbf{W}^* &\approx \sum_{i=1}^{3} \mathbf{u}_i \sigma_i \mathbf{v}_i^\mathsf{T} && (6.54)\\
&= \mathbf{U}'\mathbf{D}'\mathbf{V}'^\mathsf{T} && (6.55)\\
&= \left(\mathbf{U}'\mathbf{D}'^{0.5}\right)\left(\mathbf{D}'^{0.5}\mathbf{V}'^\mathsf{T}\right) && (6.56)\\
&= \mathbf{M}''\mathbf{S}'' && (6.57)
\end{aligned}$$

The approximation step in (6.54) is optimal in the sense that the difference between both sides is minimal in terms of the Frobenius norm. Reached by singular value decomposition, the matrices $\mathbf{M}''$ and $\mathbf{S}''$ are determined only up to a invertible matrix $\mathbf{A}$ since

$$\mathbf{M}''\mathbf{S}'' = \mathbf{M}''\left(\mathbf{A}\mathbf{A}^{-1}\right)\mathbf{S}'' = \left(\mathbf{M}''\mathbf{A}\right)\left(\mathbf{A}^{-1}\mathbf{S}''\right). \tag{6.58}$$

This affine ambiguity is resolved by applying constraints that arise from the structure of the above equations. With

$$\begin{aligned}
\mathbf{M}' &= \mathbf{M}''\mathbf{A} \quad \text{and} && (6.59)\\
\mathbf{S}' &= \mathbf{A}^{-1}\mathbf{S}'' && (6.60)
\end{aligned}$$

it is known from (6.45) and (6.46) that

$$\begin{aligned}
\|\mathbf{m}'_f\|_2 &= \|\mathbf{n}'_f\|_2, \quad \text{hence} && (6.61)\\
\|\mathbf{m}''_f\mathbf{A}\|_2 &= \|\mathbf{n}''_f\mathbf{A}\|_2, && (6.62)\\
\left(\mathbf{m}''_f\mathbf{A}\right)\left(\mathbf{m}''_f\mathbf{A}\right)^\mathsf{T} &= \left(\mathbf{n}''_f\mathbf{A}\right)\left(\mathbf{n}''_f\mathbf{A}\right)^\mathsf{T}, && (6.63)\\
\mathbf{m}''_f\mathbf{A}\mathbf{A}^\mathsf{T}\mathbf{m}''^\mathsf{T}_f - \mathbf{n}''_f\mathbf{A}\mathbf{A}^\mathsf{T}\mathbf{n}''^\mathsf{T}_f &= 0 && (6.64)
\end{aligned}$$

since $\mathbf{m}'_f$ and $\mathbf{n}'_f$ must be vectors of a rotation matrix multiplied by z_f. As constituting different rows of a rotation matrix, the respective vectors must

also fulfill

$$\mathbf{m}'_f \mathbf{n}'^{\mathsf{T}}_f = 0, \quad \text{hence} \tag{6.65}$$

$$\mathbf{m}''_f \mathbf{A}\mathbf{A}^{\mathsf{T}} \mathbf{n}''^{\mathsf{T}}_f = 0. \tag{6.66}$$

Further, the scaling of $\mathbf{M}'$ is fixed by

$$\|\mathbf{m}'_1\|_2 = 1, \quad \text{hence} \tag{6.67}$$

$$\mathbf{m}''_1 \mathbf{A}\mathbf{A}^{\mathsf{T}} \mathbf{m}''^{\mathsf{T}}_1 - 1 = 0. \tag{6.68}$$

The expressions (6.64), (6.66), and (6.68) state homogeneous, linear constraints on the symmetric matrix

$$\mathbf{Q} = \mathbf{A}\mathbf{A}^{\mathsf{T}}. \tag{6.69}$$

Once $\mathbf{Q}$ and thus $\mathbf{A}$ are computed, the final sign ambiguity remains,

$$\mathbf{W}^* = \mathbf{M}'\mathbf{S}' = (-\mathbf{M}')\,(-\mathbf{S}')\,. \tag{6.70}$$

Without additional constraints, only the resulting reconstruction errors can help to decide for the final choice of $\mathbf{M}$ and $\mathbf{S}$.

Now that we know about weak-perspective factorization, we aim to apply the full perspective projection model. Employing the symbols introduced above, the perspectively mapped and normalized image points are

$$u^\star_{fp} = \frac{\mathbf{I}_f(\mathbf{s}_p - \mathbf{t}_f)}{\mathbf{K}_f(\mathbf{s}_p - \mathbf{t}_f)} = \frac{\frac{\mathbf{I}_f(\mathbf{s}_p - \mathbf{t}_f)}{-\mathbf{K}_f\mathbf{t}_f}}{\frac{\mathbf{K}_f\mathbf{s}_p}{-\mathbf{K}_f\mathbf{t}_f} + 1} = \frac{u_{fp}}{\epsilon_{fp} + 1} = \frac{u_{fp}}{\lambda_{fp}}, \tag{6.71}$$

$$v^\star_{fp} = \frac{\mathbf{J}_f(\mathbf{s}_p - \mathbf{t}_f)}{\mathbf{K}_f(\mathbf{s}_p - \mathbf{t}_f)} = \frac{\frac{\mathbf{J}_f(\mathbf{s}_p - \mathbf{t}_f)}{-\mathbf{K}_f\mathbf{t}_f}}{\frac{\mathbf{K}_f\mathbf{s}_p}{-\mathbf{K}_f\mathbf{t}_f} + 1} = \frac{v_{fp}}{\epsilon_{fp} + 1} = \frac{v_{fp}}{\lambda_{fp}} \tag{6.72}$$

using $\lambda_{fp} = 1 + \epsilon_{fp}$ and $\epsilon_{fp} = (\mathbf{K}_f\mathbf{s}_p)(-\mathbf{K}_f\mathbf{t}_f)^{-1}$. The coefficients λ_{fp} relate weak-perspective image points to the perspective ones, which allows to establish an iterative factorization scheme as done in [Han and Kanade, 1999]. Assuming perspectively projected points $(u^\star_{fp}, v^\star_{fp})^{\mathsf{T}}$, the according weak-perspective expressions $(\lambda_{fp}u^\star_{fp}, \lambda_{fp}v^\star_{fp})^{\mathsf{T}}$ are valid input to the weak-perspective factorization method. By initializing all λ_{ij} to 1, the first iteration is a pure weak-perspective factorization. Afterwards, the λ_{ij} are

adjusted using the latest result, the next weak-perspective factorization uses the updated λ_{ij}, and so on. Han and Kanade show that this kind of iteration reaches convergence towards the perspective reconstruction of points and camera poses. The nature of this method is to be seen as self-calibration using intrinsic camera parameters. As a part of this, the method also occurs as an algorithm for 3D reconstruction jointly optimizing the result for all points in all views.

In the following, we examine the performance of the described reconstruction method with respect to specific scene characteristics.

6.7.2 Simulating Planning Aspects

While this section has an experimental character, it rather describes simulations to reveal theoretical relations between a 3D scene, which consists of points and cameras, and the resulting reconstruction quality achieved by factorization. We aim to apply these relations to view planning in the sense that we have influence on the camera poses and the selection of points to reconstruct.

As a simulation environment we employ the following setup, which is exemplified in Figure 6.14 with $P = 50$ points, $F = 50$ cameras, and a sphere radius $r = 10$. We randomly select uniformly distributed 3D points inside the cube $[-0.5, 0.5]^3$, which we call *unit cube* in the following. The simulated cameras are located on a half sphere, and their z-axes are oriented towards the center of the corresponding sphere. Again, we randomly select the positions of the cameras on the half sphere, which are two parameters, and the direction of the x-/y-axes, which is one parameter since the orientation of the z-axis is fixed. Throughout the simulations we vary the numbers of points and cameras and other scene characteristics specified below. For a defined set of scene characteristics, we run a Monte-Carlo analysis using 100 cycles. For each single cycle, we simulate the 3D scene, compute the image points, feed the image points to the factorization yielding a 3D reconstruction, and align the reconstructed to the reference points using the known correspondences. As the main quality criterion, we use the reconstruction errors as box-whisker plots of the 100 cycles for each configuration, i. e. the errors $\epsilon_{2.5}$, ϵ_{25}, ϵ_{50}, ϵ_{75}, and $\epsilon_{97.5}$ describe the quantiles in the sorted sequence of error values, and ϵ_{50} is the median error.

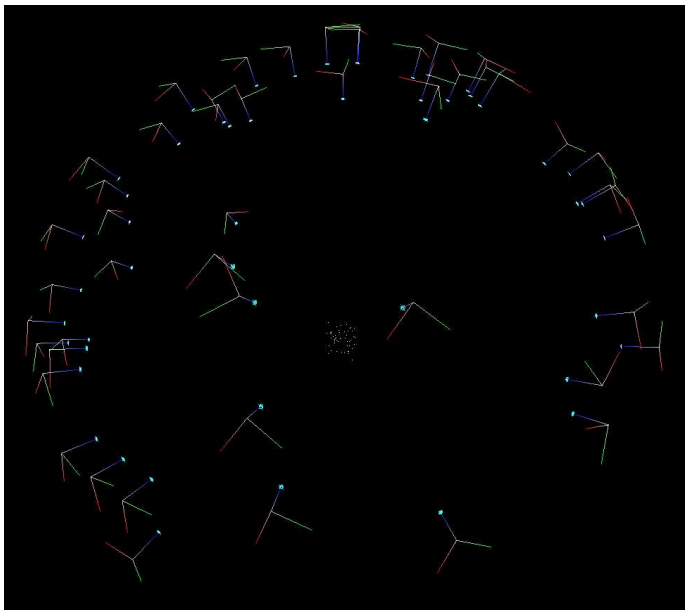


Figure 6.14: An example of a simulated scene of 50 3D points (white dots) in the unit cube and 50 cameras on a half sphere, the optical axes of which (blue lines) are oriented towards the center of the corresponding sphere with radius 10. The camera x-/y-axes (red/green lines) are tangential to the sphere surface. More details are given in Section 6.7.2.

6.7.2.1 Counting Degrees of Freedom

As a first framing aspect we investigate the minimal numbers of points and cameras necessary for the weak-perspective factorization. On the one side, the known parameters are P image points in F images, whereat we centralize the points for each image. This makes

$$2PF - 2F = 2(P-1)F \tag{6.73}$$

known parameters. On the other side, we need to fix the degrees of freedom corresponding to P centralized 3D points, F camera rotations, and the depth informations z_f for F cameras, which in total are

$$3P - 3 + 3F + F = 3P + 4F - 3 \tag{6.74}$$

degrees of freedom. The method hence claims

$$2(P-1)F \geq 3P + 4F - 3. \tag{6.75}$$

As a result, the weak-perspective factorization needs at least four points seen in five frames to succeed,

$$P_{\min} = 4, \tag{6.76}$$
$$F_{\min} = 5. \tag{6.77}$$

In addition to the number of parameters, we have to respect rank criteria. Since the algebraic fact

$$\mathsf{rk}(\mathbf{AB}) \leq \min(\mathsf{rk}(\mathbf{A}), \mathsf{rk}(\mathbf{B})) \tag{6.78}$$

also applies to the product $\mathbf{W}^* = \mathbf{M}''\mathbf{S}''$, the given data need to satisfy the condition

$$\mathsf{rk}(\mathbf{W}^*) \geq 3, \tag{6.79}$$

whereat theory suggests equality and a larger rank is due to noise or modeling insufficiencies. Later on, the simulations reveal that the rank criterion is of particular importance for the factorization method.

6.7.2.2 Assessing the Numbers of Points and Cameras

Regarding a controlled environment used for view planning, we may assume to have influence on the numbers of points and cameras in one particular reconstruction step. This is to say that we have control over selecting a certain subset of points to reconstruct with a certain number of views. Besides the actual camera positions, these numbers are the first the check for factorization. Hence, we simulate 3D scenes featuring $P = 5$, 10, 20, 40 points, and each point number is related to a number of cameras $F \in [5, 50]$. The points are uniformly distributed inside the unit cube, the cameras on a half sphere. The radius of the corresponding sphere is $r = 10$, and the sphere center equals $(0, 0, 0)^\mathsf{T}$, which is also the gravity center of the unit cube.

To give an impression of the error distributions, we first provide the box plots for $P = 10$ points and $F = 5, 6, \ldots, 50$ cameras in Figure 6.15. The decreasing progression of the error quantiles indicates that using more random cameras provides a more stable reconstruction. Another noticeable observation is that the best error quantiles $\epsilon_{2.5}$ are nearly constant for all numbers of cameras. This lower boundary, in our understanding, is due to the used numerical methods in combination with aspects of machine precision. So, the interesting part of the error evaluation is the behavior of the median error

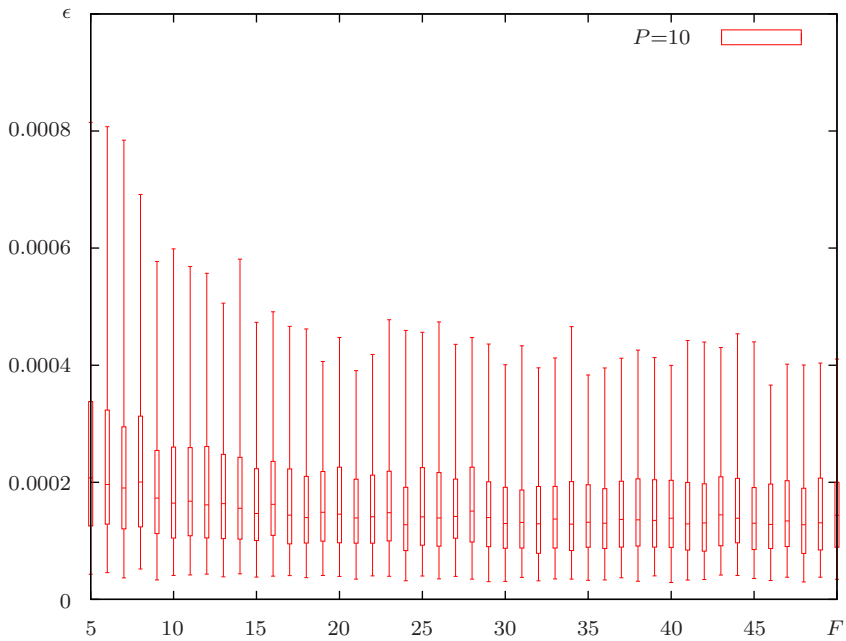


Figure 6.15: Box plots of the reconstruction errors ϵ for $\mathsf{P} = 10$ points and $\mathsf{F} = 5, 6, \ldots, 50$ cameras. The points lie in a cube with edge length 1, and the cameras on a half sphere with radius $\mathsf{r} = 10$, cf. Figure 6.14.

and the upper error quantiles. We further investigate the effects of different numbers of random points in Figure 6.16. For the sake of readability, we just note the median errors ϵ_{50} for the simulations with $P = 5$, 10, 20, 40 points. As a first observation, the median errors are decreasing using more random points. Second, this improvement of the results is subject to saturation as there is a large improvement from $P = 5$ to $P = 10$ and nearly no improvement from $P = 20$ to $P = 40$ points. Third, using more random points causes the error curve to be more smooth, which indicates more robust results. Regarding the relation between the number of points and the number of cameras, there is no clear statement to derive from Figure 6.16, since more points alone and more cameras alone both improve the stability. We run further simulations to find a hint on the reasons for the varying reconstruction stability.

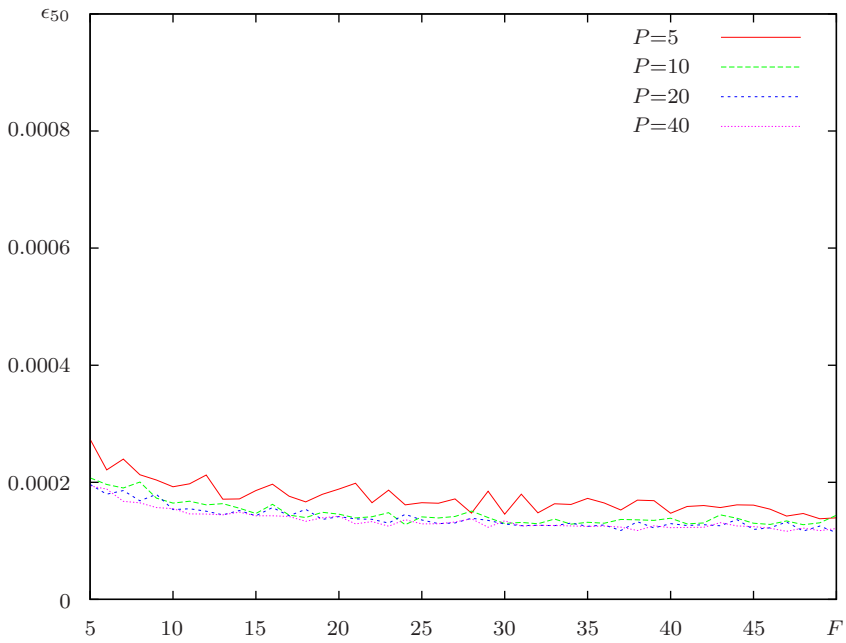


Figure 6.16: Median errors ϵ_{50} for simulations featuring different point numbers. An increasing number of points improves both accuracy and reconstruction stability.

6.7.2.3 Assessing Issues Regarding the Weak-perspective Model

The factorization method applied here is an iterative procedure running a weak-perspective factorization in each step. While this approach approximates a reconstruction assuming the perspective model, it still is not equivalent. For this reason, we examine the effects of scene characteristics that are not included in the weak-perspective projection model.

Performing a weak-perspective projection can be illustrated in the following geometric manner. There is the weak-perspective camera observing a set of 3D scene points. Further imagine a virtual support plane that is parallel to the image plane of the camera and leads through the gravity center of all scene points. Now, the weak-perspective projection can be expressed as an orthographic projection of the scene points to the virtual support plane followed by a perspective projection of these virtual support points.

One crucial aspect of the selected projection model is the scene depth. To be more precise, the crucial aspect is the scene depth in relation to the

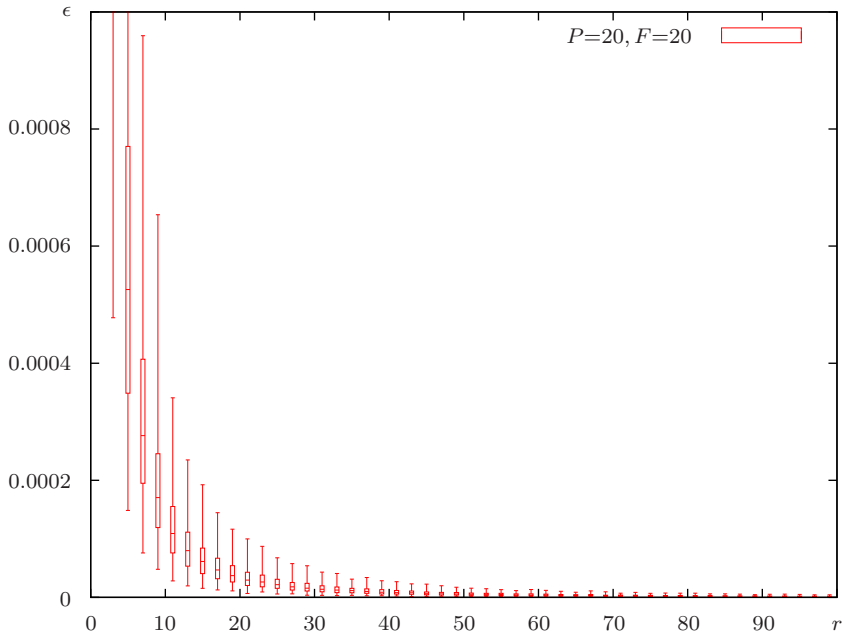


Figure 6.17: Box plots of the reconstruction errors for different values of radius r, i. e. for different distances between scene points and cameras and constant extent of scene points. Larger radii yield better accuracy.

distance between camera and scene points. If all scene points are situated on the virtual support plane, then the scene depth is zero and the weak-perspective projection equals the perspective one. Otherwise, the distance between camera and scene points should be large compared to the scene depth in order to keep model-based errors small. Figure 6.17 shows the reconstructions errors for a simulation setup with $F = 20$ random cameras and $P = 20$ random points in the unit cube. Again, the cameras lie on a half sphere. The corresponding sphere has its center in the center of the unit cube, but we evaluate different radii $r = 1, 3, 5, \ldots, 99$ and keep the scene depth constant. While we used $r = 10$ in the previous experiments, we recognize from Figure 6.17 that a larger radius, ceteris paribus, significantly reduces the reconstruction error, which is due to the smaller deviation from the weak-perspective model.

For general scenes, another important aspect is the centering of the image points, whereat the center is the principal point of the image. The geometrical visualization described above implies that the model-based error is mini-

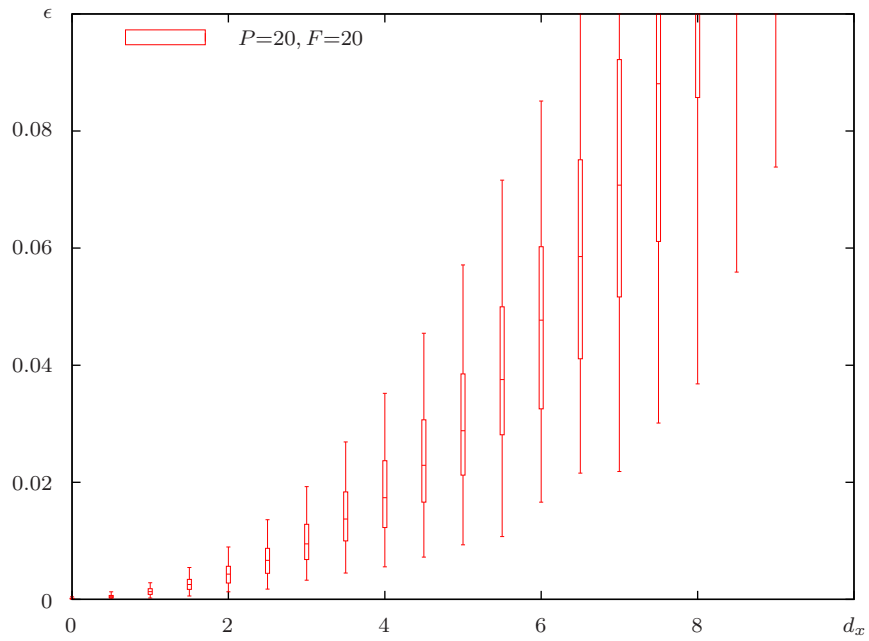


Figure 6.18: Box plots of the reconstruction errors for decentration $\mathsf{d_x}$ and sphere radius $\mathsf{r} = 10$. The weak-perspective model penalizes the gravity center of the image points being out of $(0, 0)^\mathsf{T}$.

mal for image points featuring the gravity center at the principal point. Any decentration increases the difference between the perspective and the weak-perspective projection. We generate the decentration of the image points by the decentration of the unit cube inside the sphere of possible camera positions. Without loss of generality, we keep the unit cube at the origin and apply a decentration $d_x = 0, 2, 4, 6, 8, 10$ to the position of the sphere center. The sphere features $r = 10$. Again, $F = 20$ random cameras observe $P = 20$ random points. The error values in Figure 6.18 indicate a severe influence of decentration on the reconstruction error. Large decentrations increase the error by several magnitudes.

The effects described in this section are due to the properties of the weak-perspective projection model. Yet, there are other scene characteristics having impact on the factorization result.

6.7.2.4 Assessing Rank Criteria

Despite the conditions of the weak-perspective projection model, we want to find crucial aspects depending on the mathematics of factorization. As the most critical computational step we identify the actual factorization and rank approximation in (6.54) to (6.57). Since this step involves the singular value decomposition, the ranks of the concerned matrices are of particular importance. Rank deficiencies cause arbitrary vectors $\mathbf{u}_i$, $\mathbf{v}_i$, see (6.54), since the corresponding singular value σ_i is zero or close to zero. This yields arbitrary submatrices of the structure and motion matrices, the effects of which we demonstrate by the following experiments. We identify kinds of data dependencies and measure their impact on the reconstruction result.

