Technische Universität München, Institut für Informatik
Chair for Computer-Aided Medical Procedures & Augmented Reality

Tracking Errors in Augmented Reality

Martin A. Bauer

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Hans Michael Gerndt
Prüfer der Dissertation:	1. Univ.-Prof. Gudrun J. Klinker, Ph.D.
	2. Univ.-Prof. Nassir Navab, Ph.D.

Die Dissertation wurde am 23.04.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 10.09.2007 angenommen.

Abstract

Tracking is an important component of augmented reality systems. Tracking means to determine and follow the position and orientation of an object with respect to some reference coordinate system over time. As with all physical measurements the estimate is affected by errors. These errors propagate through the chain of transformations from the tracking systems to the final application,until they appear as alignment errors in the augmented reality application.

This thesis deals with the mathematical tools that are needed to estimate the final resulting precision, as well as with visualization concepts for exploring and understanding tracking errors. The special case of optical tracking systems is further analyzed to provide useful initial estimates of the tracking error. A set of examples finally shows the application of the error estimation in real-world setups.

Zusammenfassung

Das *Tracking* ist ein wichtiger Bestandteil von Anwendungen im Bereich der Erweiterten Realität. *Tracking* bedeutet, die Position und Ausrichtung eines Objektes relativ zu einem Referenzkoordinatensystem über einen gewissen Zeitraum hin zu bestimmen und zu verfolgen. Wie jeder andere Messprozess ist auch dieser Prozess fehlerbehaftet. Diese Messfehler pflanzen sich durch die gesamte Kette der Transformationen von der Messung bis hin zur tatsächlichen Anwendung fort, wo sie dann als Fehler in der Überlagerung sichtbar werden.

Die vorliegende Arbeit befasst sich sowohl mit den mathematischen Methoden, die nötig sind um die endgültige Genauigkeit abzuschätzen, als auch mit Visualisierungsmöglichkeiten um diese Fehler zu untersuchen und zu verstehen. Für den Spezialfall eines optischen Trackingsystems wird ein Verfahren zur Abschätzung des Messfehlers vorgestellt. Die Nützlichkeit der präsentierten Methoden wird anhand einer Reihe von Beispielen verdeutlicht.

Contents

Part I Introduction

Part II Modeling and Estimating Uncertain Transformations

Part III Applications

Part IV Appendix

Part I

Introduction

1 Augmented Reality

Augmented reality is a field of computer science, which deals with a special form of human-computer interaction. The main concept of augmented reality is to augment a user's perception of the reality with additional information. This additional information can be computer-generated, or simply information that humans can not perceive directly with their senses.

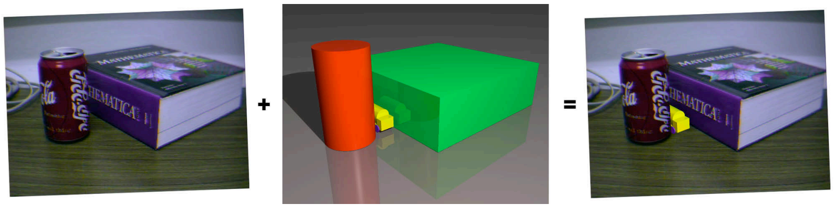


Fig. 1.1. Example scenario for augmenting the perception of the reality by computer-mediated information

An example can be seen in figure 1.1. The real scene is augmented by a virtual yellow car. To provide correct integration into the real scene, the virtual scene contains additionally a rough model of the reality.

The term *augmented reality* is formed in analogy to *virtual reality*, where the user is fully immersed into a virtual world. In contrast to that, augmented reality tries to bring the virtual information from the computer 'back to the real world' [208]. According to Milgram's definition of the *reality-virtuality continuum* [121], both augmented reality and the field of *augmented virtuality* together form the field of *mixed reality*. *Augmented virtuality* denotes here virtual environments that are enhanced by information captured from the reality, like textures or videos. While the term is actually rarely used in the literature nowadays, *augmented reality* and *mixed reality* are mostly used as synonyms. Sometimes also the term *enhanced reality* occurs [74, 159, 118] interchangeably in this context.

Initially the term referred only to the use of *head-mounted displays* [120], but soon included more senses [12]. However, currently most of the research is still only focused on augmenting the visual sense using several different presentation techniques.

1.1 History and Definition

The history of augmented reality starts – even though the term was not used at this time – with the first *head-mounted displays* [147, 29] by Sutherland [177] and Furness [67] in the late 1960s. The ability to see the reality through the display did soon inspire researchers to the idea of spatially registered visual information.