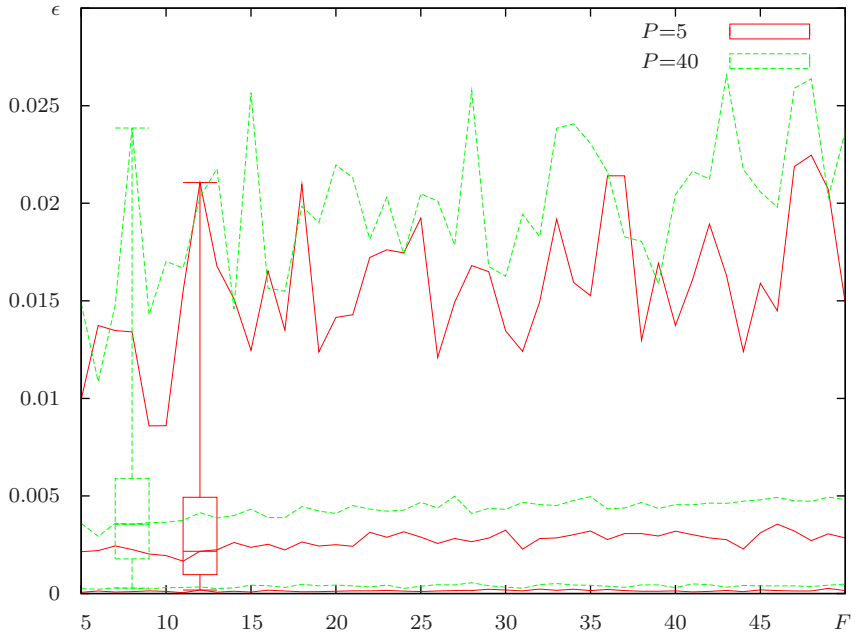


Figure 6.19: Reconstruction errors for $\mathsf{P} = 5/40$ collinear points (red/green) in the unit cube, $\mathsf{F} = 5, 6, \ldots, 50$ cameras and $\mathsf{r} = 10$. In comparison, the median error for 5 random points is around 0.0002.

First, we show the effects of a rank deficiency of the structure matrix $\mathbf{S}$ in Figure 6.19. Here, we use $P = 5$ and $P = 40$ collinear points passing the origin in the unit cube, $F = 5, 6, \ldots, 50$ random cameras, and the radius of the camera (half) sphere $r = 10$. Thus, $\mathsf{rk}(\mathbf{S}) = 1$. In general, collinear points

would provide $\mathsf{rk}(\mathbf{S}) = 2$, but in accordance to (6.49) factorization assumes scene points featuring their gravity center at the origin. This means that the original structure matrix may feature rank two, but the factorization tries to restore one with rank one. For $P = 5$, the respective median error ϵ_{50} using random points lies around 0.0002 and $\epsilon_{97.5}$ around 0.0005, both of which are clearly exceeded in this test. While the low error quantiles compare to the case of random points, the upper ones are much higher, which indicates a decreased stability of the reconstruction. Unlike for random points, using more collinear points decreases the stability even more.

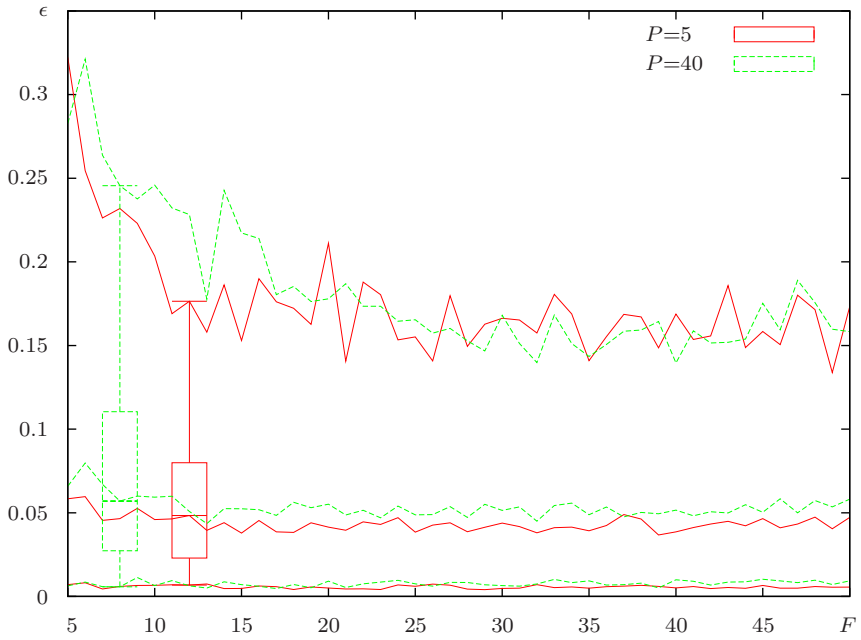


Figure 6.20: Reconstruction errors for $\mathsf{P} = 5/40$ coplanar points (red/green) in the unit cube, $\mathsf{F} = 5, 6, \ldots, 50$ cameras and $\mathsf{r} = 10$. In comparison, the median error for 5 random points is around 0.0002.

We repeat the previous test with the same parameters, but settle for coplanar instead of collinear points in the unit cube, hence $\mathsf{rk}(\mathbf{S}) = 2$. We select points in the Y-Z plane. Just as for collinear points, general coplanar points cause rank three of the structure matrix, but factorization assumes the gravity center of the scene points at the origin, thus trying to restore a rank-2 structure matrix. The errors printed in Figure 6.20 are again larger than for collinear points. In relation to the edge length 1 of the unit cube, the

reconstruction results seem to be more or less random and useless. At this point we again refer to the weak-perspective projection model. As outlined above, this model best approximates and is even equivalent to the perspective model if all scene points are located on the virtual support plane. This appears to constitute an inherent theoretical drawback of the weak-perspective factorization method: When the weak-perspective model best approximates the perspective model, the factorization will fail.

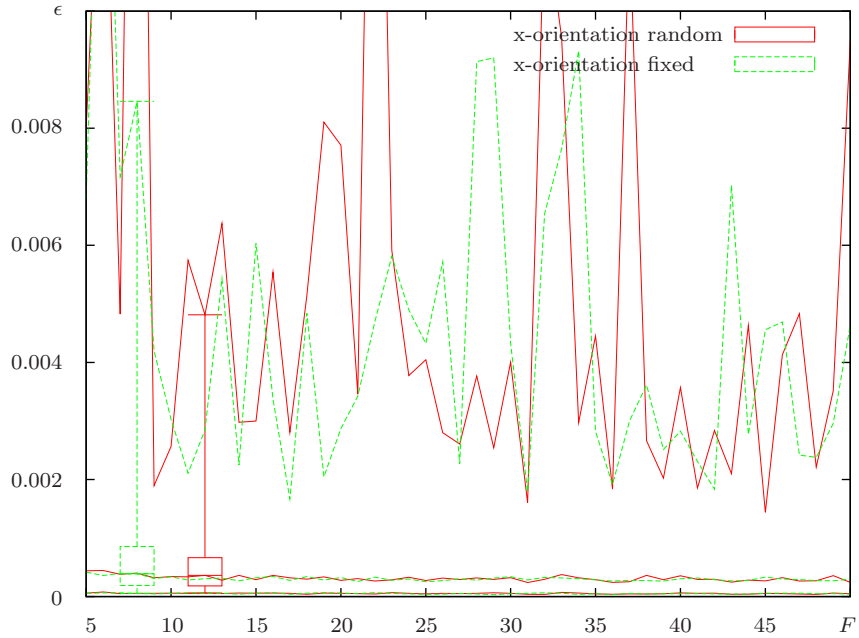


Figure 6.21: Reconstruction errors for $P = 5$ random points and $F = 5, 6, \ldots, 50$ cameras, the positions of which are restricted to one meridian of the camera sphere with $r = 10$. The x-/y-orientations are random (red) or the x-orientation is constant (green). In comparison, the median error for random cameras is around 0.0002.

The previous tests used random camera position on the whole camera half sphere and scene points restricted to subspaces. In the following we do it vice versa and examine constrained camera positions, while the scene points are fully random in the unit cube.

Figure 6.21 gives the reconstruction errors of simulations featuring the following setup. $P = 5$ random points in the unit cube are imaged by $F = 5, 6, \ldots, 50$ cameras that lie on one meridian of the camera half sphere,

i. e. the azimuth angle of the camera position is the same for all cameras. As for camera orientations, we examine random x-/y-directions and a fixed x-direction for all cameras. The z-direction is still directed towards the center of the camera sphere. We note that the resulting median errors ϵ_{50} are comparable to the fully random setup evaluated in Figure 6.16. However, the reconstruction stability is much worse in the sense that constrained camera positions feature much larger upper error quantiles. This applies to both random and fixed x-/y-directions of the cameras. Since the motion matrix still features $\mathsf{rk}(\mathbf{M}) = 3$ in these cases, there has to be another critical step. As such we find the step of imposing the metric constraints (6.64), (6.66), and (6.68) on the matrix $\mathbf{Q}$ in (6.69). The metric constraints mainly depend on the direction vectors of the camera orientations. There have to be enough linearly independent camera orientations in order to fix the six degrees of freedom featured by $\mathbf{Q}$. In addition, the whole range of each parameter should be covered in order to avoid adaptation to noise. For constrained camera orientations, these conditions are violated. To emphasize this ill-conditioned behavior, we simulate cameras featuring identical orientations. Naturally, these cameras cannot lie on a sphere and look to its center at the same time. So, we place the cameras in the plane $Z = 10$ and confine the positions to a 1×1-area in order to control decentration effects. In addition to the difficulties in the computation of $\mathbf{Q}$, this setup provides the deficiency $\mathsf{rk}(\mathbf{M}) = 2$ as well. As the error results in Figure 6.22 outline, the reconstructions are useless.

Finally, we check another possible interdependence of the input data. To this end we employ the fully random setup from Section 6.7.2.2, but we add the following modification for the selection of camera positions. Instead of randomly selecting each camera position once, we do this ten times for each camera and eventually choose the position that yields image points providing the maximal ratio of singular values. This ratio will be 1 if the image points are equally distributed inside of a circle centered at the origin, otherwise the ratio increases for the image points advancing collinearity. Perfectly collinear image points introduce dependencies to the measurement matrix $\mathbf{W}$, which possibly reduce its rank. To confirm it again, the $P = 5$ points are simulated fully random in the unit cube. As the error plot in Figure 6.23 demonstrates, the dependencies cause an increased instability of the reconstruction results. The median error is about ten times larger as for random camera positions without constraints. Using more cameras does not improve the results.

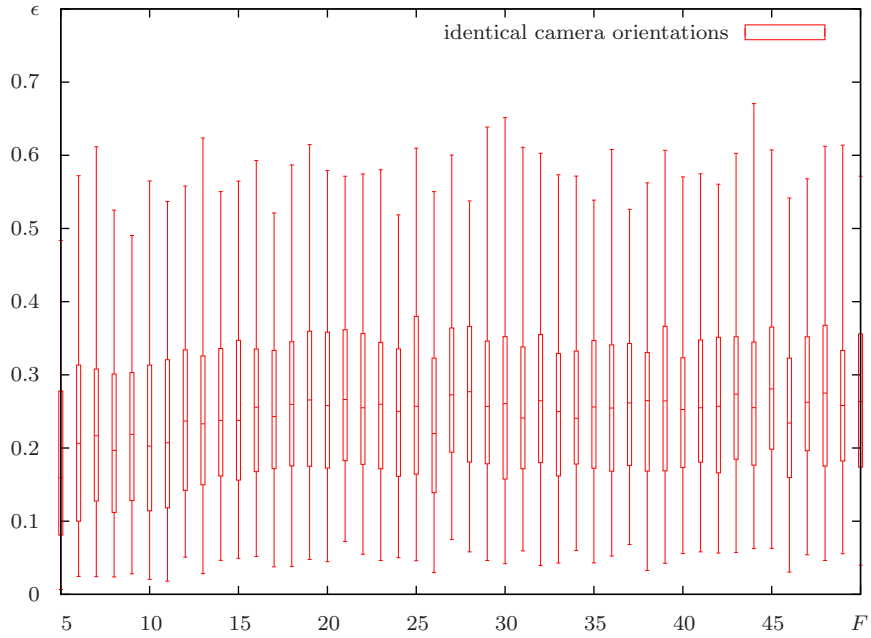


Figure 6.22: Reconstruction errors for $\mathsf{P} = 5$ random points and $\mathsf{F} = 5, 6, \ldots, 50$ cameras featuring identical orientations. The points are still established inside the unit cube with edge length 1.

6.7.3 Implications for View Planning Using Iterations of Weak-perspective Factorization

Employing the factorization method appears to be favorable since it finds a reconstruction that is optimal with respect to all points in all views. As always, the optimality is based on a set of more or less explicit assumptions, and it is most interesting how the method can cope with violations of these assumptions. In the context of view planning, we are given the opportunity to sustain the assumptions and to avoid critical configurations. Assessing the above simulation results, we now list first implications for view planning using factorization.

- If we have no information about the scene points, the simulation results suggest to use as many points as possible in order to avoid deficient point configurations. The reconstruction of technical objects states a special challenge, since it is easily possible to select points on a linear edge or in a plane. It would be the responsibility of the planning

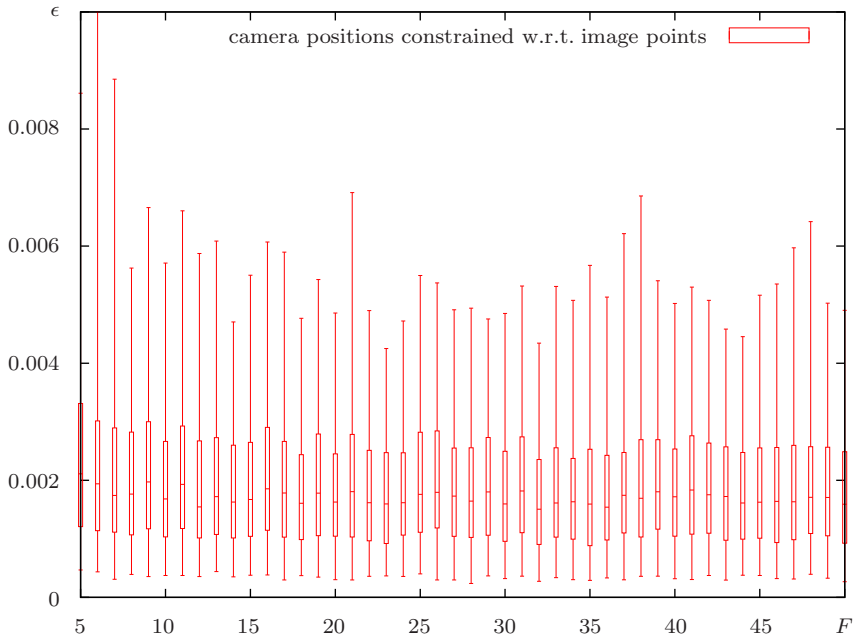


Figure 6.23: Reconstruction errors for $\mathsf{P} = 5$ random points and $\mathsf{F} = 5, 6, \ldots, 50$ camera positions that are random but optimized towards a maximal ratio of the singular values of the image points, i. e. preferably collinear image points in the respective image.

method to avoid the selection of such point sets, which could be done by an initial coarse estimation of the 3D structure using a PMD camera as in [Munkelt et al., 2009].

- Regarding the weak-perspective projection model, the cameras should be placed such that the image points feature their centroid at the principal point of the image. This avoids reconstruction errors caused by decentration. Furthermore, the simulations imply to keep as much distance as possible between the cameras and the scene points. Of course, this is a bad idea for practical application since this measure increases the negative effect of positioning errors in the image plane, for instance, tracking errors. Hence, there is no clear statement concerning the camera-scene distance. If there is a way to relate the modeling error, which suggests large distances, and the positioning error in the image plane, which suggests small distances, in the particular

planning application, then the planning method should optimize the resulting error expectation. Another way of dealing with the ratio of scene depth and camera distance would a selection of specific points to reconstruct, which would again require an initial 3D estimation. The part to optimize would be the selection of linearly independent scene points featuring a minimal space occupation, i. e. local scene depth.

- Again, we point to the theoretical shortcoming of the weak-perspective factorization, cf. Section 6.7.2.4, that the factorization fails when the weak-perspective model works best, i. e. if all scene points are located in the virtual support plane.

- In contrast to triangulation, pure camera translation is not enough for the factorization method to succeed. Since the motion matrix holds scaled versions of the camera orientations, identical orientations cause rank deficiencies. It follows that a planning method should provide significant changes of the camera orientations and should optimize this regarding dependencies and range.

- In spite of specific numerical methods, the factorization still takes perspectively projected image points and infers on the 3D scene structure. The setup in Section 6.3 describing a model of the maximal reconstruction error hence applies as well, and so do the conclusions derived in Section 6.3.2. Thus, the planning method should aim to establish a perpendicular ray of sight towards the main uncertainty direction of entity to be optimized.

- Any kind of linear or close to linear dependency of the relevant terms introduces possible instability to the factorization method, which opens the door for future research on this topic.

The factorization method offers a variety of links to view planning. We understand the above simulations and conclusions as a first step to perform view planning with respect to the reconstruction method. Regarding the factorization method, future work is necessary to further investigate the algebraic and numerical relations between a 3D scene and the achieved reconstruction quality as well as the influence of noise. A specific interesting question is what kind of dependencies may exist between the points and camera parameters, and how these dependencies influence accuracy and stability of the factorization method. Further, we may focus not only on the

abstract computational steps like singular value decomposition, but as well address actual numerical procedures like Householder transformation. Finally, it seems appealing to deal with the one serious practical drawback of the factorization method: We need to see each point in each image. In the first major publication [Tomasi and Kanade, 1992], the authors already deal with this problem and propose a hierarchical application that basically is a partial reconstruction and mapping of the missing points. In contrast, view planning allows a scenario in which, based on an initial 3D estimation, the number and positions of cameras observing a certain subset of scene points may be optimized.

Chapter 7

Experimental Evaluation

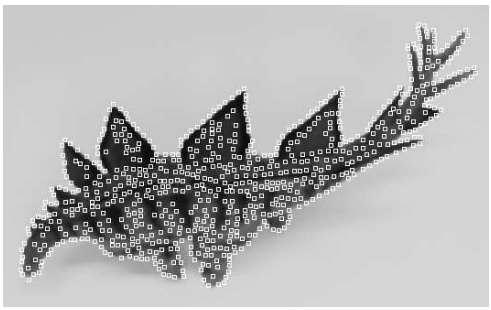
In the previous chapter, we presented our approach to modular, online next-best-view planning using a passive camera. Now, we want to assess the performance of the proposed modules and of further practical planning components. This chapter is explicitly restricted to evaluate planning components and applications. For an in-depth testing of the GKLT tracking method and the 3D surface registration, the reader is referred to the Sections 4.5 and 5.5, respectively.

7.1 Basic Aspects of the Experimental Evaluation

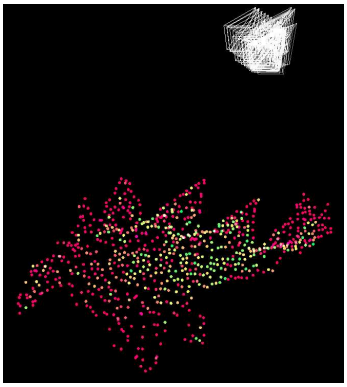
The hardware of our controlled environment is a robotic arm Stäubli RX90L as shown in the Figures 1.1 and 4.14, on which a 640×480 Sony DFW-VL500 firewire camera is mounted. For the robotic arm, the repeat accuracy in positioning is ± 0.025mm, and the angular resolution of the joints is $0.002°$. These data imply that the more critical part regarding the accuracy of camera poses is the hand-eye calibration. We determine this transformation between the coordinate frames of the gripper and the camera by means of the MATLAB Camera Calibration toolbox, in particular by the parts developed by C. Wengert, see `http://www.vision.ee.ethz.ch/software/calibration_toolbox/calibration_toolbox.php`.

The general experimental setup is outlined in the Figures 7.1 and 7.2. We place the object of interest within range of the robotic arm. Before the

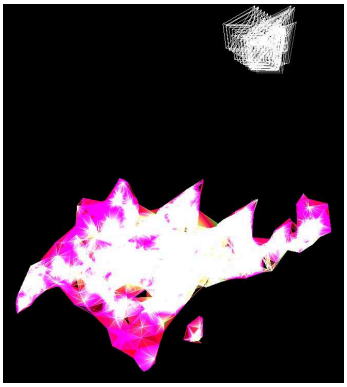
(a) The real-world Dino Detlef in the laboratory, placed within range of the robotic arm shown in Figure 1.1.



(b) One image of the sequence. The markers indicate features tracked by GKLT tracking within the planning procedure.



(c) Concurrent, robust 3D reconstruction by GKLT. Only point estimations and cameras are shown.



(d) Concurrent 3D estimation including probabilistic surface estimation.

Figure 7.1: General experimental setup exemplified by Dino Detlef.

start of the planning, we initialize framing aspects, for instance, the number P of feature points that are detected in the first frame and shall be tracked and reconstructed by GKLT tracking. We further tell the planning method which kind of planning to use, if the probabilistic surface estimation shall be run concurrently, and we define the parameters of the positioning sphere. The concept of the positioning sphere is illustrated in the Figures 6.8 to 6.10. This sphere is given by the center $\mathbf{C} = (C_1, C_2, C_3)^\mathsf{T}$ and radius r_C. A camera position is then defined by the azimuth angle $0° \leq \alpha < 360°$ and the height angle $0° \leq \beta \leq 90°$ assuming that the optical axis is oriented towards $\mathbf{C}$ and the camera's roll angle is normalized. This setup offers the great advantage that we can easily keep the object in the camera viewport. The model implies

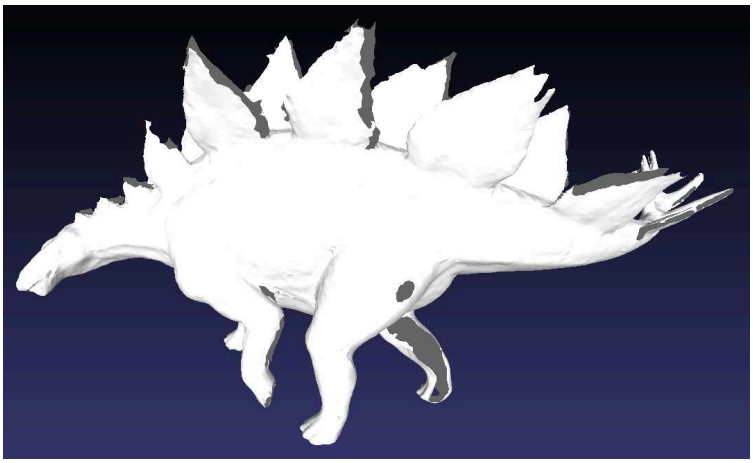


Figure 7.2: Ground-truth 3D model of Dino Detlef obtained by a fringe-projection system.

a theoretical, but not a practical restriction of the full 6D Euclidean space of camera poses. Since we normalize the roll angle of the camera, we use only five degrees of freedom, but we still reach any position and any direction of the optical axis using the positioning sphere.

Given the initial parameters, we adjust the camera position and take the first image and detect features, see Figure 7.1(b). It follows a sequence of 5 random camera movements in order to initialize the 3D point estimations and the covariance data, see Figure 7.1(c). Subsequently, the actual planning is invoked and takes control over data analysis and view planning, which might include the concurrent probabilistic surface estimation, see Figure 7.1(d). During the planning procedure we collect and log the data necessary for the evaluation. To derive absolute error measures for accuracy evaluation, we use a 3D ground-truth model as in Figure 7.2. The model is given either as a CAD model or as a high-accuracy measurement yielded by a fringe-projection system as in [Kühmstedt et al., 2007]. We perform a semi-automated coarse registration of the model into the measurement space and align an actual measurement to the model using the ICP algorithm. For the comparison of different planning methods we employ identical setups and initializations.

7.2 Assessing Accuracy Optimization Regarding the Extended E-criterion

The first series of experiments shall be dedicated to the accuracy optimization presented in Section 6.3. To this end we perform 3D reconstruction of different objects and evaluate the uncertainty as well as the absolute 3D error. As for uncertainty, we refer to (6.37) and use the sum of eigenvalues

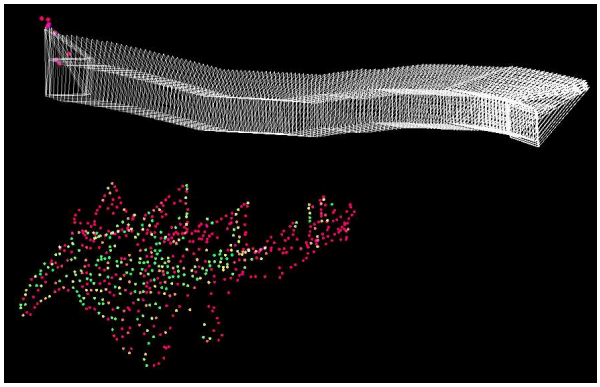
$$\lambda_\Sigma = \text{trace}(\Lambda) = \lambda_1 + \lambda_2 + \lambda_3 \tag{7.1}$$

of the according covariance matrix as a measure of uncertainty. In terms of accuracy criteria, we do not employ the compact notation giving the mean error but stick to the box-whisker representation $\epsilon_{2.5}, \ldots, \epsilon_{97.5}$ of the absolute 3D errors in meters.

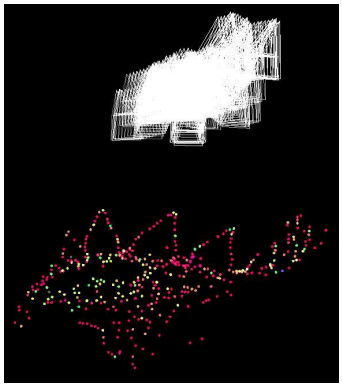
7.2.1 Catching Points

One first and trivial thing to recognize is that we cannot improve the 3D estimation of a point that we do not observe. Hence, we have to ensure that the points to optimize are being tracked in the images. The GKLT tracking method supports to continue tracking a previously lost feature. Exploiting this property, we catch lost points during the planning procedure. Specifically, we analyze for each view if there are any features that have been lost more than $t_{\text{lost}} \in \mathbb{N}$ frames ago. If so, we determine the most recent camera pose from which the feature has been seen and make a random step from there. This is the actual *catching* attempt, which we repeat at most $t_{\text{catch\_max}} \in \mathbb{N}$ times for each point. If the tracking does not succeed in one of the attempts, we *ignore* the feature and the according 3D point. An ignored feature will not be tracked or caught any more. The according 3D point is part of the resulting 3D reconstruction only in case that its uncertainty rating has a relative rank of less than $t_{\text{rank\_ignore}} \in [0, 1]$.

The setup of the following test is an initialization of 700 features to track and the goal to minimize the uncertainty of the worst point. We run the according planning procedure with and without catching of lost points. The catching is invoked if any point was not tracked for $t_{\text{lost}} = 10$ frames. In this case, we run at most $t_{\text{catch\_max}} = 1$ attempt to reinitialize tracking for this feature. In case of failure, the respective feature will be ignored. The according 3D point is part of the resulting 3D reconstruction only if its uncertainty



(a) Smooth camera path according to accuracy optimization. (Un-)Successful tracking is not regarded.



(b) Camera path is interrupted by catching operations. Direction changes, if worst point changes.

Figure 7.3: 3D scenes and camera positions achieved with/without catching of lost points.

rank is better than $t_{\text{rank\_ignore}} = 0.2$, i. e. if it is among the one fifth least uncertain points. Figure 7.3 demonstrates the effects the catching operations have on the planned camera path. Without catching, the camera path in Figure 7.3(a) strictly obeys the accuracy optimization of the worst point. If the according feature is not being tracked, the worst point estimate will not change. In contrast, catching lost features ensures to actually observe the features. The uncertainties of the point estimates change, the actual worst point estimate is getting improved and replaced by another worst point, which explains the changing direction of the camera path in Figure 7.3(b). The numerical evaluations of the measured uncertainties, Figure 7.4(a), and 3D errors in meters, Figure 7.4(b), underline that without catching of lost features the reachable improvement is strictly bound, which also is a kind of trivial. With an increasing number of frames, more features will be lost by the feature tracker, and the according point estimates will not be improved. In Figure 7.4(b) we see that the reconstruction achieved without catching is more accurate than the planned one, but only in the first frames. Afterwards, the version with catching performs better. This effect may occur since any planning process is initialized randomly. We further observe that the uncertainty and error plots are not strictly monotone. This property will apply to all the following experiments since feature tracking and 3D reconstruction,

which provide the terms being evaluated, are always subject to noise.

Unless stated otherwise, we use catching of lost features with the above parameters for all the experiments following.

7.2.2 Evaluating Random, Regular, and Planned Camera Movements

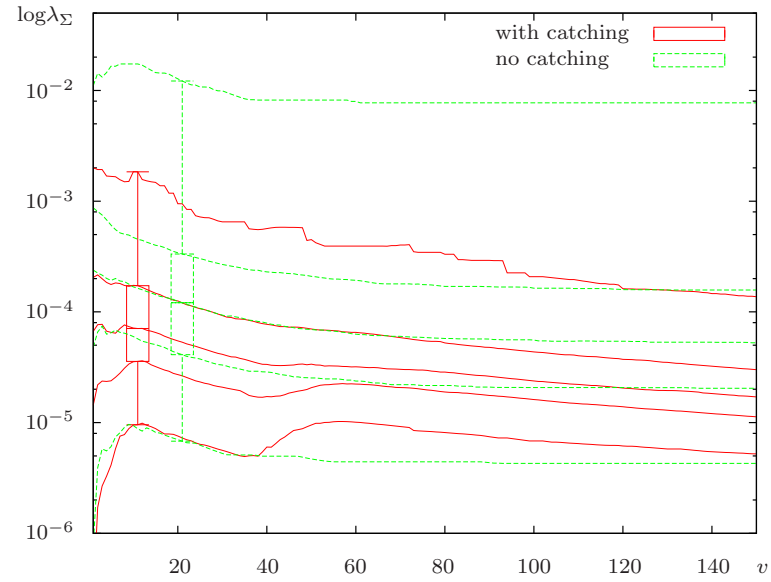
This set of experiments investigates the performance of the accuracy optimization with respect to EEC as presented in Section 6.3.2. In order to be justified to disregard visibility issues, we initialize 100 features in a planar scene that is favorably oriented, see Figure 7.5(a). The Figures 7.5(b) to 7.5(d) exemplify the camera paths resulting from random movements and view planning for accuracy optimization. First, for the view planning strategy IMPROVE_WORST each camera movement is being planned for accuracy optimization with respect to EEC regarding the respective worst point, i. e. the center of uncertainty and the main direction of uncertainty are defined by the respective worst point. Second, the strategy RANDOM uses strictly random camera movements in terms that each next camera pose is random on the camera sphere while regarding the constrained step size. Third, we initially select a random direction for the camera motion and keep it fixed for the strategy FIXED_DIRECTION. This test pattern corresponds to the regular camera movements often used for comparison by other authors. Since all reconstruction runs are initialized randomly, we present the numerical results of three runs for each strategy. For the accuracy evaluation, we perform plane fitting to the reconstructed points and measure point-to-plane distances. We comprehend the strategies for camera movement used by now:

RANDOM No view planning. Each camera movement random.

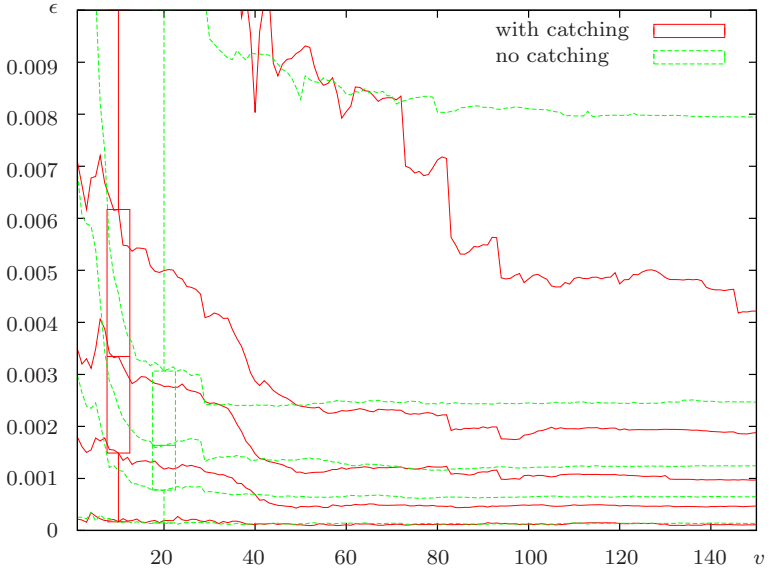
FIXED_DIRECTION No view planning. One random direction of camera movement fixed.

IMPROVE_WORST View planning for accuracy optimization w.r.t. EEC. Uncertainty data provided by respective worst point.

The view planning strategy IMPROVE_WORST is one first way of implementing accuracy optimization with respect to the extended E-criterion, in which we refer to the respective worst 3D point estimate. From Figure 7.6 we observe that the three runs IMPROVE_WORST 1, 2, and 3 yield comparable

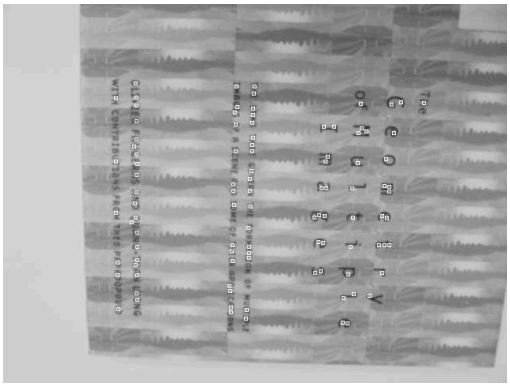


(a) Evaluation of uncertainties logλ_Σ.

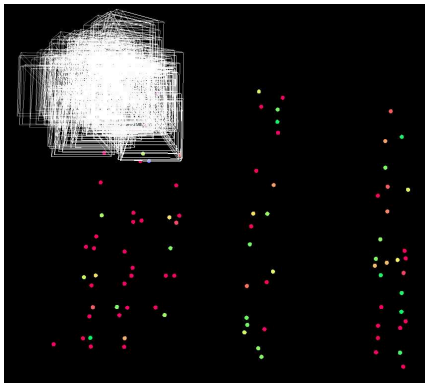


(b) Evaluation of 3D errors ϵ in meters.

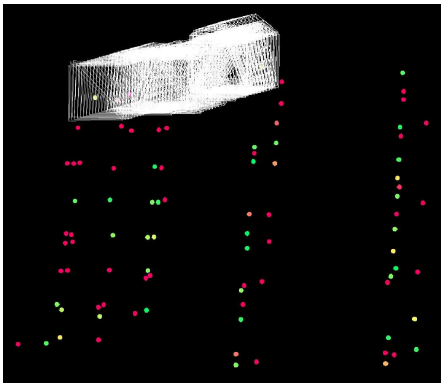
Figure 7.4: Numerical evaluation comparing ignoring and catching of lost points after v views. Trivially, uncertainties and errors are reduced only for points that are tracked. For an increasing number of lost features and without catching, the values stagnate. Catching allows to further reduce uncertainty and errors.



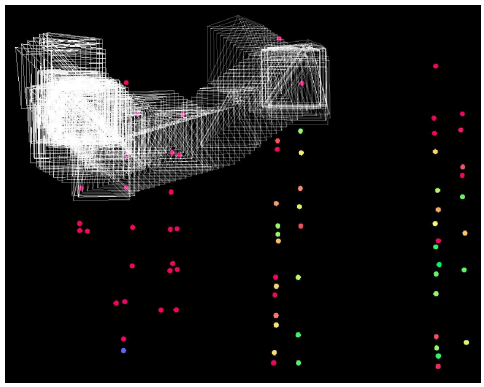
(a) 100 features on a book cover courtesy of [Faugeras and Luong, 2001].



(b) 3D scene achieved by random camera movements.



(c) 3D scene achieved by camera movements towards a random, fixed direction.



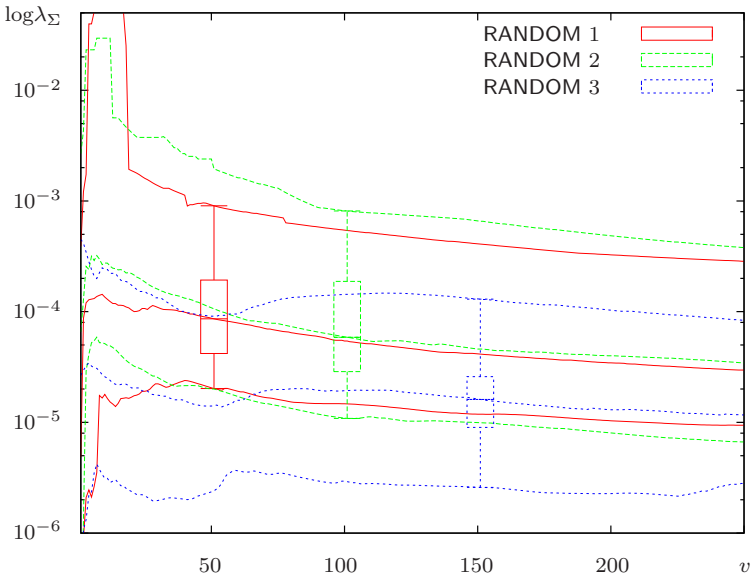
(d) 3D scene achieved by view planning as in Section 6.3.2. Improving the respective worst point.

Figure 7.5: Example image of coplanar points and 3D scenes achieved with different strategies using 250 frames each.

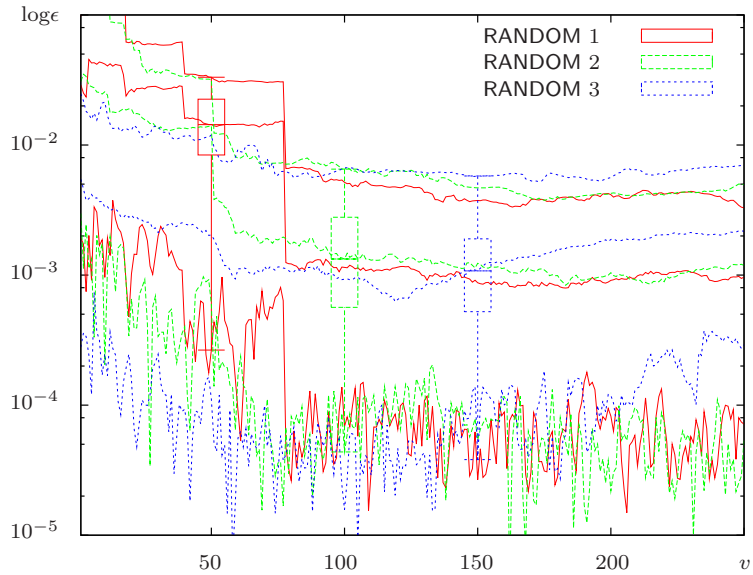
(a) Evaluation of uncertainties logλ_Σ.

(b) Evaluation of 3D errors logϵ in meters.

Figure 7.6: Numerical results for three runs using view planning according to IMPROVE_WORST.

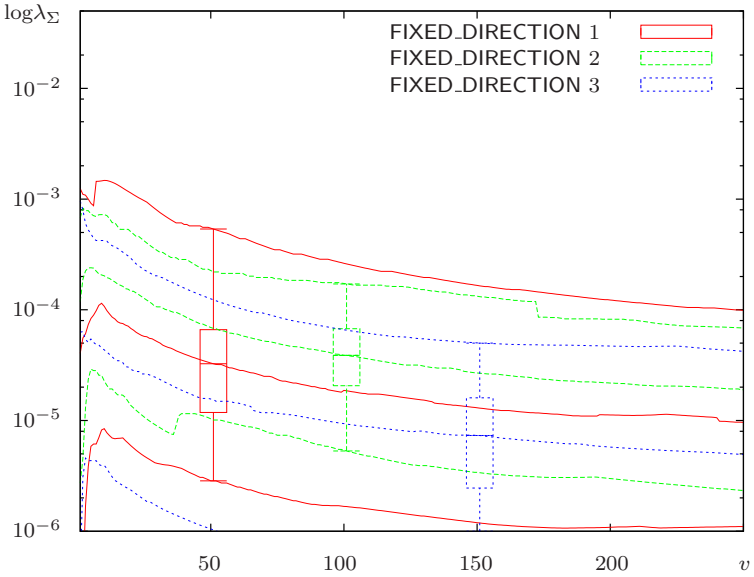


(a) Evaluation of uncertainties logλ_Σ. Wide spread between different runs.

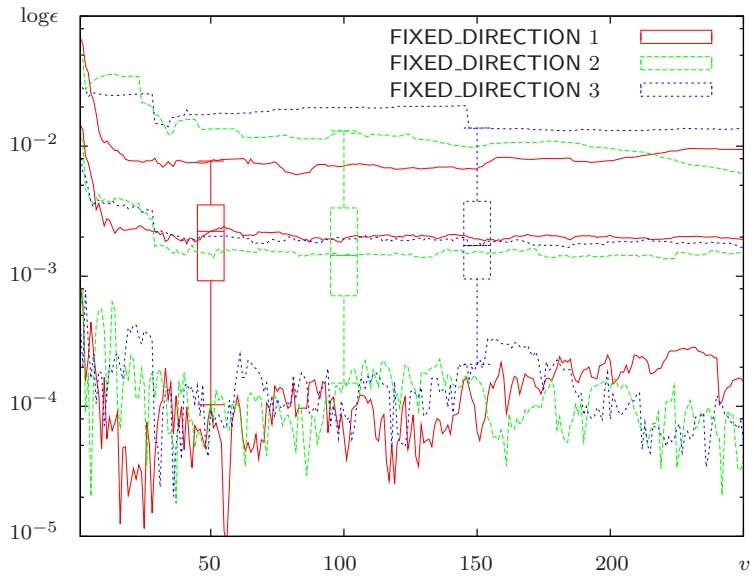


(b) Evaluation of 3D errors logϵ in meters. Wide spread, unstable curves, errors are even increasing.

Figure 7.7: Numerical results for three runs without view planning and according to strategy RANDOM.



(a) Evaluation of uncertainties logλ_Σ. Wide spread between different runs.



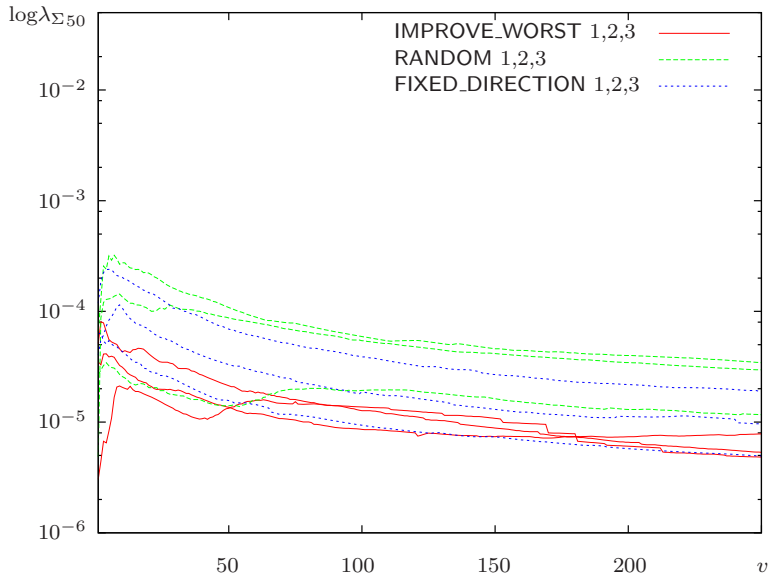
(b) Evaluation of 3D errors logϵ in meters. Errors stagnate at a high level.

Figure 7.8: Numerical results for three runs without view planning and according to strategy FIXED_DIRECTION, which performs regular camera movements.

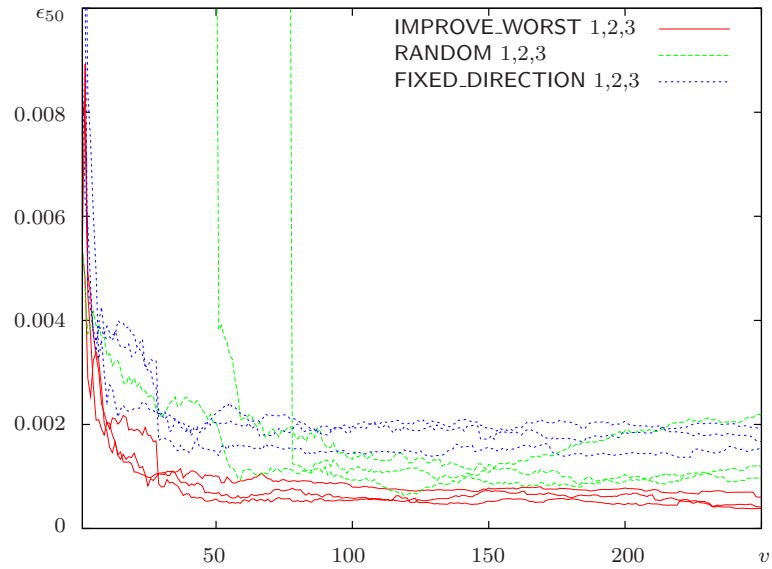
(a) Evaluation of uncertainties logλ_Σ. IMPROVE_WORST provides lowest uncertainty values and steepest descent.

(b) Evaluation of 3D errors logϵ in meters. IMPROVE_WORST provides smallest errors and most narrow error distribution.

Figure 7.9: Comparing the worst run of IMPROVE_WORST with the best ones of RANDOM and FIXED_DIRECTION.



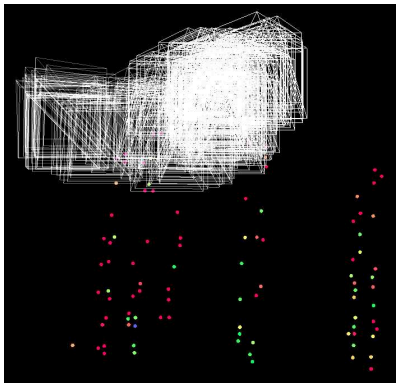
(a) Evaluation of median uncertainties $\log\lambda_{\Sigma 50}$. Curves according to IMPROVE_WORST are compact at the low end.



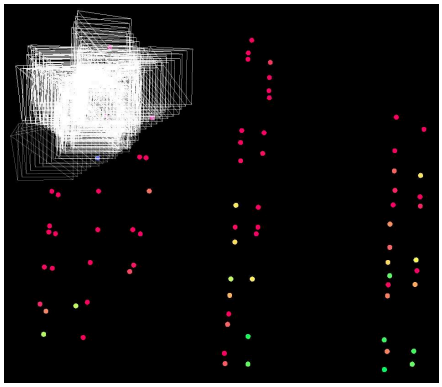
(b) Evaluation of median 3D errors ϵ_{50} in meters. View planning regarding strategy IMPROVE_WORST outperforms non-planning strategies, curves level out around 0.5mm.