While the *virtual reality* community evolved over the years, the technical limitations were still not allowing the use of the VR hardware for a widespread use of augmented reality. Only in the early 1990s the technology was advanced enough, and soon applications started to appear [30, 15, 145].

It is unclear when the term *augmented reality* for this part of the virtual reality research was first used. Although the first research paper containing the term was published in 1992 [30], it is most probable that it was already in widespread use [145] at that time. In february 1993, a small workshop on 'Augmented Reality and Ubiquitous Computing' was held at the M.I.T., leading to a special issue on computer augmented environments [208] in the *Communications of the ACM* journal, which really coined the term in several articles.

In the commonly accepted definition by Azuma [12], an augmented reality system has the following three requirements: First, the system combines real and virtual objects in a real environment, Second, this combination needs to be interactive and in real time. Third, the combination of the real and virtual objects has to be registered in the three-dimensional world.

In a later publication, this definition was extended [10] stating that the real and virtual objects should be aligned with each other rather than being registered in three dimensions. This slight distinction in the definition actually denotes a significant difference when it comes to the area of tracking: Not only the three-dimensional registration accuracy in the measurement space is important, but what is even more of interest is the actual alignment in the application space. If the application consists o two-dimensional graphical overlay on camera images, this two-dimensional overlay has to be aligned correctly. If measuring distances in three-dimensional space is the purpose of the application, the alignment needs to be accurate in three dimensions. The alignment has to be achieved in the whole working volume of the application. In particular this includes all kinds of error propagation as presented in chapter 4.2 and 5.

Based on these definitions, every augmented reality system requires some tracking of the user, his head or more specifically his eyes, and any body parts needed for the interaction with the system. To combine the real and virtual information, some kind of display hardware is needed, and in between a software infrastructure allowing the capturing and processing of the necessary tracking and interaction information and the generation of the virtual information according to the application. The authoring or preparation of three-dimensional content is also an important issue, sometimes even solved again using an augmented reality system.

1.2 Tracking Requirements for Augmented Reality

To be able to align virtual, computer mediated objects with reality, we need to know the pose of the object relative to the user's pose, or depending on the used display technology also relative to the display's pose. To obtain the illusion of coexistence of virtual and real, this pose has to be as accurate as possible [8]. Many different tracking systems exist [119, 21, 28, 148] that can give us this pose. While none of them is perfect [207], we can still get useful results by carefully designing the tracking system and sometimes using a hybrid combination of different tracking systems [200].

The term 'tracking' in general means to observe the measured state of some object over time. It is used with this meaning in a wide range of application fields including engineering, finance and logistics. In the context of computer vision, tracking could be the following of the pose of an object from one frame to the next. The tracked state here would contain for example the position of the object on the frame together with other features like the color or the momentum. In the virtual reality community however, the term tracking is used in a broader way: A tracking system is a system that returns the pose of an object with a certain frequency, no matter if actually a tracking algorithm in the computer vision sense has been used for the pose reconstruction.

In the context of this thesis we follow the definition from virutal reality and denote with *tracking* the process of estimating and following the pose of an object over time. For the accuracy considerations in the following chapters however, we only consider *on-measurement* accuracy – exactly at the time the measurement was made. Continuous accuracy estimates in between measurements could be achieved by using an additional motion model [3].

The requirements on the accuracy of a tracking system depend on the application. Usually these requirements are not imposed on the measuring process itself but rather on the propagated error and the application space [113]. We take now a closer look at the main application areas for augmented reality with respect to their specific accuracy requirements.

1.2.1 Medical Applications

Medical applications have been subject to augmented reality research from the beginning. One reason for this might be the already existing highly instrumented environment in modern operating rooms. Another driving force is the need for novel user interfaces to deal with the increasingly large amounts of multidimensional medical data. Three-dimensional data sets are regularly generated for diagnosis and intraoperative navigation. In intraoperative navigation, tracked instruments are displayed within three-dimensional imaging data to provide visual guidance for the surgeon. The main difference to augmented reality is the lack of *in-situ* visualization. The navigation information is displayed potentially together with preoperative planning data on a separate screen referenced to the imaging data. For the extension to augmented reality, the planning data and additional information is displayed *in-situ* registered with the actual patient. This can be performed using head-mounted displays [26], or by augmenting already existing optical instruments like endoscopes [159], microscopes [51], or x-ray machines [125].