Figure 7.10: Comparing median values of all runs from all strategies.

results. The uncertainty values are strictly decreasing and the median 3D errors reach approximately 0.5mm. Regarding the strategy RANDOM in Figure 7.7, the results feature a larger spread. While the uncertainties also decrease with a growing number of frames, the uncertainty differences between the single runs are large. The absolute error values appear as chaotic. There are extremely large accuracy differences especially for the lower frame numbers. Additionally, the error values are even increasing at the end of two of the three runs. As expected for the general case, decreasing uncertainty values do not correspond to decreasing absolute errors for the RANDOM strategy. This is similar for the FIXED_DIRECTION strategy shown in Figure 7.8. The uncertainties provide decreasing progress, but the absolute 3D errors stagnate at a high level or even increase. In Figure 7.9 we compare the worst performing planned run IMPROVE_WORST 1 with the best ones of the other strategies. The uncertainties depicted in Figure 7.9(a) reveal that the planning strategy yields a lower uncertainty and a steeper descent. Regarding the absolute 3D errors in Figure 7.9(b), the strategy IMPROVE_WORST manages to achieve a higher accuracy and a more narrow error distribution than the other, non-planning strategies. The strict improvement starts right after the random initialization and levels out with only small oscillation. This behavior is again outlined in Figure 7.10, where we comprehend the median values of all runs from all strategies. The uncertainty curves of the three runs of IMPROVE_WORST are compactly settled at the low end of the whole field of uncertainty values. As for accuracy judgment, we set the graph in Figure 7.10(b) back to non-log scaling, which displays zero level and the true relations. Hence, we state that all the planned runs achieve a better accuracy than all the other ones for nearly each single view. After 250 views, view planning according to IMPROVE_WORST reaches median 3D errors between 0.4mm and 0.7mm, RANDOM performs between 1mm and 2.1mm, and FIXED_DIRECTION yields between 1.5mm and 2mm. In comprehension of these tests, the simple planning strategy IMPROVE_WORST outperforms the non-planning strategies by simply optimizing the camera movement with respect to the worst point estimate. Regarding runtimes, one cycle of image acquisition, feature tracking, and camera movement requires around 6s, which is independent of the used strategy. Most of this time is spent for moving the robotic arm, network traffic, and feature tracking. The calculations of the actual planning are a matter of milliseconds.



(a) 3D scene achieved by strategy IMPROVE_MEAN.



(b) 3D scene achieved by strategy IMPROVE_CLUSTER.

Figure 7.11: 3D scenes achieved with different strategies using 250 frames each. Observed scene as in Figure 7.5(a).

7.2.3 Evaluating Different Planning Strategies

The previous section outlines the accuracy gain reached by view planning with respect to EEC regarding the respective worst point. We may now state the question which other suitable ways there are to apply accuracy optimization with respect to EEC. Section 6.3.2 describes the required input data as the center of uncertainty and the main direction of uncertainty. In addition to the strategy IMPROVE_WORST, we now evaluate two other ways to provide the input data for accuracy optimization with respect to EEC. Examples of resulting 3D scenes including camera paths are shown in Figure 7.11. The strategy IMPROVE_MEAN takes all points into account and establishes the center and main direction of uncertainty as the weighted sums combining the terms of each single point. The weights are given by the uncertainty ratings λ_Σ of the point estimates. An intermediate way is pursued by the strategy IMPROVE_CLUSTER. Here, we extract the uncertainty data from a subset of point estimates containing the worst one. The elements of this focused cluster need to answer both a spatial and an uncertainty requirement. Regarding the uncertainty, the point estimates in the cluster feature an uncertainty rating larger than the median uncertainty. In the spatial domain, we run k-means clustering using 10 clusters and take the cluster containing the worst point. In combination, we select the point estimates of the spatial cluster containing the worst point that also feature

an uncertainty rating worse than the median uncertainty. We consider the elements of this final cluster to compute weighted means of the uncertainty center and main direction. So, the additional planning strategies are:

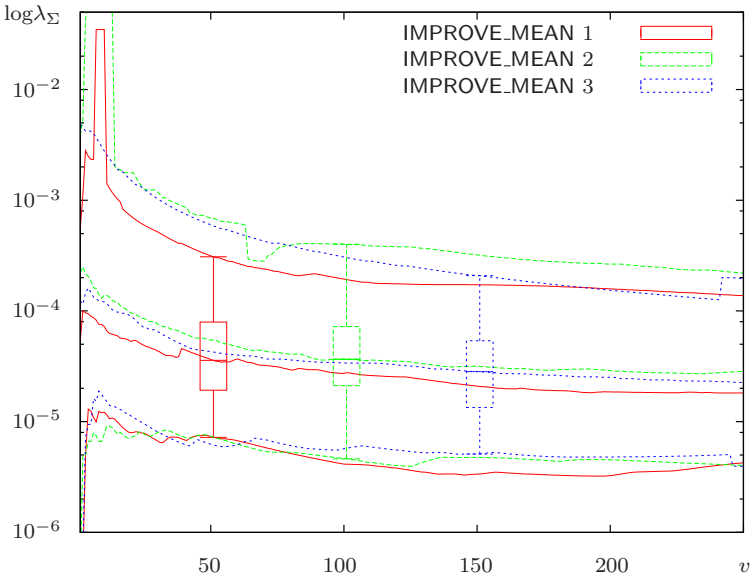
IMPROVE_MEAN View planning for accuracy optimization w.r.t. EEC. Uncertainty data provided by weighted mean of all points.

IMPROVE_CLUSTER View planning for accuracy optimization w.r.t. EEC. Uncertainty data provided by weighted mean of point cluster that comprises uncertain points in the same spatial region.

The numerical results for three runs using the strategy IMPROVE_MEAN are plotted in Figure 7.12. The measured uncertainties feature similar, high levels and fail to provide a continuous, steep descent. This observation is supported by the error curves in Figure 7.12(b) illustrating unstable behavior with even increasing errors. To relate the strategy to RANDOM and IMPROVE_WORST, Figure 7.13 shows the respective median values together. It is shown that the three runs following IMPROVE_MEAN perform even worse than RANDOM, while IMPROVE_WORST is still the best. There seems to be no obvious reason for these poor results of IMPROVE_MEAN. On the one side, the dominating weight of the respective worst point should also dominate the movement criteria. On the other side, the mass of better points may have too much influence on the combined uncertainty data and blur in an unfavorable manner. In addition, the quasi-static nature of the uncertainty measures combined over all points favors a camera movement that is quasi-regular and hence close to the weak-performing, regular strategy FIXED_DIRECTION.

We employ the strategy IMPROVE_CLUSTER to exclude the good point estimates from the combined uncertainty measure and present the numerical results of three runs in Figure 7.14. The uncertainty curves perform a descent steeper than for IMPROVE_MEAN, and the error curves are more stable. In Figure 7.15(b), the comparison with RANDOM and IMPROVE_WORST displays error curves that are better than for RANDOM and close to the ones of IMPROVE_WORST, which still performs best. The tests suggest that view planning with respect to EEC works better if the uncertainty data is derived from a preferably small set of the most uncertain points, which culminates in using just the worst point.

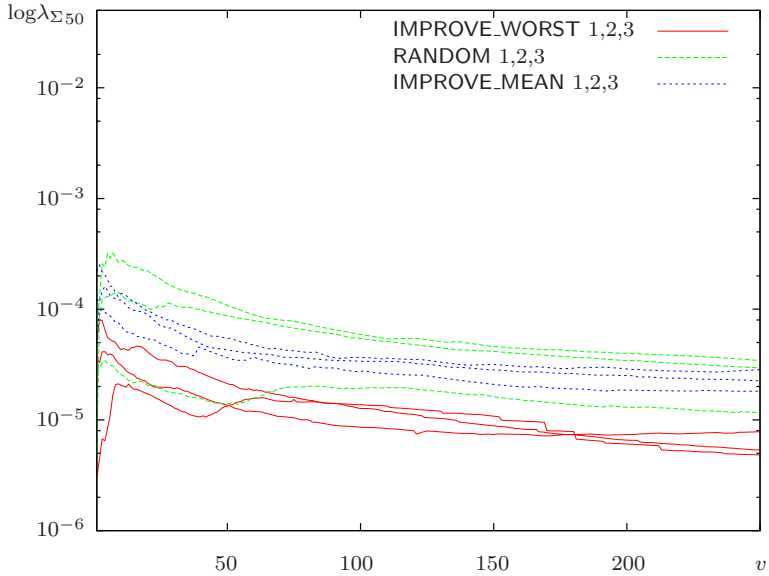
In conclusion, the recommendation to use the strategy IMPROVE_WORST just holds for the general planning goal of accuracy optimization regarding



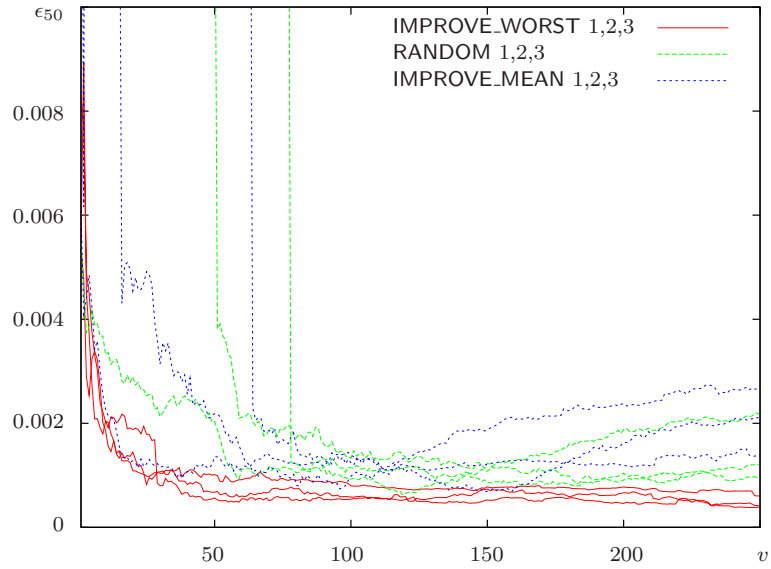
(a) Evaluation of uncertainties logλ_Σ.

(b) Evaluation of 3D errors logϵ in meters.

Figure 7.12: Numerical results for three runs using view planning according to IMPROVE_MEAN.

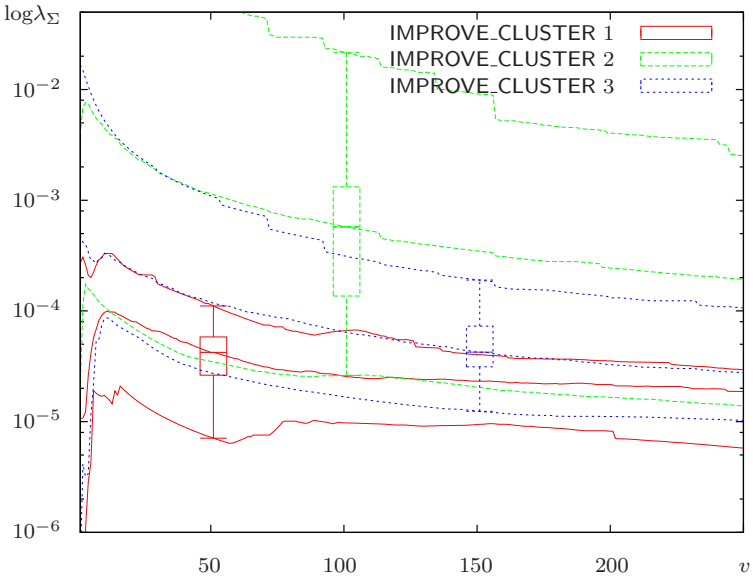


(a) Evaluation of median uncertainties log$\lambda_{\Sigma 50}$. Curves according to IMPROVE_MEAN are close to RANDOM curves with only smooth descent.



(b) Evaluation of median 3D errors ϵ_{50} in meters. View planning using strategy IMPROVE_MEAN performs even worse than RANDOM.

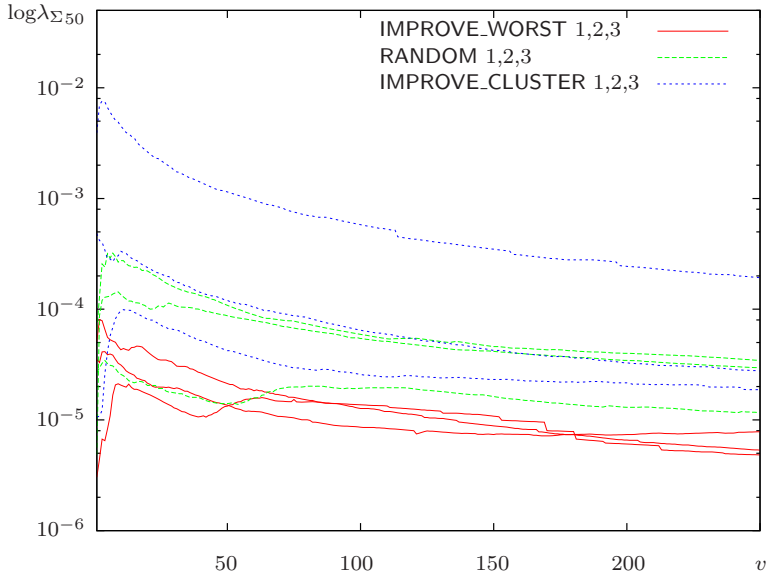
Figure 7.13: Comparing median values of all runs of IMPROVE_MEAN with RANDOM and IMPROVE_WORST.



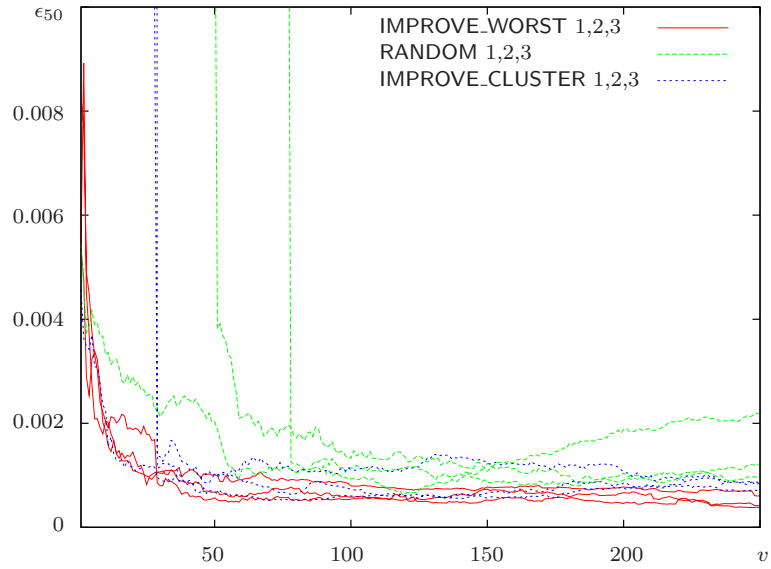
(a) Evaluation of uncertainties logλ_Σ.

(b) Evaluation of 3D errors logϵ in meters.

Figure 7.14: Numerical results for three runs using view planning according to IMPROVE_CLUSTER.



(a) Evaluation of median uncertainties log$\lambda_{\Sigma 50}$. Curves according to IMPROVE_CLUSTER feature wide spread but continuous descent.



(b) Evaluation of median 3D errors ϵ_{50} in meters. View planning using strategy IMPROVE_CLUSTER performs better than RANDOM.

Figure 7.15: Comparing median values of all runs of IMPROVE_CLUSTER with RANDOM and IMPROVE_WORST.

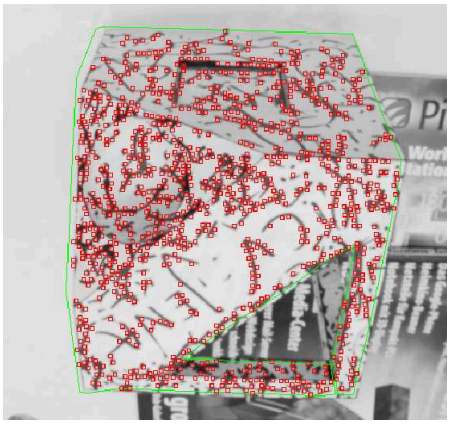
all points. In the end, the actual specific goal needs to be represented by the particular strategy used. For instance, we may aim to further improve the best points or a special subset of points, or just to guarantee a maximal uncertainty level for all points. The performance of the strategy IMPROVE_WORST underlines that view planning with respect to the extended E-criterion provides effective means for accuracy optimization. Since this strategy focuses on one point, it is an elementary component allowing to address various specific types of view planning for accuracy optimization.

7.3 Assessing the Probabilistic Surface Estimation

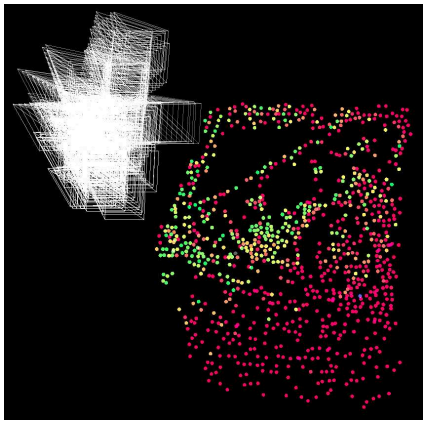
In Section 6.4 we presented a way to concurrently establish a surface triangulation within the online planning approach, where each triangle features a probability to be part of the actual surface to be reconstructed. Applying the benchmarking criteria developed in Section 6.1, we now assess a resulting surface. For this experiment, the probabilistic estimation of a surface triangulation uses the following parameters. After the initial stage of 15 frames, we consider the 0.75-low-uncertainty quantile of point estimates and define the static voxelspace according to the 3D bounding box of these points. The resolution of the voxelspace is set to 100 voxels along the dimension providing the maximal extent. After the voxelspace is being established, we map each voxel to each frame and update the variances of the intensity values that are the basis for the surface probability of a single voxel. After every 50 frames, we perform the whole surface estimation as outlined in Section 6.4, i. e. we run the complete 3D Delaunay triangulation, filter out impossible triangles that would have covered an observed point, and assess the surface probabilities of the remaining triangles based on the surface probabilities of neighboring voxels.

Since this is a somehow awkward situation of evaluating both reconstruction result and the evaluation methodology, we first render reference data for comparison using only reconstructed 3D points without surface information. To this end we compute the benchmark criteria for accuracy in terms of 3D errors ϵ, the mean distance of nearest neighbors μ_d, and the coverage c. According to the statements regarding completeness in Section 6.1, we compute μ_d using reconstructed points and, using the 3D CAD model, derive a

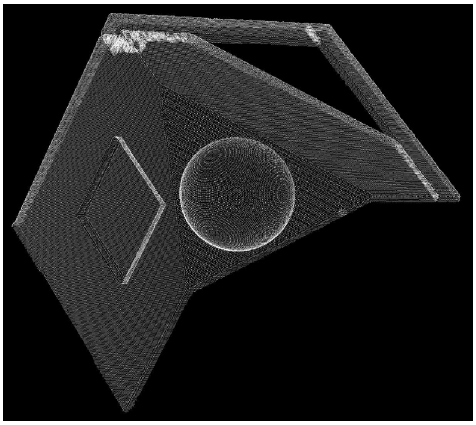
homogeneous reference point cover that shall provide the same value of μ_d. We hence achieve a benchmark rating employing only the reconstructed 3D points without surface triangulation. Afterwards, we realize our suggestion for evaluating non-discrete surface information by evaluating sampled points.



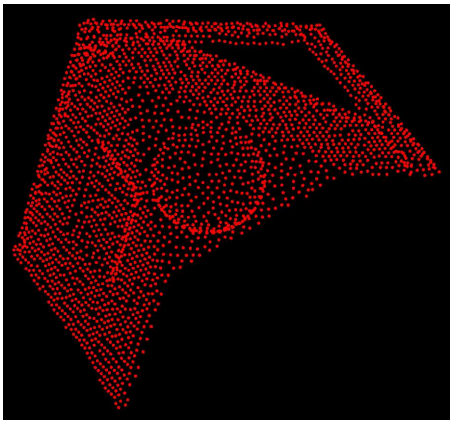
(a) First frame showing NBV object and 1012 features selected.



(b) Points of the 3D reconstruction after 350 frames, $\mu_d = 3.2039$mm.



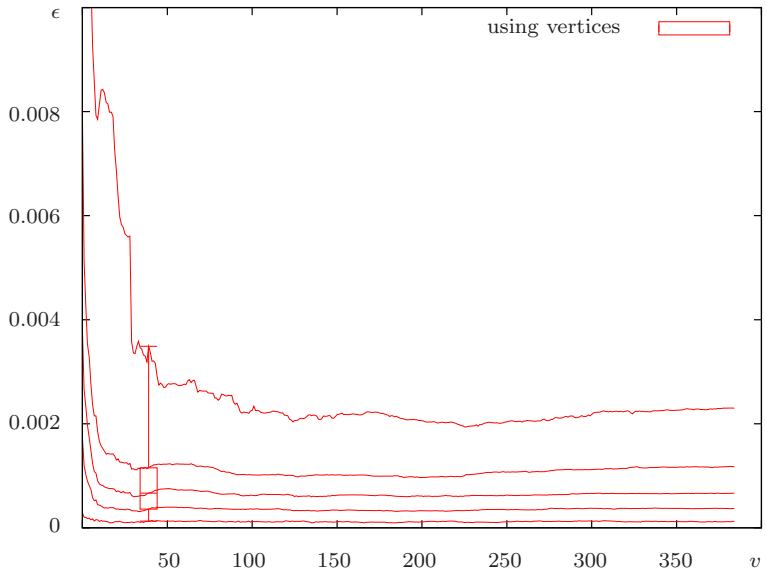
(c) Dense point cover of the CAD model of the NBV object featuring $\mu_d = 0.3966$mm.



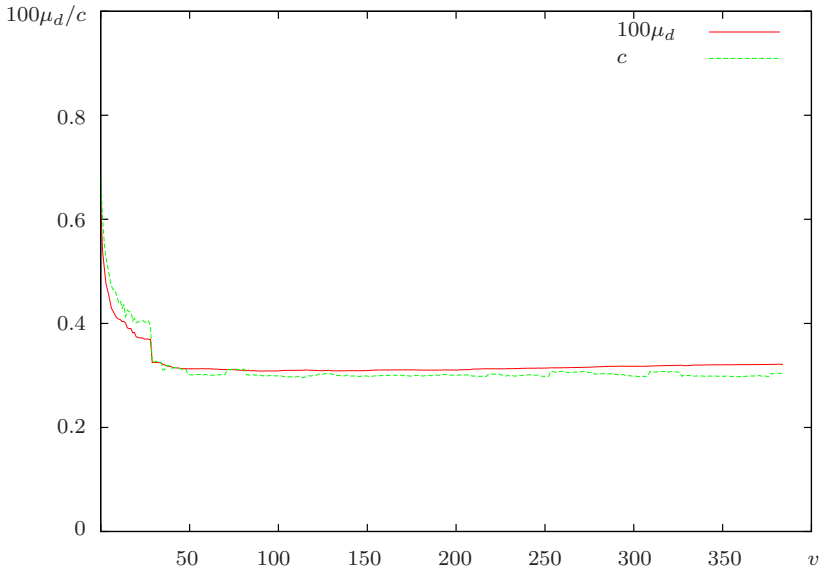
(d) Homegeneous point cover statistically derived from the dense one, $\mu_d = 3.1974$mm.

Figure 7.16: Reconstruction example and completeness reference feature the same mean distance of nearest neighbors μ_d. Surface information not used.

The first steps of computing reference data, in which we just refer to the reconstructed 3D points, are visualized in Figure 7.16. Determining the



(a) Evaluation of 3D errors of reconstructed points in meters, i. e. only vertices of the reconstructed surface triangulation are used. Median error levels out at 0.6mm.

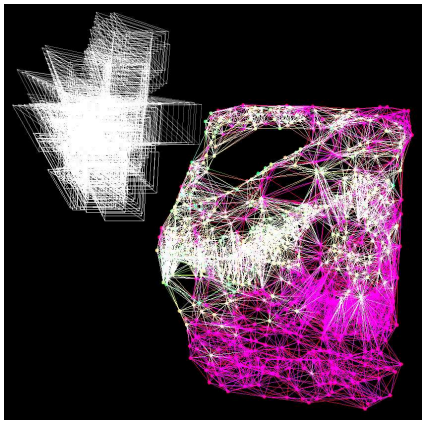


(b) Evaluation of the mean distance of nearest neighbors μ_d in meters and the coverage c. Reconstruction quickly reaches a steady state at 3.2mm for μ_d and 0.3 for c.

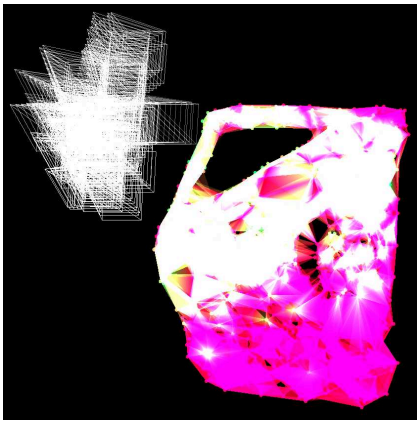
Figure 7.17: Benchmarking accuracy and completeness using only reconstructed 3D points, i. e. the vertices of the estimated 3D surface triangulation.

mean distance of nearest neighbors μ_d of the reconstruction in Figure 7.16(b) provides a measure of resolution. A large value of μ_d corresponds to a low resolution, and vice versa. Further, we use this measure μ_d to statistically derive a homogeneous point cover of the reference CAD model with the same property as shown in Figure 7.16(d). To yield meaningful results, we selected the relevant part of the CAD model that actually may be covered by the reconstruction. This enables us to yield the proportion of model points that are covered by reconstructed points, which is the coverage c. The quantitative results of this first step of evaluation are shown in Figure 7.17. Since we employ the optimization method IMPROVE_WORST, we again observe a strict reduction of the reconstruction error in Figure 7.17(a). By the completeness evaluation in Figure 7.17(b) it is underlined that the reconstruction rather quickly shows stable behavior in the measured terms. After the first 50 frames, where the largest error reduction takes place, we find nearly constant values of resolution $\mu_d \approx 3.2$mm and coverage $c \approx 0.3$. While the resolution value is just given by the reconstructed points, also the coverage seems reasonable when comparing Figures 7.16(b) and 7.16(d). The tendency of the reconstruction to form tight clusters, which is given by the feature selection, both yields a small mean distance of nearest neighbors and relatively large gaps between these clusters. Compared with a homogeneous reference point cover featuring the same mean distance of nearest neighbors, it is obvious that a large amount of reference points will not be covered by a reconstructed point.

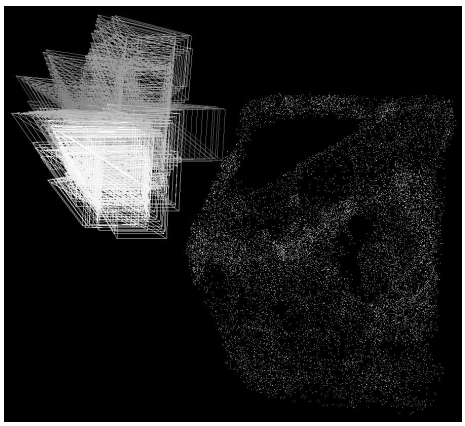
As the second part, we now include the estimated, non-discrete surface information as depicted in the Figures 7.18(a) and 7.18(b). According to the NBV benchmarking scheme we refer to the 3D triangles of the estimated surface triangulation by resampling 3D points. We consider all triangles featuring a surface probability larger than zero. For the process of resampling we choose a target number of points for the whole surface and then randomly resample from each single triangle regarding the proportion of the triangle area. In general, we round down the number of points for each triangle. However, triangles that would be assigned zero points provide one resampled point, finally. It is this strategy that generates numbers of resampled points as in the Figures 7.18(c) and 7.18(d), where we were aiming at 10^4 and 10^5 points, respectively. Regarding the quantitative assessment in Figure 7.19 where we aim to resample each reconstruction with 10^4 points, the single values are identical to the ones for vertices in Figure 7.17 for frame numbers smaller than 50. This is due to the fact that the surface probabilities of all



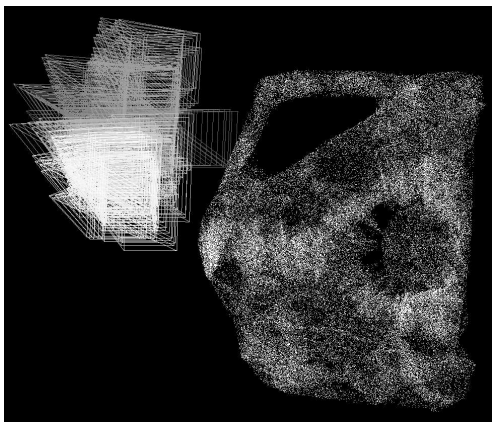
(a) Reconstruction after 350 frames. Lines indicate all 17791 triangles of the probabilistic surface estimation.



(b) Same reconstruction with visualized triangles. Probability of single triangle is coded by transparency in the viewport. Overlaying triangles cause coloring towards white.



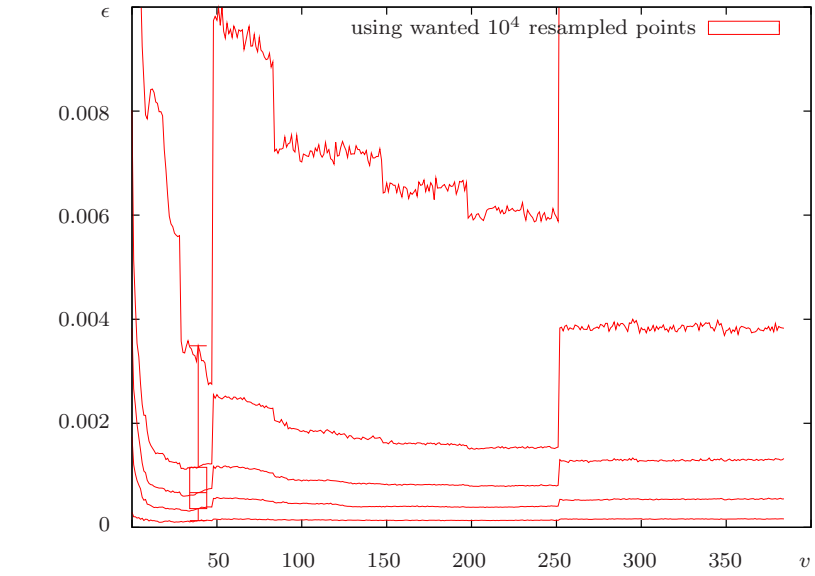
(c) Random point cover of the estimated surface with 14549 points, aimed at 10^4 points.



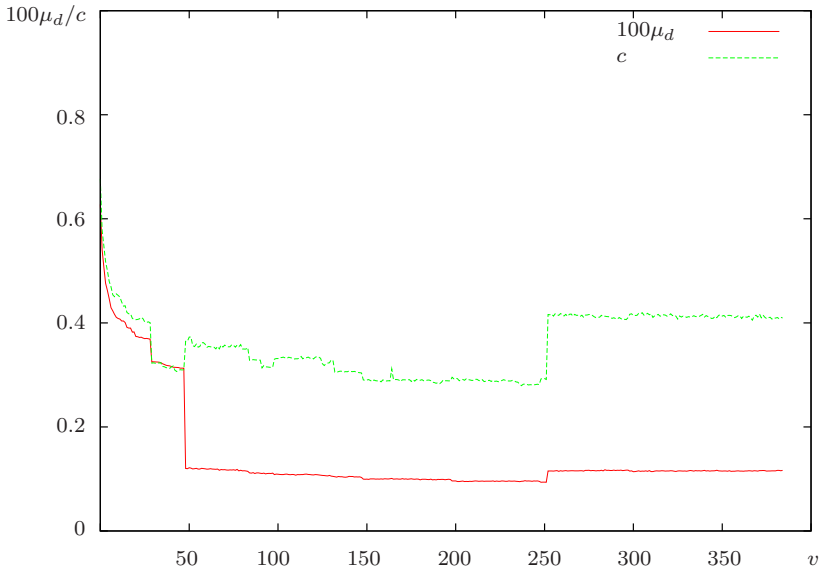
(d) Random point cover of the estimated surface with 93963 points, aimed at 10^5 points.

Figure 7.18: Reconstruction example and resampled surfaces used for evaluating the estimated surface.

triangles are evaluated every 50 frames in this experiment. With an actual surface estimation being present, the 3D errors increase. While in some period the median errors lie only slightly above the vertex-based values, the



(a) Evaluation of 3D errors of randomly resampled points in meters. Errors are larger than without resampling.



(b) Evaluation of the mean distance of nearest neighbors μ_d in meters and the coverage c. Resolution and coverage are both larger than without resampling.

Figure 7.19: Benchmarking accuracy and completeness using wanted 10^4 randomly resampled points of the reconstructed surface triangulation in Figure 7.18(b) and triangles with a surface probability larger than zero. First surface estimation is triggered at frame 50. Results vary according to the continuously updated surface probabilities of voxels.

larger error quantiles ϵ_{75} and $\epsilon_{97.5}$ show a significant gain. This tells us that the mass of resampled points is close to the actual object surface, but some of them are not. A direct implication is that also the mass of the estimated triangles is close to the actual object surface $\mathcal{S}$. Regarding the large error quantiles, we again note that all triangles T with $P(\mathsf{T} \in \mathcal{S}) > 0$ are used for resampling. Resampling more points than vertices makes us expecting a lower μ_d and a thus increased resolution, which is confirmed by the plot in Figure 7.19(b). However, it may be surprising that the increase in coverage is not more than around 10%. This last property is due to several reasons. First, our resampling strategy is not strictly designed for creating a homogeneously distributed set of points, which could be improved independent of the measurement process. Second, the process of measurement and surface estimation, by design, establishes a set of 3D triangles being attributed by a surface probability. Without further processing, this general construct allows multiple layers of triangles in the same surface region. So, for instance, depending on the reconstructed vertices, a planar surface region will be represented by several layers of 3D triangles stack upon each other. Such an unwanted effect will cause a low coverage value.

Table 7.1: Resampling of the reconstruction in Figure 7.18(b) using $\mathsf{N_R}$ points. Regard all triangles with surface probability $P(\mathsf{T} \in \mathcal{S}) > 0$. Using more points, errors increase, resolution increases, but coverage remains nearly constant.

N_R	$\epsilon_{2.5}$ (mm)	ϵ_{25} (mm)	ϵ_{50} (mm)	ϵ_{75} (mm)	$\epsilon_{97.5}$ (mm)	μ_d (mm)	c $\in [0, 1]$
14549	0.16	0.53	1.29	3.82	14.11	1.15	0.41
26578	0.18	0.71	1.93	5.48	15.25	0.96	0.39
49019	0.19	0.86	2.37	6.11	15.46	0.77	0.39
72697	0.19	0.89	2.51	6.29	15.67	0.67	0.40
96869	0.19	0.91	2.55	6.28	15.55	0.61	0.40
vertices	0.12	0.37	0.66	1.16	2.26	3.20	0.30

One of the first questions raised by the preceding evaluation is how the number of resampled points influences the benchmark. To answer this question, we focus on the stable reconstruction after 350 frames and resample the reconstructed surface using all triangles with $P(\mathsf{T} \in \mathcal{S}) > 0$ and different numbers N_R of resampled points, and we present the benchmark results in Table 7.1. Unsurprisingly, an increasing number of resampled points yields

an decreasing mean distance of nearest neighbors μ_d. However, the coverage value cannot be influenced by a simple resampling and remains nearly constant. Furthermore, the increasing number N_R emphasizes the presence of triangles that are not part of the actual object surface $\mathcal{S}$ by boosting the upper error quantiles.

Table 7.2: Resampling of the reconstruction in Figure 7.18(b). Regard all $\mathsf{N_T}$ triangles featuring surface probability $P(\mathsf{T} \in \mathcal{S}) > t_{SP}$. Excluding triangles with low probability slightly improves accuracy.

t_{SP}	N_T	$\epsilon_{2.5}$ (mm)	ϵ_{25} (mm)	ϵ_{50} (mm)	ϵ_{75} (mm)	$\epsilon_{97.5}$ (mm)	μ_d (mm)	c $\in [0,1]$
0.00	11587	0.16	0.53	1.29	3.82	14.11	1.15	0.41
0.25	10685	0.16	0.53	1.23	3.55	14.15	1.21	0.41
0.50	8106	0.16	0.51	1.18	3.44	13.92	1.30	0.35
0.75	4515	0.15	0.49	1.15	3.46	13.87	1.56	0.31
0.99	1422	0.17	0.51	1.19	3.80	12.98	2.22	0.21

A further issue nearly ignored up to now is the surface probability $P(\mathsf{T} \in \mathcal{S})$ of a triangle T. We now refer to this probability in the following manner. Selecting a minimal surface probability t_{SP}, we use only the triangles with a surface probability higher than t_{SP} for resampling. In order to avoid distortion effects caused by different relative point densities, we aim to keep the number of resampled points constant for each triangle throughout this experiment. Practically, we consider the accumulated area of all triangles, which we relate to a wanted number of 10^4 resampled points, and the accumulated area of triangles filtered by applying a minimal surface probability of t_{SP}. As Table 7.2 shows, there is only a slight improvement of the accuracy when excluding triangles with low surface probability. These results again underline that the estimated surface triangulation is bound to the vertices, which are the reconstructed 3D points. The triangles depend on the vertices, and the vertices depend on the selection of image features, which depend on the optical structure of the object surface. And so does the consistency checking of the triangles. While the result in Figure 7.18 illustrates a reasonable surface estimation, there are triangles within the result that deviate from the true object surface, but that are hard to exclude based on the actual image data. Further work is necessary to create an actual single-layer surface of the object. Regarding completeness issues, it is natural that an exclusion of

triangles accompanied by a reduced overall number of points leads to reduced resolution and coverage.

Regarding runtime requirements, we note computation times for the above example of application. The machine employed as planning host is a Core2 Duo with 2.4GHz and 8 GB RAM. The further hardware is dedicated to the robot control and image acquisition and consists of a Pentium IV providing a hardware interface to the control unit of the robotic arm and camera connectivity. According to details of the procedure, we use a static voxel space containing $100 \times 100 \times 81$ voxels, the sampling of which takes 0.2s per image. As for the step from voxels to triangles, we establish a complete Delaunay triangulation yielding 6050 tetrahedrons and 12142 unique triangles, which takes 8.4s. Filtering out triangles that would cover an observed point takes, in this example, 6.1s per view. This is to say that for each view it takes around 6s to check for each estimated 1012 3D point if it is covered by one of the 12142 triangles in the tested view. For the reason of small baselines, we perform this kind of visibility checking only for every tenth view. The actual surface probabilities for the remaining 8819 possible triangles are estimated based on the surface probabilities of the voxels within 0.5s. We find it a suitable strategy to run the voxel sampling for each frame, but use a certain increment for the rest of the surface analysis. In the above experiment we used an increment of 50 frames for updating the triangles and the according surface probabilities. The whole planning run took approximately two hours for 384 frames.

7.4 Applying the Probabilistic Surface Estimation

We have presented a probabilistic estimation of surface triangulations that serves as a basic module for retrieving non-discrete surface information. As the result we yield a set of 3D triangles with attached probabilities to belong to the object surface. Now, we show ways to apply this surface information within our view planning approach.

7.4.1 Visibility Analysis

Computing the visibility of a point given the camera position is a crucial as well as elementary part in many planning tasks. The proposed surface

estimation does achieve this visibility analysis and additionally provides a coverage probability as in (6.44).

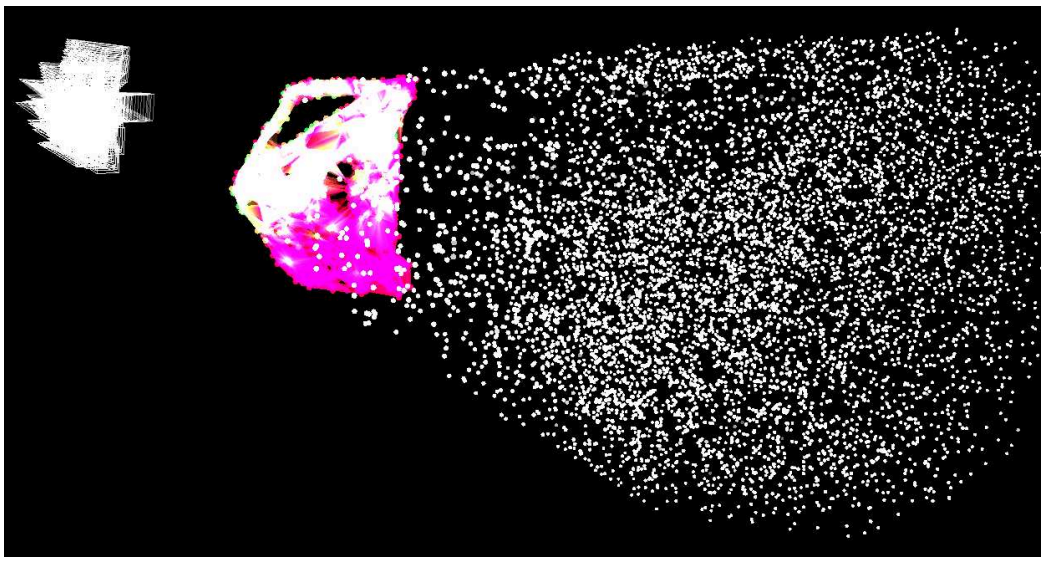


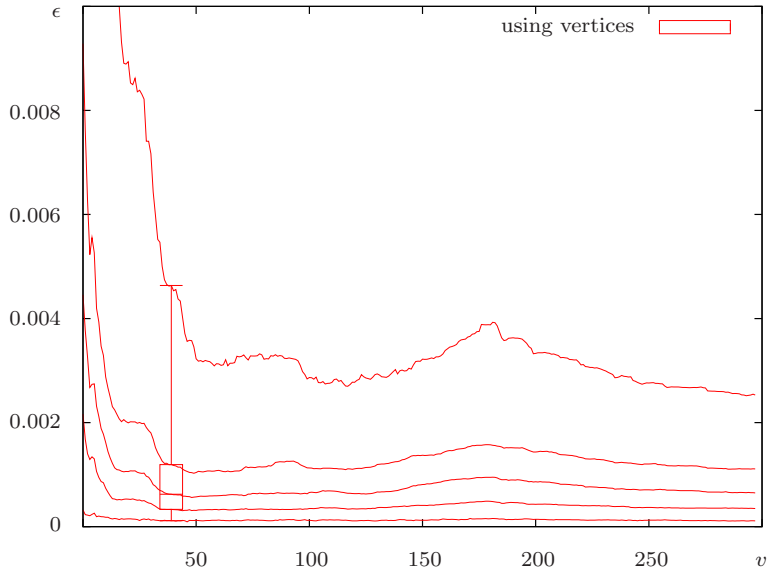
Figure 7.20: 3D scene, i. e. cameras and estimated surface, as in Figure 7.18(b). Additional points are random samples that have not been observed by any camera, which is decided using the estimated surface. The depicted samples hence describe the unseen part of the measurement volume.

The practical realization of the visibility analysis is based on simple geometric calculations. Given a camera position and the point in question, we compute the intersection points of the ray of sight and all surface triangles. For all existing intersection points we check the order along the ray of sight and, in case of coverage, note the maximal surface probability of the covering triangles. To illustrate the effectivity of this approach, we use the reconstructed 3D scene shown in Figure 7.18(b) and compute the visibility of random point samples. As Figure 7.20 highlights, the visualized point samples are not seen by any of the cameras. Consequently, these samples describe the unseen part of the measurement volume, which provides a hint for further planning steps. Trivially, this kind of visibility analysis is also valid for avoiding self-occlusions during the planning procedure.

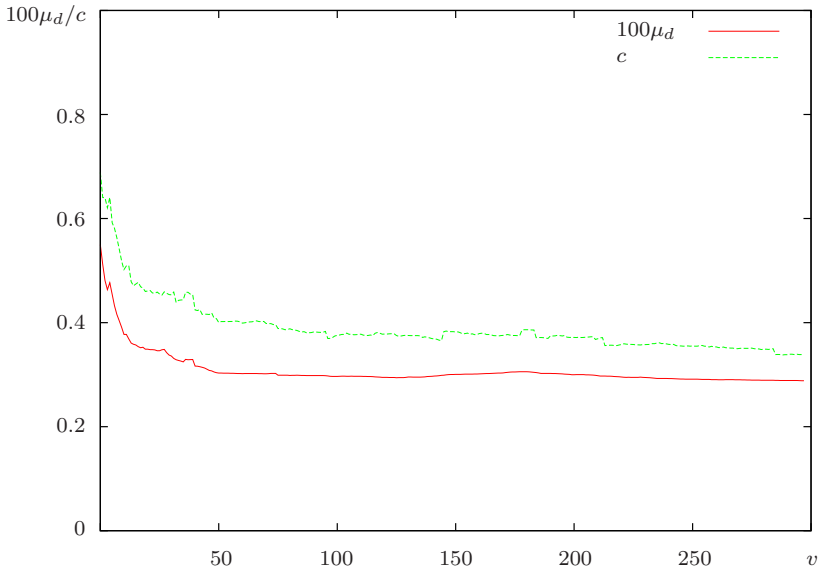
7.4.2 View Planning Using Surface Normals

A criterion used for view planning in many other works dealing with range scanners is sensing perpendicular to the surface to be reconstructed. This criterion optimizes reflectivity of projected signals as well as the sampling resolution on the surface. Coming back to passive cameras and feature tracking, scanning along the surface normal still produces a favorable spatial resolution of the surface within the feature patch used for tracking. It is another question if this is really an advantage when the initial reference patch has been sampled from an acute angle. Nevertheless, we implement this criterion employing the concurrent surface estimation in the following manner. We again refer to the current worst point and consider the spherical neighborhood of this point. For this experiment we use a diameter of 5% of the maximal diameter of the reconstructed object. Any triangle of the surface estimation that has one point within this neighborhood of the worst point is considered for the local surface normal. Starting with the normals defined by the selected triangles, the final approximation of the local surface normal is the weighted mean of these, whereat one weight is the product of the triangle's relative size and its surface probability. Hence, the normals contributed by large triangles with high surface probabilities are dominating. With respect to the strategy IMPROVE_WORST applied beforehand, we now do not regard the uncertainty of the point estimation but strictly refer to the estimated local surface normal around this point.

The quantitative evaluation presented in Figure 7.21 demonstrates that using surface normals is one way to improve the 3D reconstruction. However, we note the documented possibility of a more unstable behavior compared with the strategy IMPROVE_WORST. Here are the boundaries of the current state of the surface estimation. The whole method does, at this point of time, not allow to run the complete surface estimation for each frame. Consequently, it may happen that the optimization follows the hints provided by a surface estimation the is not up to date.



(a) Evaluation of 3D errors of randomly resampled points in meters.



(b) Evaluation of the mean distance of nearest neighbors μ_d in meters and the coverage c.

Figure 7.21: Benchmarking accuracy and completeness considering the vertices of an estimated surface. The optimization strategy is moving the camera to sense the worst point preferably along the local surface normal. While the 3D errors are mainly decreasing, the reduction is not as straight and stable as the one achieved by IMPROVE_WORST.

Chapter 8

Conclusion

We conclude this thesis by summarizing the main contributions and suggesting topics for future research.