The final alignment accuracy between a virtual object and the real patient now depends not only on the accuracy with which the patient or instruments can be tracked, but rather on the accuracy of the relative pose between the display and the real object. This involves several error propagation steps, which need to be as small as possible. Hoff et.al. [84] propose a combination of fixed and head-mounted sensors to minimize the errors. Sauer et.al. [152] have presented a design for a combined head-mounted display and optical tracking system minimizing the propagated alignment error by using an *inside-out* configuration for the tracking. Stetten et.al. [171] have worked on an enhanced ultrasound transducer equipped with a semitransparent display. The display is mounted in a way such that the ultrasound image is aligned with the reality by design from any viewing angle, requiring no tracking at all and therefore minimizing the tracking error. Unfortunately, similar solutions are not always possible due to anatomical constraints.

In general, medical augmented reality systems are used as a tool to support surgeons in their primary task and should not require too many changes in the workflow of this task. Additionally, the procedures can in general not be repeated if the accuracy was insufficient, and they are subject to hard real-time constraints.

1.2.2 Industrial Applications

Industrial applications of augmented reality can be used across the whole product life cycle [64]. Orthogonal to the life cycle phases, the applications can be classified into three main areas: Among the first proposed applications of augmented reality were construction and repair scenarios [30, 54]. Here usually the accuracy requirements are not as strict as in medical applications since the displayed information mainly consists of abstract instructions. In a second area, augmented reality is used to present virtual objects, such as preliminary car designs, together with existing real objects in a natural environment [104, 65]. Here like in virtual reality scenarios the photorealistic effect is much more important than the absolute accuracy. Due to the interactivity of such applications, the dynamic errors resulting from fast movements have much higher impact than the static registration errors. Thirdly, augmented reality can be used for actual measurements between real and virtual objects [135, 129]. In this case case the accuracy requirements are similar to the medical scenarios. However, in most cases the augmentation can be rather static, for example by putting the camera capturing the real scene on a tripod.

The *International Standards Organisation (ISO)* has published a *Guide to the Expression of Uncertainty in Measurement* [88, 89] which states guidelines for measurements in industrial applications. These guidelines have also been implemented in their according national counterparts [191, 192].

1.2.3 Other Applications

A large range of augmented reality applications have been proposed in the field of gaming, teaching and entertainment. Typically these applications are still of mostly academic nature. Here, the researchers focus on other aspects of augmented reality (see section 1.3) than tracking accuracy, and therefore the requirements are not as strict.

In particular in the research for interaction techniques, it is more important to reduce the dynamic errors due to movements than to improve the overall accuracy, since the magnitude of the dynamic error is in such applications much larger than the static registration error [11].

1.3 Research Issues

If we take a look at the accepted research papers at the *International Symposium on Mixed and Augmented Reality*, the main conference for augmented reality over the last five years (cf. figure 1.2), we can see that the two most challenging research areas are still the tracking and calibration areas as well as the research in rendering and interaction, as stated by Bishop et.al. [27] in 1992 already and confirmed by Azuma et.al. [12, 10] and Klinker [103] in the late 1990s. However, if we take a look inside the research areas, we can see that there has been some advance and also some change in focus in all areas.

1.3.1 Tracking and Calibration

Research in the tracking and calibration area has started from simple optical marker-based tracking systems, but now focuses with increased generality more and more on markerless tracking in unprepared environments. While in

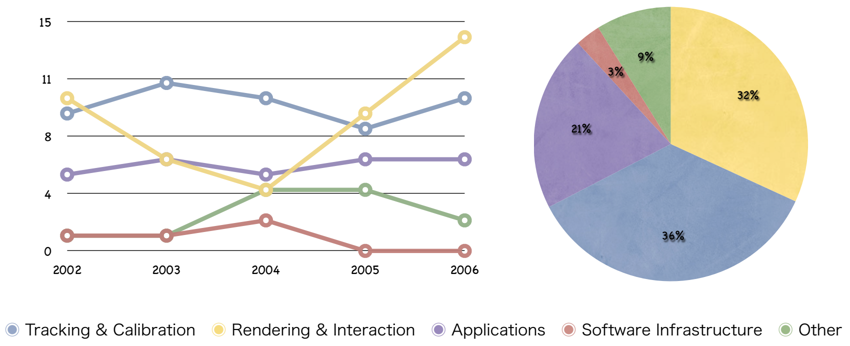


Fig. 1.2. Research Issues at the International Symposium on Mixed and Augmented Reality (ISMAR) over the years 2002–2006 by number of accepted papers.

the beginning only small movements could be tracked without large markers, now natural feature tracking becomes usable in more and more situations.

On the calibration side, progress has been made in the use of different tracking modalities in unified frameworks and the estimation of the resulting errors. With the increasing commitment to applications outside of the purely academic use, the accuracy of possibly complex tracking and calibration setups has moved again more into the focus of research.