8.1 Summary

In this work we examine the topic of next-best-view planning for 3D reconstruction, whereat we focus on using an intensity camera without active illumination. We start by identifying the necessity of view planning and well-defined reconstruction goals when using a controlled environment, and we establish a problem formalization. Since the investigated, formal-theoretic approaches to solve the abstract problem of view planning are not promising, we settle for a modular, online approach respecting small baselines given by feature tracking. After reviewing relevant literature and computer vision basics, we present the Guided KLT feature tracking as one module that is important for our view planning approach. Based on the well-known KLT feature tracking, Guided KLT tracking incorporates known camera parameters into the tracking process while regarding uncertainty. In the final stage of extension the method additionally performs concurrent, robust 3D estimation of a tracked feature. Guided KLT tracking hence uses additional knowledge to improve tracking accuracy and duration, which is outlined by the experimental results. In general, this work is concerned with data-driven view planning, i. e. we do not state assumptions about the object shape or use a prior model of the object to be reconstructed. We just rely on the observed data. However, we also present a method for 3D coarse registration

of surface triangulations that has applications to model-based view planning. The basic step of this registration technique is the computation of local invariant features of surface triangulation based on invariants of area moments. To this end we compute exact surface moments of 3D surface triangulations that yield direct surface features without sampling or projection. As the according experiments emphasize, the presented features allow 3D coarse registration of surface triangulations as well as handling tasks of classification, identification, and clustering. Regarding view planning, the method thus provides means to identify, for instance, special partial surfaces that require a special view plan. As the view planning framework we propose a modular, online system performing cycles of camera movement, image acquisition, and a complex consisting of data analysis and actual view planning as required. Our main planning focus is directed towards accuracy optimization. We theoretically motivate and implement an optimal solution for accuracy optimization with respect to the Extended E-criterion that we define in this work. The experimental evaluation shows an efficient strategy of applying the presented accuracy optimization. Despite the fact that solving completeness issues is limited when employing a passive camera, we further present a concurrent, probabilistic surface estimation that achieves a non-discrete surface estimation of the object to reconstruct. By means of this surface estimation we present visibility analysis and motivate to tackle model-based view planning. Finally, we investigate view planning focusing on the reconstruction method of factorization. Based on the identification of factorization aspects that are crucial in theory, we conduct simulations and draw conclusions for according view planning approaches.

8.2 Future Work

The methods presented here provide, amongst others, theoretical motivation, a formal framework, and a generic procedural approach to online next-best-view planning. We further propose extended feature tracking and 3D coarse registration with applications to view planning and we solve specific planning tasks. In doing so, we touch several fields of computer vision and pattern recognition and hence prepare to extend this work towards different directions. With regard to Guided KLT tracking and 3D coarse registration of surface triangulations, the reader may be referred to the individual outlooks in Sections 4.6 and 5.6, respectively. In consideration of our generic, modu-

lar, online approach to view planning we want to sketch some ideas for future research.

To begin with, all the runtimes noted regard research source code that is not overambitiously optimized with respect to computation time and memory usage. While this is mainly relevant for practical application, some of the presented methods and results may be improved by using more efficient code and more powerful hardware. For instance, the surface estimation may benefit from using octrees for organizing points and triangles. Hence, it would be suitable to run the probabilistic surface estimation in each planning cycle and thereby boost its accuracy.

The presented module for accuracy optimization allows to cover this specific aspect of view planning quite well, since it is independent of the entity to optimize as long as the required input is given. Another question is implementing the actual planning strategy that may be enriched by additional paradigms. In this context we already outlined catching and ignoring points when indicated. Other ideas worth investigating may be different clustering approaches regarding the optimization of a certain cluster of points. Further, we suggest to introduce a smoothing paradigm into the accuracy optimization. This can be done, for instance, by generally improving the worst point, but selecting the new worst point just if the current one is not any longer amongst the 10% worst ones. By this means we avoid an alternating declaration of the two worst points as the entities to optimize, which may in special cases lead to a null optimization.

With regard to completeness issues and non-discrete surface estimation, we proposed basic methods to establish a set of 3D triangles accompanied by the respective surface probabilities. The online result provides information important for subsequent planning steps and yields a reasonable representation of the object surface. However, at least two open aspects remain. First, further research may deal with actual completeness optimization, which may included closing holes, scanning unseen regions, and detecting new feature points in certain regions. These steps require an extended management of points to reconstruct, since some points will intentionally not be seen and are hence not to be caught or ignored. Second, the current final solution features a lack of beauty. As we expect a closed, single-layer representation of the actual object surface, further steps are necessary to build an according surface based on the provided estimation and image data.

Factorization for 3D reconstruction is one particular method that is worth being investigated with respect to view planning. In doing so, we detected

several dependencies between theoretical aspects of factorization and consequent view planning requirements. On the one side, future research may again focus on factorization to better understand possible data interdependencies and their effects on the result, which includes entering the level of numerical methods. On the other side, any method used for view planning may carefully be examined for their relation to a view planning framework. This is to say that all these methods may use additional information given by the controlled environment and may provide optimization criteria for view planning regarding this focused method. For example, applying the presented GKLT tracking, which does already use additional prior knowledge, provides the planning optimization criterion of sensing preferably perpendicular to the local surface region. The possible impact on the resulting reconstruction should be taken into account. Based on the planning goals and the knowledge about the particular methods, view planning should find a parameter adjustment favoring an optimal common behavior of the sub-methods providing the largest combined positive impact on the relevant properties of the reconstruction result.

8.3 Acknowledgment

To start with, this work was made possible by grant DE 735/5-2 of the Deutsche Forschungsgemeinschaft (`http://www.dfg.de`), which we highly appreciate.

In person, the author is deeply grateful to his advisor, Prof. Dr.-Ing. J. Denzler, and to Dr. habil. H. Süße for their guidance as well as priceless theoretical and practical advice. Further all-important, sincere thanks are given to all the colleagues at the Chair for Computer Vision of the Friedrich-Schiller University of Jena, who were an inspiring and reliable company and made work fun to do.

Appendix A

Derivatives of Warping Functions

Regarding the explanations on Guided KLT tracking from Chapter 4, we find the Jacobian of the respective warping function an important component of the optimal update rule, for instance, in (4.26). The actual derivatives with respect to the warping parameters depend on the formulation of the warping function and, of course, on the parameterization. In the following, we note the Jacobians for common warping functions and parameters of KLT and GKLT tracking.

A.1 Warping Functions Used with the Original KLT Tracking

The most simple form of the warping function is given as pure translation of image positions $\mathbf{x} = (x, y)^\mathsf{T}$,

$$W^t(\mathbf{x}, \mathbf{p}^t) = \begin{pmatrix} W_x^t(\mathbf{x}, \mathbf{p}^t) \\ W_y^t(\mathbf{x}, \mathbf{p}^t) \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x + \Delta x \\ y + \Delta y \end{pmatrix} \tag{A.1}$$

using $\mathbf{p}^t = (\Delta x, \Delta y)^\mathsf{T}$. The according Jacobian is

$$\nabla_{\mathbf{p}^t} W^t(\mathbf{x}, \mathbf{p}^t) = \begin{pmatrix} \frac{\partial W_x^t(\mathbf{x}, \mathbf{p}^t)}{\partial \Delta x} & \frac{\partial W_x^t(\mathbf{x}, \mathbf{p}^t)}{\partial \Delta y} \\ \frac{\partial W_y^t(\mathbf{x}, \mathbf{p}^t)}{\partial \Delta x} & \frac{\partial W_y^t(\mathbf{x}, \mathbf{p}^t)}{\partial \Delta y} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{A.2}$$

For affine warping, which is the most common choice for the warping function, we get

$$\begin{aligned} W^a(\mathbf{x}, \mathbf{p}^a) &= \begin{pmatrix} W_x^a(\mathbf{x}, \mathbf{p}^a) \\ W_y^a(\mathbf{x}, \mathbf{p}^a) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= \begin{pmatrix} a_{11}x + a_{12}y + \Delta x \\ a_{21}x + a_{22}y + \Delta y \end{pmatrix} \end{aligned} \tag{A.3}$$

with $\mathbf{p}^a = (\Delta x, \Delta y, a_{11}, a_{12}, a_{21}, a_{22})^\mathsf{T}$ and the Jacobian

$$\begin{aligned} &\nabla_{\mathbf{p}^a} W^a(\mathbf{x}, \mathbf{p}^a) \\ &= \begin{pmatrix} \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial \Delta x} & \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial \Delta y} & \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{11}} & \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{12}} & \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{21}} & \frac{\partial W_x^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{22}} \\ \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial \Delta x} & \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial \Delta y} & \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{11}} & \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{12}} & \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{21}} & \frac{\partial W_y^a(\mathbf{x},\mathbf{p}^a)}{\partial a_{22}} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{pmatrix}. \end{aligned} \tag{A.4}$$

It is rarely done, but possible to select a general 2D projective transformation as warping function

$$W^p(\mathbf{x}, \mathbf{p}^p) = \begin{pmatrix} W_x^p(\mathbf{x}, \mathbf{p}^p) \\ W_y^p(\mathbf{x}, \mathbf{p}^p) \end{pmatrix} = \begin{pmatrix} \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + 1} \\ \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + 1} \end{pmatrix} \tag{A.5}$$

using projective parameters $\mathbf{p}^p = (p_{11}, p_{12}, \ldots, p_{31}, p_{32})^\mathsf{T}$. The according Jacobian follows as

$$\begin{aligned} &\nabla_{\mathbf{p}^p} W^p(\mathbf{x}, \mathbf{p}^p) = \begin{pmatrix} \frac{\partial W_x^p(\mathbf{x},\mathbf{p}^p)}{\partial p_{11}} & \cdots & \frac{\partial W_x^p(\mathbf{x},\mathbf{p}^p)}{\partial p_{32}} \\ \frac{\partial W_y^p(\mathbf{x},\mathbf{p}^p)}{\partial p_{11}} & \cdots & \frac{\partial W_y^p(\mathbf{x},\mathbf{p}^p)}{\partial p_{32}} \end{pmatrix} \\ &= \frac{1}{p_{31}x + p_{32}y + 1} \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -\frac{(p_{11}x + p_{12}y + p_{13})x}{p_{31}x + p_{32}y + 1} & -\frac{(p_{11}x + p_{12}y + p_{13})y}{p_{31}x + p_{32}y + 1} \\ 0 & 0 & 0 & x & y & 1 & -\frac{(p_{21}x + p_{22}y + p_{23})x}{p_{31}x + p_{32}y + 1} & -\frac{(p_{21}x + p_{22}y + p_{23})y}{p_{31}x + p_{32}y + 1} \end{pmatrix}. \end{aligned} \tag{A.6}$$

A.2 Warping Functions for GKLT Tracking

Compared to the original KLT, GKLT tracking uses translation parameters along and perpendicular to the respective epipolar line,

$$\begin{aligned} W^t_{EU}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m}) &= \begin{pmatrix} W^t_{EU,x}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m}) \\ W^t_{EU,y}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m}) \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\frac{l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \\ &= \begin{pmatrix} x - \frac{l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ y + \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \end{aligned} \tag{A.7}$$

using $\mathbf{p}^t_{EU} = (\lambda_1, \lambda_2)^\mathsf{T}$, $(l_1, l_2, l_3)^\mathsf{T} = \mathbf{F}\tilde{\mathbf{m}}$, fundamental matrix $\mathbf{F}$, and $l_1 \neq 0$. Accordingly, we get the Jacobian

$$\nabla_{\mathbf{p}^t_{EU}} W^t_{EU}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m}) = \begin{pmatrix} \frac{\partial W^t_{EU,x}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m})}{\partial \lambda_1} & \frac{\partial W^t_{EU,x}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m})}{\partial \lambda_2} \\ \frac{\partial W^t_{EU,y}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m})}{\partial \lambda_1} & \frac{\partial W^t_{EU,y}(\mathbf{x}, \mathbf{p}^t_{EU}, \mathbf{m})}{\partial \lambda_2} \end{pmatrix} = \begin{pmatrix} -l_2 & l_1 \\ l_1 & l_2 \end{pmatrix}. \tag{A.8}$$

The affine, epipolar warping function with respect to uncertainty is

$$\begin{aligned} W^a_{EU}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m}) &= \begin{pmatrix} W^a_{EU,x}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m}) \\ W^a_{EU,y}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m}) \end{pmatrix} \\ &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\frac{l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11} x + a_{12} y - \frac{l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ a_{21} x + a_{22} y + \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \end{aligned} \tag{A.9}$$

with $\mathbf{p}^a_{EU} = (\lambda_1, \lambda_2, a_{11}, a_{12}, a_{21}, a_{22})^\mathsf{T}$ and the Jacobian

$$\begin{aligned} &\nabla_{\mathbf{p}^a_{EU}} W^a_{EU}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m}) \\ &= \begin{pmatrix} \frac{\partial W^a_{EU,x}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial \lambda_1} & \frac{\partial W^a_{EU,x}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial \lambda_2} & \frac{\partial W^a_{EU,x}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial a_{11}} & \cdots & \frac{\partial W^a_{EU,x}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial a_{22}} \\ \frac{\partial W^a_{EU,y}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial \lambda_1} & \frac{\partial W^a_{EU,y}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial \lambda_2} & \frac{\partial W^a_{EU,y}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial a_{11}} & \cdots & \frac{\partial W^a_{EU,y}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m})}{\partial a_{22}} \end{pmatrix} \\ &= \begin{pmatrix} -l_2 & l_1 & x & y & 0 & 0 \\ l_1 & l_2 & 0 & 0 & x & y \end{pmatrix}. \end{aligned} \tag{A.10}$$

For the case of a general 2D homography, it is not possible to isolate the translational part of the transformation. So for now, GKLT tracking uses warping functions up to affine transformations.

Appendix B

3D Surface Moments

In Section 5.4.1 we derive an efficient and direct way to compute moments M_{kln} of 3D surface triangulations. The basis is the computation of the respective moments m_{kln} for a single 3D triangle T defined by corner points, or vertices, $\mathbf{X}_c = (X_c, Y_c, Z_c)^\mathsf{T}$, $1 \leq c \leq 3$. In (5.15) we evolve that this computation makes use of the 2D area moments m'_{pq} listed in (5.11). We further use the constant C regarding the parameterization in (5.8),

$$C = \sqrt{EG - F^2} \tag{B.1}$$

with

$$E = X_u^2 + Y_u^2 + Z_u^2 = (X_1 - X_3)^2 + (Y_1 - Y_3)^2 + (Z_1 - Z_3)^2, \tag{B.2}$$
$$G = X_v^2 + Y_v^2 + Z_v^2 = (X_2 - X_3)^2 + (Y_2 - Y_3)^2 + (Z_2 - Z_3)^2, \tag{B.3}$$
$$\begin{aligned} F &= X_u X_v + Y_u Y_v + Z_u Z_v \\ &= (X_1 - X_3)(X_2 - X_3) + (Y_1 - Y_3)(Y_2 - Y_3) + (Z_1 - Z_3)(Z_2 - Z_3). \end{aligned} \tag{B.4}$$

For the first two moments we explicitly note the steps (5.12) to (5.15).

Using the parameterization (5.8), the zero-order moment is

$$m_{000} = C \iint_{\mathsf{D}} \left[(u(X_1 - X_3) + v(X_2 - X_3) + X_3)^0 (u(Y_1 - Y_3) + v(Y_2 - Y_3) + Y_3)^0 \right.$$
$$\left. (u(Z_1 - Z_3) + v(Z_2 - Z_3) + Z_3)^0 \right] \mathrm{d}u\,\mathrm{d}v \tag{B.5}$$
$$= C \iint_{\mathsf{D}} \mathrm{d}u\,\mathrm{d}v \tag{B.6}$$
$$= C m'_{00} \tag{B.7}$$
$$= \frac{1}{2} C. \tag{B.8}$$

We observe that (B.8) expresses the dependence on the triangle corner points by the term C. The first-order moment with respect to the X-coordinate yields

$$m_{100} = C \iint_{\mathsf{D}} \left[(u(X_1 - X_3) + v(X_2 - X_3) + X_3)^1 (u(Y_1 - Y_3) + v(Y_2 - Y_3) + Y_3)^0 \right.$$
$$\left. (u(Z_1 - Z_3) + v(Z_2 - Z_3) + Z_3)^0 \right] \mathrm{d}u\,\mathrm{d}v \tag{B.9}$$
$$= C \iint_{\mathsf{D}} (u(X_1 - X_3) + v(X_2 - X_3) + X_3)\,\mathrm{d}u\,\mathrm{d}v \tag{B.10}$$
$$= C \left[\iint_{\mathsf{D}} u(X_1 - X_3)\,\mathrm{d}u\,\mathrm{d}v + \iint_{\mathsf{D}} v(X_2 - X_3)\,\mathrm{d}u\,\mathrm{d}v + \iint_{\mathsf{D}} X_3\,\mathrm{d}u\,\mathrm{d}v \right] \tag{B.11}$$
$$= C \left[(X_1 - X_3) \iint_{\mathsf{D}} u\,\mathrm{d}u\,\mathrm{d}v + (X_2 - X_3) \iint_{\mathsf{D}} v\,\mathrm{d}u\,\mathrm{d}v + X_3 \iint_{\mathsf{D}} \mathrm{d}u\,\mathrm{d}v \right] \tag{B.12}$$
$$= C \left[(X_1 - X_3) m'_{10} + (X_2 - X_3) m'_{01} + X_3 m'_{00} \right]. \tag{B.13}$$
$$= C \left[\frac{1}{6}(X_1 - X_3) + \frac{1}{6}(X_2 - X_3) + \frac{1}{2} X_3 \right]. \tag{B.14}$$

Further first-order moments are

$$m_{010} = C \left[\frac{1}{6}(Y_1 - Y_3) + \frac{1}{6}(Y_2 - Y_3) + \frac{1}{2} Y_3 \right] \tag{B.15}$$

and

$$m_{001} = C\left[\frac{1}{6}(Z_1 - Z_3) + \frac{1}{6}(Z_2 - Z_3) + \frac{1}{2}Z_3\right]. \tag{B.16}$$

Proceeding to second order, we find

$$\begin{aligned} m_{200} = C\Bigg[&\frac{1}{12}(X_1 - X_3)^2 + \frac{1}{12}(X_2 - X_3)^2 + \frac{1}{12}(X_1 - X_3)(X_2 - X_3) \\ &+\frac{1}{3}(X_1 - X_3)X_3 + \frac{1}{3}(X_2 - X_3)X_3 + \frac{1}{2}X_3^2\Bigg], \end{aligned} \tag{B.17}$$

$$\begin{aligned} m_{020} = C\Bigg[&\frac{1}{12}(Y_1 - Y_3)^2 + \frac{1}{12}(Y_2 - Y_3)^2 + \frac{1}{12}(Y_1 - Y_3)(Y_2 - Y_3) \\ &+\frac{1}{3}(Y_1 - Y_3)Y_3 + \frac{1}{3}(Y_2 - Y_3)Y_3 + \frac{1}{2}Y_3^2\Bigg], \end{aligned} \tag{B.18}$$

$$\begin{aligned} m_{002} = C\Bigg[&\frac{1}{12}(Z_1 - Z_3)^2 + \frac{1}{12}(Z_2 - Z_3)^2 + \frac{1}{12}(Z_1 - Z_3)(Z_2 - Z_3) \\ &+\frac{1}{3}(Z_1 - Z_3)Z_3 + \frac{1}{3}(Z_2 - Z_3)Z_3 + \frac{1}{2}Z_3^2\Bigg], \end{aligned} \tag{B.19}$$

and

$$\begin{aligned} m_{110} = C\Bigg[&\frac{1}{12}(X_1 - X_3)(Y_1 - Y_3) + \frac{1}{12}(X_2 - X_3)(Y_2 - Y_3) \\ &+\frac{1}{24}((X_1 - X_3)(Y_2 - Y_3) + (X_2 - X_3)(Y_1 - Y_3)) \\ &+\frac{1}{6}((X_1 - X_3)Y_3 + (Y_1 - Y_3)X_3) + \frac{1}{6}((X_2 - X_3)Y_3 + (Y_2 - Y_3)X_3) \\ &+\frac{1}{2}X_3Y_3\Bigg], \end{aligned} \tag{B.20}$$

$$m_{101} = C\left[\frac{1}{12}(X_1 - X_3)(Z_1 - Z_3) + \frac{1}{12}(X_2 - X_3)(Z_2 - Z_3)\right.$$
$$+\frac{1}{24}((X_1 - X_3)(Z_2 - Z_3) + (X_2 - X_3)(Z_1 - Z_3))$$
$$+\frac{1}{6}((X_1 - X_3)Z_3 + (Z_1 - Z_3)X_3) + \frac{1}{6}((X_2 - X_3)Z_3 + (Z_2 - Z_3)X_3)$$
$$\left.+\frac{1}{2}X_3Z_3\right], \tag{B.21}$$

$$m_{011} = C\left[\frac{1}{12}(Y_1 - Y_3)(Z_1 - Z_3) + \frac{1}{12}(Y_2 - Y_3)(Z_2 - Z_3)\right.$$
$$+\frac{1}{24}((Y_1 - Y_3)(Z_2 - Z_3) + (Y_2 - Y_3)(Z_1 - Z_3))$$
$$+\frac{1}{6}((Y_1 - Y_3)Z_3 + (Z_1 - Z_3)Y_3) + \frac{1}{6}((Y_2 - Y_3)Z_3 + (Z_2 - Z_3)Y_3)$$
$$\left.+\frac{1}{2}Y_3Z_3\right]. \tag{B.22}$$

Finally, the third-order 3D surface moments of a 3D triangle are

$$m_{300} = C\left[\frac{1}{20}(X_1 - X_3)^3 + \frac{1}{20}(X_2 - X_3)^3 + \frac{1}{20}(X_1 - X_3)^2(X_2 - X_3)\right.$$
$$+\frac{1}{20}(X_1 - X_3)(X_2 - X_3)^2 + \frac{1}{4}(X_1 - X_3)^2X_3 + \frac{1}{4}(X_2 - X_3)^2X_3$$
$$+\frac{1}{4}(X_1 - X_3)(X_2 - X_3)X_3 + \frac{1}{2}(X_1 - X_3)X_3^2 + \frac{1}{2}(X_2 - X_3)X_3^2$$
$$\left.+\frac{1}{2}X_3^3\right], \tag{B.23}$$

$$m_{030} = C \left[\frac{1}{20}(Y_1 - Y_3)^3 + \frac{1}{20}(Y_2 - Y_3)^3 + \frac{1}{20}(Y_1 - Y_3)^2(Y_2 - Y_3) \right.$$
$$+\frac{1}{20}(Y_1 - Y_3)(Y_2 - Y_3)^2 + \frac{1}{4}(Y_1 - Y_3)^2 Y_3 + \frac{1}{4}(Y_2 - Y_3)^2 Y_3$$
$$+\frac{1}{4}(Y_1 - Y_3)(Y_2 - Y_3)Y_3 + \frac{1}{2}(Y_1 - Y_3)Y_3^2 + \frac{1}{2}(Y_2 - Y_3)Y_3^2$$
$$\left. +\frac{1}{2}Y_3^3 \right], \tag{B.24}$$

$$m_{003} = C \left[\frac{1}{20}(Z_1 - Z_3)^3 + \frac{1}{20}(Z_2 - Z_3)^3 + \frac{1}{20}(Z_1 - Z_3)^2(Z_2 - Z_3) \right.$$
$$+\frac{1}{20}(Z_1 - Z_3)(Z_2 - Z_3)^2 + \frac{1}{4}(Z_1 - Z_3)^2 Z_3 + \frac{1}{4}(Z_2 - Z_3)^2 Z_3$$
$$+\frac{1}{4}(Z_1 - Z_3)(Z_2 - Z_3)Z_3 + \frac{1}{2}(Z_1 - Z_3)Z_3^2 + \frac{1}{2}(Z_2 - Z_3)Z_3^2$$
$$\left. +\frac{1}{2}Z_3^3 \right], \tag{B.25}$$

$$
\begin{aligned}
m_{012} = C \Bigg[& \frac{1}{20}(Y_1 - Y_3)(Z_1 - Z_3)^2 + \frac{1}{20}(Y_2 - Y_3)(Z_2 - Z_3)^2 \\
& + \frac{1}{60}\left(2(Y_1 - Y_3)(Z_1 - Z_3)(Z_2 - Z_3) + (Y_2 - Y_3)(Z_1 - Z_3)^2\right) \\
& + \frac{1}{60}\left(2(Z_1 - Z_3)(Y_2 - Y_3)(Z_2 - Z_3) + (Y_1 - Y_3)(Z_2 - Z_3)^2\right) \\
& + \frac{1}{12}\left(2(Y_1 - Y_3)(Z_1 - Z_3)Z_3 + (Z_1 - Z_3)^2 Y_3\right) \\
& + \frac{1}{12}\left(2(Y_2 - Y_3)(Z_2 - Z_3)Z_3 + (Z_2 - Z_3)^2 Y_3\right) \\
& + \frac{1}{24}(2(Y_1 - Y_3)(Z_2 - Z_3)Z_3 + 2(Z_1 - Z_3)(Y_2 - Y_3)Z_3 \\
& \qquad + 2(Z_1 - Z_3)(Z_2 - Z_3)Y_3) \\
& + \frac{1}{6}\left((Y_1 - Y_3)Z_3^2 + 2(Z_1 - Z_3)Y_3 Z_3\right) \\
& + \frac{1}{6}\left((Y_2 - Y_3)Z_3^2 + 2(Z_2 - Z_3)Y_3 Z_3\right) + \frac{1}{2}Y_3 Z_3^2 \Bigg],
\end{aligned}
\tag{B.26}
$$

$$
\begin{aligned}
m_{102} = C \Bigg[& \frac{1}{20}(X_1 - X_3)(Z_1 - Z_3)^2 + \frac{1}{20}(X_2 - X_3)(Z_2 - Z_3)^2 \\
& + \frac{1}{60}\left(2(X_1 - X_3)(Z_1 - Z_3)(Z_2 - Z_3) + (X_2 - X_3)(Z_1 - Z_3)^2\right) \\
& + \frac{1}{60}\left(2(Z_1 - Z_3)(X_2 - X_3)(Z_2 - Z_3) + (X_1 - X_3)(Z_2 - Z_3)^2\right) \\
& + \frac{1}{12}\left(2(X_1 - X_3)(Z_1 - Z_3)Z_3 + (Z_1 - Z_3)^2 X_3\right) \\
& + \frac{1}{12}\left(2(X_2 - X_3)(Z_2 - Z_3)Z_3 + (Z_2 - Z_3)^2 X_3\right) \\
& + \frac{1}{24}(2(X_1 - X_3)(Z_2 - Z_3)Z_3 + 2(Z_1 - Z_3)(X_2 - X_3)Z_3 \\
& \qquad + 2(Z_1 - Z_3)(Z_2 - Z_3)X_3) \\
& + \frac{1}{6}\left((X_1 - X_3)Z_3^2 + 2(Z_1 - Z_3)X_3 Z_3\right) \\
& + \frac{1}{6}\left((X_2 - X_3)Z_3^2 + 2(Z_2 - Z_3)X_3 Z_3\right) + \frac{1}{2}X_3 Z_3^2 \Bigg],
\end{aligned}
\tag{B.27}
$$

$$
\begin{aligned}
m_{120} = C \Bigg[& \frac{1}{20}(X_1 - X_3)(Y_1 - Y_3)^2 + \frac{1}{20}(X_2 - X_3)(Y_2 - Y_3)^2 \\
& + \frac{1}{60}\left(2(X_1 - X_3)(Y_1 - Y_3)(Y_2 - Y_3) + (X_2 - X_3)(Y_1 - Y_3)^2\right) \\
& + \frac{1}{60}\left(2(Y_1 - Y_3)(X_2 - X_3)(Y_2 - Y_3) + (X_1 - X_3)(Y_2 - Y_3)^2\right) \\
& + \frac{1}{12}\left(2(X_1 - X_3)(Y_1 - Y_3)Y_3 + (Y_1 - Y_3)^2 X_3\right) \\
& + \frac{1}{12}\left(2(X_2 - X_3)(Y_2 - Y_3)Y_3 + (Y_2 - Y_3)^2 X_3\right) \\
& + \frac{1}{24}(2(X_1 - X_3)(Y_2 - Y_3)Y_3 + 2(Y_1 - Y_3)(X_2 - X_3)Y_3 \\
& \qquad + 2(Y_1 - Y_3)(Y_2 - Y_3)X_3) \\
& + \frac{1}{6}\left((X_1 - X_3)Y_3^2 + 2(Y_1 - Y_3)X_3 Y_3\right) \\
& + \frac{1}{6}\left((X_2 - X_3)Y_3^2 + 2(Y_2 - Y_3)X_3 Y_3\right) + \frac{1}{2}X_3 Y_3^2 \Bigg],
\end{aligned}
\tag{B.28}
$$

$$
\begin{aligned}
m_{021} = C \Bigg[& \frac{1}{20}(Z_1 - Z_3)(Y_1 - Y_3)^2 + \frac{1}{20}(Z_2 - Z_3)(Y_2 - Y_3)^2 \\
& + \frac{1}{60}\left(2(Z_1 - Z_3)(Y_1 - Y_3)(Y_2 - Y_3) + (Z_2 - Z_3)(Y_1 - Y_3)^2\right) \\
& + \frac{1}{60}\left(2(Y_1 - Y_3)(Z_2 - Z_3)(Y_2 - Y_3) + (Z_1 - Z_3)(Y_2 - Y_3)^2\right) \\
& + \frac{1}{12}\left(2(Z_1 - Z_3)(Y_1 - Y_3)Y_3 + (Y_1 - Y_3)^2 Z_3\right) \\
& + \frac{1}{12}\left(2(Z_2 - Z_3)(Y_2 - Y_3)Y_3 + (Y_2 - Y_3)^2 Z_3\right) \\
& + \frac{1}{24}(2(Z_1 - Z_3)(Y_2 - Y_3)Y_3 + 2(Y_1 - Y_3)(Z_2 - Z_3)Y_3 \\
& \qquad + 2(Y_1 - Y_3)(Y_2 - Y_3)Z_3) \\
& + \frac{1}{6}\left((Z_1 - Z_3)Y_3^2 + 2(Y_1 - Y_3)Y_3 Z_3\right) \\
& + \frac{1}{6}\left((Z_2 - Z_3)Y_3^2 + 2(Y_2 - Y_3)Y_3 Z_3\right) + \frac{1}{2}Y_3^2 Z_3 \Bigg],
\end{aligned}
\tag{B.29}
$$

$$
\begin{aligned}
m_{201} = C \Bigg[& \frac{1}{20}(Z_1 - Z_3)(X_1 - X_3)^2 + \frac{1}{20}(Z_2 - Z_3)(X_2 - X_3)^2 \\
& + \frac{1}{60}\left(2(Z_1 - Z_3)(X_1 - X_3)(X_2 - X_3) + (Z_2 - Z_3)(X_1 - X_3)^2\right) \\
& + \frac{1}{60}\left(2(X_1 - X_3)(Z_2 - Z_3)(X_2 - X_3) + (Z_1 - Z_3)(X_2 - X_3)^2\right) \\
& + \frac{1}{12}\left(2(Z_1 - Z_3)(X_1 - X_3)X_3 + (X_1 - X_3)^2 Z_3\right) \\
& + \frac{1}{12}\left(2(Z_2 - Z_3)(X_2 - X_3)X_3 + (X_2 - X_3)^2 Z_3\right) \\
& + \frac{1}{24}(2(Z_1 - Z_3)(X_2 - X_3)X_3 + 2(X_1 - X_3)(Z_2 - Z_3)X_3 \\
& \qquad + 2(X_1 - X_3)(X_2 - X_3)Z_3) \\
& + \frac{1}{6}\left((Z_1 - Z_3)X_3^2 + 2(X_1 - X_3)X_3 Z_3\right) \\
& + \frac{1}{6}\left((Z_2 - Z_3)X_3^2 + 2(X_2 - X_3)X_3 Z_3\right) + \frac{1}{2}X_3^2 Z_3 \Bigg],
\end{aligned}
\tag{B.30}
$$

$$
\begin{aligned}
m_{210} = C \Bigg[& \frac{1}{20}(Y_1 - Y_3)(X_1 - X_3)^2 + \frac{1}{20}(Y_2 - Y_3)(X_2 - X_3)^2 \\
& + \frac{1}{60}\left(2(Y_1 - Y_3)(X_1 - X_3)(X_2 - X_3) + (Y_2 - Y_3)(X_1 - X_3)^2\right) \\
& + \frac{1}{60}\left(2(X_1 - X_3)(Y_2 - Y_3)(X_2 - X_3) + (Y_1 - Y_3)(X_2 - X_3)^2\right) \\
& + \frac{1}{12}\left(2(Y_1 - Y_3)(X_1 - X_3)X_3 + (X_1 - X_3)^2 Y_3\right) \\
& + \frac{1}{12}\left(2(Y_2 - Y_3)(X_2 - X_3)X_3 + (X_2 - X_3)^2 Y_3\right) \\
& + \frac{1}{24}(2(Y_1 - Y_3)(X_2 - X_3)X_3 + 2(X_1 - X_3)(Y_2 - Y_3)X_3 \\
& \qquad + 2(X_1 - X_3)(X_2 - X_3)Y_3) \\
& + \frac{1}{6}\left((Y_1 - Y_3)X_3^2 + 2(X_1 - X_3)X_3 Y_3\right) \\
& + \frac{1}{6}\left((Y_2 - Y_3)X_3^2 + 2(X_2 - X_3)X_3 Y_3\right) + \frac{1}{2}X_3^2 Y_3 \Bigg],
\end{aligned}
\tag{B.31}
$$

$$
\begin{aligned}
m_{111} = C \Bigg[& \frac{1}{20}(X_1 - X_3)(Y_1 - Y_3)(Z_1 - Z_3) + \frac{1}{20}(X_2 - X_3)(Y_2 - Y_3)(Z_2 - Z_3) \\
& + \frac{1}{60}((X_1 - X_3)(Y_1 - Y_3)(Z_2 - Z_3) + (X_1 - X_3)(Y_2 - Y_3)(Z_1 - Z_3) \\
& \qquad + (X_2 - X_3)(Y_1 - Y_3)(Z_1 - Z_3)) \\
& + \frac{1}{60}((X_1 - X_3)(Y_2 - Y_3)(Z_2 - Z_3) + (X_2 - X_3)(Y_1 - Y_3)(Z_2 - Z_3) \\
& \qquad + (X_2 - X_3)(Y_2 - Y_3)(Z_1 - Z_3)) \\
& + \frac{1}{12}((X_1 - X_3)(Y_1 - Y_3)Z_3 + (X_1 - X_3)(Z_1 - Z_3)Y_3 \\
& \qquad + (Y_1 - Y_3)(Z_1 - Z_3)X_3) \\
& + \frac{1}{12}((X_2 - X_3)(Y_2 - Y_3)Z_3 + (X_2 - X_3)(Z_2 - Z_3)Y_3 \\
& \qquad + (Y_2 - Y_3)(Z_2 - Z_3)X_3) \\
& + \frac{1}{24}((X_1 - X_3)(Y_2 - Y_3)Z_3 + (X_1 - X_3)(Z_2 - Z_3)Y_3 \\
& \qquad + (X_2 - X_3)(Y_1 - Y_3)Z_3 + (X_2 - X_3)(Z_1 - Z_3)Y_3 \\
& \qquad + (Y_1 - Y_3)(Z_2 - Z_3)X_3 + (Y_2 - Y_3)(Z_1 - Z_3)X_3) \\
& + \frac{1}{6}((X_1 - X_3)Y_3Z_3 + (Y_1 - Y_3)X_3Z_3 \\
& \qquad + (Z_1 - Z_3)X_3Y_3) \\
& + \frac{1}{6}((X_2 - X_3)Y_3Z_3 + (Y_2 - Y_3)X_3Z_3 \\
& \qquad + (Z_2 - Z_3)X_3Y_3) + \frac{1}{2}X_3Y_3Z_3 \Bigg] .
\end{aligned}
\tag{B.32}
$$

Appendix C

3D Moment Invariants

The 3D coarse registration of surface triangulations in Chapter 5 employs the 11 3D moment invariants based on 3D volume moments presented in [Lo and Don, 1989]. We first list 11 invariants with respect to Euclidean transformations,

$$I^2_{22} = \nu(2,2)^0_0, \tag{C.1}$$
$$I^2_{222} = \eta\nu, \tag{C.2}$$
$$I^3_{33} = \nu(3,3)^0_0, \tag{C.3}$$
$$I^3_{11} = \nu(1,1)^0_0, \tag{C.4}$$
$$I^{2,3}_{233} = \nu(3,3)_2\nu_2, \tag{C.5}$$
$$I^{2,3}_{123} = \nu(3,1)_2\nu_2, \tag{C.6}$$
$$I^{2,3}_{112} = \nu(1,1)_2\nu_2, \tag{C.7}$$
$$I^3_{3333} = \nu^2(3,3)_2, \tag{C.8}$$
$$I^3_{1333} = \nu(3,3)_2\nu(3,1)_2, \tag{C.9}$$
$$I^3_{1133} = \nu^2(3,1)_2, \tag{C.10}$$
$$I^3_{1113} = \nu(3,1)_2\nu(1,1)_2. \tag{C.11}$$

Adaptations provide the following 11 invariants with respect to similarity

transformations,

$$I_{22}^{2} = \frac{\nu(2,2)_0^0}{(\nu_0^0)^2}, \tag{C.12}$$

$$I_{222}^{2} = \frac{\eta\nu}{(\nu_0^0)^3}, \tag{C.13}$$

$$I_{33}^{3} = \frac{\nu(3,3)_0^0}{(\nu_0^0)^{\frac{12}{5}}}, \tag{C.14}$$

$$I_{11}^{3} = \frac{\nu(1,1)_0^0}{(\nu_0^0)^{\frac{12}{5}}}, \tag{C.15}$$

$$I_{233}^{2,3} = \frac{\nu(3,3)_2\nu_2}{(\nu_0^0)^{\frac{17}{5}}}, \tag{C.16}$$

$$I_{123}^{2,3} = \frac{\nu(3,1)_2\nu_2}{(\nu_0^0)^{\frac{17}{5}}}, \tag{C.17}$$

$$I_{112}^{2,3} = \frac{\nu(1,1)_2\nu_2}{(\nu_0^0)^{\frac{17}{5}}}, \tag{C.18}$$

$$I_{3333}^{3} = \frac{\nu^2(3,3)_2}{(\nu_0^0)^{\frac{24}{5}}}, \tag{C.19}$$

$$I_{1333}^{3} = \frac{\nu(3,3)_2\nu(3,1)_2}{(\nu_0^0)^{\frac{24}{5}}}, \tag{C.20}$$

$$I_{1133}^{3} = \frac{\nu^2(3,1)_2}{(\nu_0^0)^{\frac{24}{5}}}, \tag{C.21}$$

$$I_{1113}^{3} = \frac{\nu(3,1)_2\nu(1,1)_2}{(\nu_0^0)^{\frac{24}{5}}}. \tag{C.22}$$

Now, we provide the moment forms that refer the above invariants to 3D

surface moments M_{kln} up to third order,

$$\nu^2(3,3)_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,3)_2^m \nu(3,3)_2^{-m}, \tag{C.23}$$

$$\nu^2(3,1)_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,1)_2^m \nu(3,1)_2^{-m}, \tag{C.24}$$

$$\nu(3,3)_2\nu(3,1)_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,3)_2^m \nu(3,1)_2^{-m}, \tag{C.25}$$

$$\nu(3,1)_2\nu(1,1)_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,1)_2^m \nu(1,1)_2^{-m}, \tag{C.26}$$

$$\nu(3,3)_2\nu_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,3)_2^m \nu_2^{-m}, \tag{C.27}$$

$$\nu(3,1)_2\nu_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(3,1)_2^m \nu_2^{-m}, \tag{C.28}$$

$$\nu(1,1)_2\nu_2 = 5^{-\frac{1}{2}} \sum_{m=-2}^{2} (-1)^{2-m} \nu(1,1)_2^m \nu_2^{-m}. \tag{C.29}$$

These expressions use

$$\nu(3,3)_2^2 = 2\sqrt{\frac{5}{42}}\nu_3^3\nu_3^{-1} - 2\sqrt{\frac{5}{21}}\nu_3^2\nu_3^0 + \sqrt{\frac{2}{7}}\left(\nu_3^1\right)^2, \tag{C.30}$$

$$\nu(3,3)_2^1 = \frac{5}{\sqrt{21}}\nu_3^3\nu_3^{-2} - \sqrt{\frac{5}{7}}\nu_3^2\nu_3^{-1} + 2\sqrt{\frac{1}{42}}\nu_3^1\nu_3^0, \tag{C.31}$$

$$\nu(3,3)_2^0 = \frac{5}{\sqrt{21}}\nu_3^3\nu_3^{-3} - \sqrt{\frac{3}{7}}\nu_3^1\nu_3^{-1} + 2\frac{2}{\sqrt{21}}\left(\nu_0^0\right)^2, \tag{C.32}$$

$$\nu(3,3)_2^{-1} = \frac{5}{\sqrt{21}}\nu_3^{-3}\nu_3^2 - \sqrt{\frac{5}{7}}\nu_3^{-2}\nu_3^1 + 2\sqrt{\frac{1}{42}}\nu_3^{-1}\nu_3^0, \tag{C.33}$$

$$\nu(3,3)_2^{-2} = 2\sqrt{\frac{5}{42}}\nu_3^{-3}\nu_3^1 - 2\sqrt{\frac{5}{21}}\nu_3^{-2}\nu_3^0 + \sqrt{\frac{2}{7}}\left(\nu_3^{-1}\right)^2, \tag{C.34}$$

$$\nu(3,1)_2^2 = \frac{-1}{\sqrt{105}}\nu_3^2\nu_1^0 + \frac{1}{\sqrt{35}}\nu_3^3\nu_1^{-1} + \frac{1}{5\sqrt{21}}\nu_3^1\nu_1^1, \tag{C.35}$$

$$\nu(3,1)_2^1 = \sqrt{\frac{2}{105}}\nu_3^2\nu_1^{-1} + \frac{1}{5\sqrt{7}}\nu_3^0\nu_1^1 - \frac{2}{5}\sqrt{\frac{2}{21}}\nu_3^1\nu_1^0, \tag{C.36}$$

$$\nu(3,1)_2^0 = -\frac{1}{5}\sqrt{\frac{3}{7}}\nu_3^0\nu_1^0 + \frac{1}{5}\sqrt{\frac{2}{7}}\nu_3^1\nu_1^{-1} + \frac{1}{5}\sqrt{\frac{2}{7}}\nu_3^{-1}\nu_1^1, \tag{C.37}$$

$$\nu(3,1)_2^{-1} = \sqrt{\frac{2}{105}}\nu_3^{-2}\nu_1^1 + \frac{1}{5\sqrt{7}}\nu_3^0\nu_1^{-1} - \frac{2}{5}\sqrt{\frac{2}{21}}\nu_3^{-1}\nu_1^0, \tag{C.38}$$

$$\nu(3,1)_2^{-2} = \frac{-1}{\sqrt{105}}\nu_3^{-2}\nu_1^0 + \frac{1}{\sqrt{35}}\nu_3^{-3}\nu_1^1 + \frac{1}{5\sqrt{21}}\nu_3^{-1}\nu_1^{-1}, \tag{C.39}$$

$$\nu(1,1)_2^2 = \frac{1}{5}\left(\nu_1^1\right)^2, \tag{C.40}$$

$$\nu(1,1)_2^1 = \frac{\sqrt{2}}{5}\nu_1^0\nu_1^1, \tag{C.41}$$

$$\nu(1,1)_2^0 = \frac{1}{5}\sqrt{\frac{2}{3}}\left(\left(\nu_0^1\right)^2 + \nu_1^1\nu_1^{-1}\right), \tag{C.42}$$

$$\nu(1,1)_2^{-1} = \frac{\sqrt{2}}{5}\nu_1^0\nu_1^{-1}, \tag{C.43}$$

$$\nu(1,1)_2^{-2} = \frac{1}{5}\left(\nu_1^{-1}\right)^2. \tag{C.44}$$

The quadratic moment invariants are

$$\nu(3,3)_0^0 = 7^{-\frac{1}{2}}\left[2\nu_3^3\nu_3^{-3} - 2\nu_3^2\nu_3^{-2} + 2\nu_3^1\nu_3^{-1} - \left(\nu_3^0\right)^2\right], \tag{C.45}$$

$$\nu(1,1)_0^0 = 3^{-\frac{1}{2}}\left[2\nu_1^1\nu_1^{-1} - \left(\nu_1^0\right)^2\right], \tag{C.46}$$

which apply the third-order complex moments

$$\nu_3^3 = \sqrt{\frac{\pi}{35}}\left[(-M_{300} + 3M_{120}) + i(M_{030} - 3M_{210})\right], \tag{C.47}$$

$$\nu_3^2 = \sqrt{\frac{6\pi}{35}}\left[(M_{201} - M_{021}) + i2M_{111}\right], \tag{C.48}$$

$$\nu_3^1 = \sqrt{\frac{3\pi}{5\sqrt{7}}}\left[(M_{300} + M_{120} - 4M_{102}) + i(M_{030} + M_{210} - 4M_{012})\right], \tag{C.49}$$

$$\nu_3^0 = \sqrt{\frac{2\pi}{5\sqrt{7}}}\left[2M_{003} - 3M_{201} - 3M_{021}\right], \tag{C.50}$$

$$\nu_3^{-1} = \sqrt{\frac{3\pi}{5\sqrt{7}}}\left[(-M_{300} - M_{120} + 4M_{102}) + i(M_{030} + M_{210} - 4M_{012})\right], \tag{C.51}$$

$$\nu_3^{-2} = \sqrt{\frac{6\pi}{35}}\left[(M_{201} - M_{021}) - i2M_{111}\right], \tag{C.52}$$

$$\nu_3^{-3} = \sqrt{\frac{\pi}{35}}\left[(M_{300} - 3M_{120}) + i(M_{030} - 3M_{210})\right], \tag{C.53}$$

and

$$\nu_1^1 = \frac{\sqrt{6\pi}}{5}\left[(-M_{300} - M_{120} - M_{102}) - i(M_{030} + M_{210} + M_{012})\right], \tag{C.54}$$

$$\nu_1^0 = \frac{2\sqrt{3\pi}}{5}\left[M_{003} + M_{201} + M_{021}\right], \tag{C.55}$$

$$\nu_1^{-1} = \frac{\sqrt{6\pi}}{5}\left[(M_{300} + M_{120} + M_{102}) - i(M_{030} + M_{210} + M_{012})\right]. \tag{C.56}$$

Furthermore, second-order moments find application in the form

$$\nu_0^0 = \frac{2\sqrt{\pi}}{3}J_1, \tag{C.57}$$

$$\nu(2,2)_0^0 = \frac{2\pi}{15\sqrt{5}}\left(\frac{8}{3}J_1^2 - 8J_2\right), \tag{C.58}$$

$$\eta\nu = \frac{2\pi\sqrt{2\pi}}{15\sqrt{35}}\left(\frac{-32}{9}J_1^3 + 16J_1J_2 - 48J_3\right) \tag{C.59}$$

with

$$J_1 = M_{200} + M_{020} + M_{002}, \tag{C.60}$$

$$J_2 = M_{200}M_{020} + M_{200}M_{002} + M_{020}M_{002} - M_{101}^2 - M_{110}^2 - M_{011}^2, \tag{C.61}$$

$$J_3 = M_{200}M_{020}M_{002} - M_{002}M_{110}^2 + 2M_{110}M_{101}M_{011} - M_{020}M_{101}^2 - M_{200}M_{011}^2. \tag{C.62}$$

Bibliography

[Agrawal and Davis, 2003] Agrawal, M. and Davis, L. (2003). Camera calibration using spheres: A dual-space approach. *Research Report CAR-TR-984, Center for Automation Research, University of Maryland*.

[Aloimonos et al., 1988] Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1988). Active Vision. *International Journal of Computer Vision (IJCV)*, 1:333–356.

[Armstrong et al., 1996] Armstrong, M., Zisserman, A., and Hartley, R. (1996). Self-calibration from image triplets. *Proceedings of the IEEE European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science (LNCS), Springer*, 1064(5):3–16.

[Audette et al., 2000] Audette, M., Ferrie, F., and Peters, T. (2000). An Algorithmic Overview of Surface Registration Techniques for Medical Imaging. *Medical Image Analysis*, 4(3):201–217.

[Bajcsy, 1988] Bajcsy, R. (1988). Active Perception. *Proceedings of the IEEE, Special Issue on Computer Vision*, 76(8):996–1005.

[Baker and Matthews, 2004] Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision (IJCV)*, 56:221–255.

[Banta et al., 2000] Banta, J., Wong, L., Dumont, C., and Abidi, M. (2000). A Next-Best-View System for Autonomous 3-D Object Reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, 30(5):589–597.

[Basri, 1996] Basri, R. (1996). Paraperspective $\equiv$ Affine. *International Journal of Computer Vision (IJCV)*, 19(2):169–179.

[Besl and McKay, 1992] Besl, P. and McKay, N. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256.

[Bose, 2003] Bose, T. (2003). *Digital Signal and Image Processing*. John Wiley & Sons, Inc.

[Brown, 1971] Brown, D. (1971). Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866.

[Brückner et al., 2008] Brückner, M., Bajramovic, F., and Denzler, J. (2008). Experimental Evaluation of Relative Pose Estimation Algorithms. *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2:431–438.

[Brückner and Denzler, 2010] Brückner, M. and Denzler, J. (2010). Active Self-calibration of Multi-camera Systems. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft fuer Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 6376:31–40.

[Chaumette et al., 1996] Chaumette, F., Boukir, S., Bouthemy, P., and Juvin, D. (1996). Structure From Controlled Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(5):492–504.

[Chen et al., 1999] Chen, C.-S., Hung, Y.-P., and Cheng, J.-B. (1999). RANSAC-based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(11):1229–1234.

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

[Chen and Li, 2005] Chen, S. Y. and Li, Y. F. (2005). Vision Sensor Planning for 3D Model Acquisition. *IEEE Transactions on Systems, Man, and Cybernetics – B (SMC)*, 35(5):894–904.

[Chen and Davis, 2008] Chen, X. and Davis, J. (2008). An Occlusion Metric for Selecting Robust Camera Configurations. *Machine Vision and Applications (MVA)*, 19(4):217–222.

[Cipolla et al., 1999] Cipolla, R., Drummond, T., and Robertson, D. (1999). Camera calibration from vanishing points in images of architectural scenes. *Proceedings of the British Machine Vision Conference (BMVC)*, 1:382–391.

[Delaunay, 1934] Delaunay, B. (1934). Sur la sphere vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society*, 39:1–38.

[Denzler, 2003] Denzler, J. (2003). *Probabilistische Zustandsschätzung und Aktionsauswahl im Rechnersehen*. Logos Verlag Berlin.

[Dey and Goswami, 2003] Dey, T. and Goswami, S. (2003). Tight Cocone: A water-tight surface reconstructor. *Proceedings of the ACM Symposium on Solid Modeling and Applications*, pages 127–134.

[Dubuisson and Jain, 1994] Dubuisson, M.-P. and Jain, A. (1994). A modified Hausdorff distance for object matching. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR)*, 1:566–568.

[Dunn et al., 2006] Dunn, E., Olague, G., and Lutton, E. (2006). Parisian camera placement for vision metrology. *Pattern Recognition Letters*, 27:1209–1219.

[Faugeras and Luong, 2001] Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. The MIT Press, Cambridge, London.

[Fischler and Bolles, 1981] Fischler, M. and Bolles, R. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

[Fisz, 1980] Fisz, M. (1980). *Wahrscheinlichkeitsrechnung und mathematische Statistik*. VEB Deutscher Verlag der Wissenschaften.

[Flusser, 2000] Flusser, J. (2000). On the independence of rotation moment invariants. *Pattern Recognition (PR)*, 33:1405–1410.

[Flusser, 2006] Flusser, J. (2006). Moment Invariants in Image Analysis. *Proceedings of the World Academy of Science, Engineering and Technology*, 11:196–201.

[Fraser, 1982] Fraser, C. (1982). Optimization of Precision in Close-Range Photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 53(5):487–493.

[Fröhlich et al., 2010] Fröhlich, B., Rodner, E., and Denzler, J. (2010). A Fast Approach for Pixelwise Labeling of Facade Images. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR 2010)*, 7.

[Furukawa and Ponce, 2009] Furukawa, Y. and Ponce, J. (2009). Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment. *International Journal of Computer Vision (IJCV)*, 84(3):257–268.

[Fusiello et al., 1999] Fusiello, A., Trucco, E., Tommasini, T., and Roberto, V. (1999). Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320.

[Fusiello et al., 1998] Fusiello, A., Trucco, E., and Verri, A. (1998). Rectification with Unconstrained Stereo Geometry. *Proceedings of the British Machine Vision Conference (BMVC)*, 1:400–409.

[Gantmacher, 1998] Gantmacher, F. (1998). *The Theory of Matrices*. AMS Chelsea Publishing.

[Garland and Heckbert, 1997] Garland, M. and Heckbert, S. (1997). Surface simplification using quadric error metrics. *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1:209–216.

[Girod et al., 2000] Girod, B., Greiner, G., and Niemann, H. (2000). *Principles of 3D Image Analysis and Synthesis*. Kluwer Academic Publishers, Boston, Dordrecht, London.

[Han and Kanade, 1999] Han, M. and Kanade, T. (1999). Perspective Factorization Methods for Euclidean Reconstruction. *Technical Report CMU-RI-TR-99-22, The Robotics Institute, Carnegie Mellon University*.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in computer vision, Second Edition*. Cambridge University Press.

[He et al., 2010] He, B., Long, Z., and Li, Y. (2010). The research of an automatic object reconstruction method based on limit visible region of the laser-scanning vision system. *Robotics and Computer-Integrated Manufacturing*.

[Heigl, 2003] Heigl, B. (2003). *Plenoptic Scene Modelling from Uncalibrated Image Sequences*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg.

[Horaud et al., 1994] Horaud, R., Christy, S., and Dornaika, F. (1994). Object Pose: The Link between Weak Perspective, Para Perspective, and Full Perspective. *International Journal of Computer Vision*, 22:173–189.

[Hu, 1962] Hu, M. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187.

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

[Hwang and Ahuja, 1992] Hwang, Y. and Ahuja, N. (1992). Gross Motion Planning — A Survey. *ACM Computing Surveys*, 24(3):219–291.

[Jeffreys and Jeffreys, 1988] Jeffreys, H. and Jeffreys, B. (1988). *Methods of Mathematical Physics, 3rd ed.* Cambridge University Press, Cambridge, England.

[Kähler, 2009] Kähler, O. (2009). Model-based Online 3D Reconstruction from Image Sequences. *PhD Thesis, Friedrich-Schiller University of Jena*.

[Kähler et al., 2008] Kähler, O., Rodner, E., and Denzler, J. (2008). On Fusion of Range and Intensity Information Using Graph-Cut for Planar Patch Segmentation. *International Journal of Intelligent Systems, Technologies and Applications*, 5(3/4):365–373.

[Kuehmstedt et al., 2001] Kuehmstedt, P., Notni, G., Hintersehr, J., and Gerber, J. (2001). Cad-cam-system for dental purpose – an industrial application. In *The 4th International Workshop on Automatic Processing of Fringe Patterns*.

[Kühmstedt et al., 2007] Kühmstedt, P., Munkelt, C., Braeuer-Burchardt, C., and Notni, G. (2007). 3D shape measurement with phase correlation based fringe projection. In Osten, W., Gorecki, C., and Novak, E. L., editors, *Optical Measurement Systems for Industrial Inspection V*, volume 6616, page 66160B. SPIE.

[Kuhn, 1955] Kuhn, H. (1955). The Hungarian Method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.

[Kutulakos and Seitz, 2000] Kutulakos, K. and Seitz, S. (2000). A Theory of Shape by Space Carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218.

[Li et al., 2005] Li, Y., He, B., Chen, S., and Bao, P. (2005). A view planning method incorporating self-termination for automated surface measurement. *Measurement Science and Technology*, 16:1865–1877.

[Lloyd, 1982] Lloyd, S. (1982). Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

[Lo and Don, 1989] Lo, C. and Don, H. (1989). 3-D Moment Forms: Their Construction and Application to Object Identification and Positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(10):1053–1064.

[Longuet-Higgins, 1981] Longuet-Higgins, H. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

[Loriot et al., 2008] Loriot, B., Seulin, R., and Gorria, P. (2008). Non-Model Based Method for an Automation of 3D Acquisition and Post-Processing. *Electronic Letters on Computer Vision and Image Analysis*, 7(3):67–82.

[Low and Lastra, 2006] Low, K.-L. and Lastra, A. (2006). Efficient Constraint Evaluation Algorithms for Hierarchical Next-Best-View Planning. *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 1:830–837.

[Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679.

[Luo and Hancock, 2003] Luo, B. and Hancock, E. (2003). A Unified Framework for Alignment and Correspondence. *Computer Vision and Image Understanding (CVIU)*, 92:26–55.

[Madsen and Christensen, 1997] Madsen, C. and Christensen, H. (1997). A Viewpoint Planning Strategy for Determining True Angles on Polyhedral Objects by Camera Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(2):158–163.

[Makadia et al., 2006] Makadia, A., Patterson, E., and Daniilidis, K. (2006). Fully automatic registration of 3d point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:1297–1304.

[Marchand and Chaumette, 1999] Marchand, E. and Chaumette, F. (1999). Active Vision for Complete Scene Reconstruction and Exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(1):65–72.

[Maronna et al., 2006] Maronna, R., Martin, R., and Yohai, V. (2006). *Robust Statistics*. Wiley Series in Probability and Statistics.

[Massios and Fisher, 1998] Massios, N. A. and Fisher, R. B. (1998). A Best Next View selection algorithm incorporating a quality criterion. *Proceedings of the British Machine Vision Conference (BMVC)*, 1:780–789.

[Maver and Bajcsy, 1993] Maver, J. and Bajcsy, R. (1993). Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(5):417–433.

[Mian et al., 2006] Mian, A. S., Bennamoun, M., and Owens, R. (2006). A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images. *International Journal of Computer Vision (IJCV)*, 66(1):19–40.

[Munkelt et al., 2010] Munkelt, C., Breitbarth, A., Notni, G., and Denzler, J. (2010). Multi-View Planning for Simultaneous Coverage and Accuracy Optimisation. *Proceedings of the British Machine Vision Conference (BMVC)*, pages 118.1–118.11.

[Munkelt et al., 2006] Munkelt, C., Kuehmstedt, P., and Denzler, J. (2006). Incorporation of a-priori information in planning the next best view. In Kobbelt, L., Kuhlen, T., Aach, T., and Westermann, R., editors, *Proceedings of Vision, Modeling, and Visualization (VMV)*, pages 261–268. Aka GmbH, Berlin.

[Munkelt et al., 2009] Munkelt, C., Trummer, M., Kühmstedt, P., Notni, G., and Denzler, J. (2009). View Planning for 3D Reconstruction Using Time-of-Flight Camera Data. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 5748:352–361.

[Munkelt et al., 2007] Munkelt, C., Trummer, M., Wenhardt, S., and Denzler, J. (2007). Benchmarking 3D Reconstructions from Next Best View Planning. *Proceedings of the IAPR Conference on Machine Vision Applications (MVA)*, 1:552–555.

[Olague and Mohr, 2002] Olague, G. and Mohr, R. (2002). Optimal camera placement for accurate reconstruction. *Pattern Recognition (PR)*, 35:927–944.

[Paquet et al., 2000] Paquet, E., Rioux, M., Murching, A., Naveen, T., and Tabatabai, A. (2000). Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16(1):103–122.

[Pito, 1999] Pito, R. (1999). A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(10):1016–1030.

[Poelman and Kanade, 1997] Poelman, C. and Kanade, T. (1997). A Paraperspective Factorization method for Shape and Motion Recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(3):206–218.

[Pukelsheim, 1993] Pukelsheim, F. (1993). *Optimal Design of Experiments*. John Whiley & Sons Inc., New York, Chichester, Brisbane, Toronto, Singapur.

[Rav-Acha and Peleg, 2006] Rav-Acha, A. and Peleg, S. (2006). Lucas-kanade without iterative warping. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*.

[Reed and Allen, 2000] Reed, M. and Allen, P. (2000). Constraint-Based Sensor Planning for Scene Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1460–1467.

[Roberts and Marshall, 1998] Roberts, D. R. and Marshall, A. D. (1998). Viewpoint Selection for Complete Surface Coverage of Three Dimensional Objects. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 740–750, Department of Computer Science, Cardiff University, PO Box 916, Cardiff. CF2 3XF, U.K.

[Rozenwald et al., 2010] Rozenwald, G., Seulin, R., and Fougerolle, Y. (2010). Fully automatic 3D digitization of unknown objects. *Proceedings of SPIE*, 7538:753807–753815.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient Variants of the ICP Algorithm. *Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 1:145–152.

[Sablatnig et al., 2003] Sablatnig, R., Tosovic, S., and Kampel, M. (2003). Next View Planning for a Combination of Passive and Active Acquisition Techniques. *Proceedings of the IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 1:62–69.

[Salman and Yvinec, 2010] Salman, N. and Yvinec, M. (2010). Surface Reconstruction from Multi-View Stereo of Large-Scale Outdoor Scenes. *The International Journal of Virtual Reality (IJVR)*, 9(1):19–26.

[Schoen and Haeusler, 2006] Schoen, N. and Haeusler, G. (2006). Automatic Coarse Registration of 3D-Surfaces by Information Theoretic Selection of Salient Points. *Applied Optics*, 45:6539–6550.

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

[Scott, 2009] Scott, W. R. (2009). Model-based View Planning. *Machine Vision and Applications (MVA)*, 20:47–69.

[Seitz et al., 2006] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo

reconstruction algorithms. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1 of *2006 IEEE Computer Society Conference on*, pages 519–528.

[Sharp et al., 2002] Sharp, G., Lee, S., and Wehe, D. (2002). ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(1):90–102.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600.

[Shih and Gerhardt, 2006] Shih, C. and Gerhardt, L. (2006). Integration of view planning with nonuniform surface sampling techniques for three-dimensional object inspection. *Optical Engineering*, 45(11):113602–113618.

[Shmuel and Werman, 1990] Shmuel, A. and Werman, M. (1990). Active Vision: 3D From An Image Sequence. *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 1:48–54.

[Stewenius et al., 2006] Stewenius, H., Engels, C., and Nister, D. (2006). Recent Developments on Direct Relative Orientation. *Journal of Photogrammetry and Remote Sensing (ISPRS)*, 60(4):284–294.

[Sturm and Triggs, 1996] Sturm, P. and Triggs, B. (1996). A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 709–720.

[Suikerbuik et al., 2004] Suikerbuik, R., Tangelder, H., Daanen, H., and Oudenhuijzen, A. (2004). Automatic feature detection in 3D human body scans. *SAE transactions*, 113(1):260–263.

[Süße, 20xx] Süße, H. (20xx). *Signalbasierte Methoden der Bildverarbeitung*. to appear.

[Süße and Ortmann, 2003] Süße, H. and Ortmann, W. (2003). Robust Matching of Affinely Transformed Objects. *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2:375–378.

[Sutton and Barto, 1998] Sutton, R. and Barto, A. (1998). *Reinforcement Learning*. Cambridge, London.

[Tarbox and Gottschlich, 1995] Tarbox, G. and Gottschlich, S. (1995). Planning for Complete Sensor Coverage in Inspection. *Computer Vision nad Image Understanding (CVIU)*, 61(1):84–111.

[Tomasi and Kanade, 1992] Tomasi, C. and Kanade, T. (1992). Shape and Motion from Image Streams under Orthography: a Factorization Method. *International Journal of Computer Vision (IJCV)*, 9(2):137–154.

[Trucco and Verri, 1998] Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice Hall.

[Trummer et al., 2008] Trummer, M., Denzler, J., and Munkelt, C. (2008). KLT Tracking Using Intrinsic and Extrinsic Camera Parameters in Consideration of Uncertainty. *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2:346–351.

[Trummer et al., 2009a] Trummer, M., Denzler, J., and Munkelt, C. (2009a). Extending GKLT Tracking – Feature Tracking for Controlled Environments with Integrated Uncertainty Estimation. *Proceedings of the IAPR Scandinavian Conference on Image Analysis (SCIA), Lecture Notes in Computer Science (LNCS), Springer*, 5575:460–469.

[Trummer et al., 2009b] Trummer, M., Denzler, J., and Munkelt, C. (2009b). Guided KLT Tracking Using Camera Parameters in Consideration of Uncertainty. *Lecture Notes in Communications in Computer and Information Science (CCIS), Springer*, 24:252–261.

[Trummer et al., 2006] Trummer, M., Denzler, J., and Süße, H. (2006). Precise 3D Measurement with Standard Means and Minimal User Interaction – Extended Single-view Reconstruction. In *Proceedings of the International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*.

[Trummer et al., 2009c] Trummer, M., Munkelt, C., and Denzler, J. (2009c). Combined GKLT Feature Tracking and Reconstruction for Next Best View Planning. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft fuer Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 5748:161–170.

[Trummer et al., 2010a] Trummer, M., Munkelt, C., and Denzler, J. (2010a). Online Next-Best-View Planning for Accuracy Optimization Using an Extended E-Criterion. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR)*.

[Trummer and Süße, 2006] Trummer, M. and Süße, H. (2006). Verfahren zur hochgenauen dreidimensionalen Vermessung und/oder Rekonstruktion von Objekten mit Hilfe digitaler Bildaufnahmen, beispielsweise zur Bildauswertung von Verkehrsstrecken. *German Patent*, DE 10 2005 043 070.8.

[Trummer et al., 2009d] Trummer, M., Süße, H., and Denzler, J. (2009d). Coarse Registration of 3D Surface Triangulations Based on Moment Invariants with Applications to Object Alignment and Identification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1:1273–1279.

[Trummer et al., 2010b] Trummer, M., Süße, H., and Denzler, J. (2010b). Verfahren zum Vergleich von in beliebigen Koordinatensystemen und in beliebigen Skalierungen vorliegenden 3D-Oberflächentriangulationen von Objekten. *German Patent*, DE 10 2009 022 600.1.

[Trummer et al., 2010c] Trummer, M., Süße, H., and Denzler, J. (2010c). Verfahren zur Registrierung zweier in unterschiedlichen Koordinatensystemen vorliegenden 3D-Oberflächentriangulationen eines Objektes unter Ähnlichkeitstransformation. *German Patent*, DE 10 2009 022 599.4.

[Trummer et al., 2005] Trummer, M., Süße, H., Denzler, J., and Ditrich, F. (2005). Metrische 3-D-Vermessung einer Straßenszene mit Spezialwissen unter minimaler Interaktion – Erweiterte monokulare Rekonstruktion. *Publikationen der Deutschen Geselllschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.*, 14:39–46.

[Voß et al., 2001] Voß, K., Süße, H., and Bräuer-Burchardt, C. (2001). Affine Point Pattern Matching. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 2191:155–162.

[Walker et al., 1991] Walker, M. W., Shao, L., and Volz, R. A. (1991). Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367.

[Wang and Gupta, 2007] Wang, P. and Gupta, K. (2007). View planning for exploration via maximal C-space entropy reduction for robot mounted range sensors. *Advanced Robotics*, 21(7):771–792.

[Wenhardt et al., 2007] Wenhardt, S., Deutsch, B., Angelopoulou, E., and Niemann, H. (2007). Active Visual Object Reconstruction using D-, E-, and T-Optimal Next Best Views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Wenhardt et al., 2006] Wenhardt, S., Deutsch, B., Hornegger, J., Niemann, H., and Denzler, J. (2006). An Information Theoretic Approach for Next Best View Planning in 3-D Reconstruction. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR)*, 1:103–106.

[Whaite and Ferrie, 1997] Whaite, P. and Ferrie, F. (1997). Autonomous Exploration: Driven by Uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(3):193–205.

[Winkelbach et al., 2004] Winkelbach, S., Rilk, M., Schoenfelder, C., and Wahl, F. (2004). Fast Random Sample Matching of 3D Fragments. *Lecture Notes in Computer Science (LNCS), Springer*, 3175:129–136.

[Wyngaerd and van Gool, 2002] Wyngaerd, J. and van Gool, L. (2002). Automatic Crude Patch Registration: Toward Automatic 3D Model Building. *Computer Vision and Image Understanding (CVIU)*, 87(1):8–26.

[Xiao et al., 2007] Xiao, G., Ong, S., and Foong, K. (2007). 3D registration of partial overlapping surfaces using a volumetric approach. *Image and Vision Computing*, 25(6):934–944.

[Xu and Li, 2006] Xu, D. and Li, H. (2006). 3-d surface moment invariants. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR)*, 4:173–176.

[Zheng et al., 1991] Zheng, J., Chen, Q., and Tsuji, S. (1991). Active Camera Guided Manipulation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1:632–638.

[Zinßer et al., 2005] Zinßer, T., Gräßl, C., and Niemann, H. (2005). High-speed feature point tracking. In *Proceedings of the Conference on Vision, Modeling, and Visualization (VMV)*.

[Zinßer et al., 2003] Zinßer, T., Schmidt, J., and Niemann, H. (2003). A refined ICP algorithm for robust 3-D correspondence estimation. *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2:695–698.

List of Own Publications

Trummer, M., Süße, H., Denzler, J., and Ditrich, F. (2005). Metrische 3-D-Vermessung einer Straßenszene mit Spezialwissen unter minimaler Interaktion – Erweiterte monokulare Rekonstruktion. *Publikationen der Deutschen Geselllschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.*, 14:39–46

Trummer, M., Denzler, J., and Süße, H. (2006). Precise 3D Measurement with Standard Means and Minimal User Interaction – Extended Single-view Reconstruction. In *Proceedings of the International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*

Munkelt, C., Trummer, M., Wenhardt, S., and Denzler, J. (2007). Benchmarking 3D Reconstructions from Next Best View Planning. *Proceedings of the IAPR Conference on Machine Vision Applications (MVA)*, 1:552–555

Trummer, M., Denzler, J., and Munkelt, C. (2008). KLT Tracking Using Intrinsic and Extrinsic Camera Parameters in Consideration of Uncertainty. *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2:346–351

Trummer, M., Süße, H., and Denzler, J. (2009d). Coarse Registration of 3D Surface Triangulations Based on Moment Invariants with Applications to Object Alignment and Identification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1:1273–1279

Munkelt, C., Trummer, M., Kühmstedt, P., Notni, G., and Denzler, J. (2009). View Planning for 3D Reconstruction Using Time-of-Flight Camera Data. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 5748:352–361

Trummer, M., Denzler, J., and Munkelt, C. (2009b). Guided KLT Tracking Using Camera Parameters in Consideration of Uncertainty. *Lecture Notes in Communications in Computer and Information Science (CCIS), Springer*, 24:252–261

Trummer, M., Denzler, J., and Munkelt, C. (2009a). Extending GKLT Tracking – Feature Tracking for Controlled Environments with Integrated Uncertainty Estimation. *Proceedings of the IAPR Scandinavian Conference on Image Analysis (SCIA), Lecture Notes in Computer Science (LNCS), Springer*, 5575:460–469

Trummer, M., Munkelt, C., and Denzler, J. (2009c). Combined GKLT Feature Tracking and Reconstruction for Next Best View Planning. *Proceedings of the Symposium of Deutsche Arbeitsgemeinschaft fuer Mustererkennung (DAGM), Lecture Notes in Computer Science (LNCS), Springer*, 5748:161–170

Trummer, M., Munkelt, C., and Denzler, J. (2010a). Online Next-Best-View Planning for Accuracy Optimization Using an Extended E-Criterion. *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR)*

Trummer, M. and Süße, H. (2006). Verfahren zur hochgenauen dreidimensionalen Vermessung und/oder Rekonstruktion von Objekten mit Hilfe digitaler Bildaufnahmen, beispielsweise zur Bildauswertung von Verkehrsstrecken. *German Patent*, DE 10 2005 043 070.8

Trummer, M., Süße, H., and Denzler, J. (2010b). Verfahren zum Vergleich von in beliebigen Koordinatensystemen und in beliebigen Skalierungen vorliegenden 3D-Oberflächentriangulationen von Objekten. *German Patent*, DE 10 2009 022 600.1

Trummer, M., Süße, H., and Denzler, J. (2010c). Verfahren zur Registrierung zweier in unterschiedlichen Koordinatensystemen vorliegenden 3D-Oberflächentriangulationen eines Objektes unter Ähnlichkeitstransformation. *German Patent*, DE 10 2009 022 599.4

Index