1.3.2 Rendering and Interaction

From the user interface side, interaction with three-dimensional content as well as immersive authoring of content is still subject to intensive research. Advanced rendering techniques, as they are developed in the field of computer graphics, appear in augmented reality more often as well.

One problem for commercial applications remains here the authoring of large amounts of three-dimensional interactive content from existing data [172].

1.3.3 Applications and Other Issues

Applications of augmented reality also appear at other conferences specific to their particular field. Especially in the medical field, the *onference on Medical Image Computing and Compter-Assisted Intervention (MICCAI)* has seen an increased amount of publications that are closely related to the field of augmented reality recently. For industrial applications, there have been workshops on industrial augmented reality in the side program of ISMAR in the last two years. In Germany, the industry together with the ministry for education and research defined the research project ARVIKA [7, 64, 65], and a successor project ARTESAS [6] to investigate the potential use of augmented reality in industry. This shows the ongoing interest in the industry on augmented reality technologies. Also for academic applications we can see from figure 1.2 that there is continuous interest. Supporting frameworks for building augmented reality applications are also still an active research field.

2

Introduction to Ubiquitous Tracking

In the context of this thesis, tracking is the continuous estimation of the pose of an object relative to some reference frame. Many technologies to achieve this goal exist, but none of them is perfect [207]. To gain more flexibility in tracking, it is necessary to be able to combine different tracking technologies together without re-implementing the whole application. In principle we want to be able to get the pose of any object relative to any other object at any time. We call this concept *Ubiquitous Tracking*.

To reach this goal, we have a developed a layered tracking and calibration framework [201, 200] which can be used to model, describe and implement tracking and calibration algorithms. The *Ubiquitous Tracking* model consists in the following three layers:

A Formal Description. The first step towards reusability is to have a unified way to describe complex tracking and calibration scenarios such that common subproblems can be identified in different setups. To this end we have defined [127] the *spatial relationship graph* as a formal way to describe the spatial relationships between the objects in a scenario. We present the formal description model in section 2.1 where we also show how this description can be used by itself to discuss frequent tracking problems.

A Library of Algorithms. Once we have a description of the setup and of the problem, we need algorithms to solve the problem. We have identified [141] basic subproblems and corresponding solutions from the literature which will be described in detail in section 7. Pustka et.al. [141] call this concept *Spatial Relationship Patterns* in the style of design patterns in software engineering. In addition to that, an algorithm is presented that can be used to automatically find possible solutions by consecutively applying the patterns until the goal can be reached. This library needs to get extended and updated with new algorithms that are newly developed or that are not yet covered by the library.

Implementation Concepts. While the described concepts are not depending on a single implementation but should by definition cover all possible tracking and calibration systems, there are however implementations that suit the philosophy of reusable components better than other. We present in section 2.2 four frameworks from different research groups and discuss their respective advantages and disadvantages.

2.1 Formal Description

At the moment, developing an augmented reality tracking system requires a lot of in-depth knowledge of calibration algorithms, in particular when the system is going to use more than only one external tracking system or display. Often there are several ways to compute a coordinate system transformation for the visualization. The decision is usually done based on personal experience of the programmer and then hard-coded into the system. We have proposed [128] an abstract description of all the spatial relationships in such a system to facilitate the reuse of successful implementations in new setups.

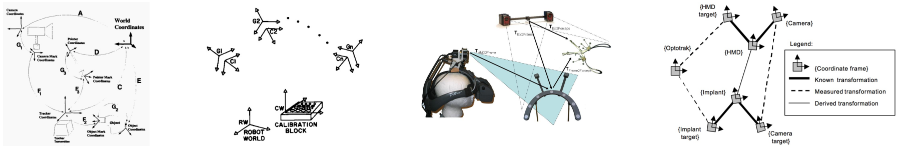
Like software engineering has moved from trial-and-error to modeling and specification, and computer graphics has moved from programming every single primitive to declaring complex scenes, we want to make the step from reimplementing calibration and tracking algorithms to the automatic generation of the required data flow and algorithms from a declaration of the setup [141].

2.1.1 Descriptive Language

For the purpose of tracking and calibration, we restrict our description of objects in the system to the description of the spatial relationship of the objects among themselves, i.e. we do not consider other properties like shape, color, or material which might be of interest when rendering the objects. Every rigid object is represented in its own coordinate system, where a single rigid object can be associated with several distinctive coordinate systems are statically connected to each other by rigid transformations. In the real world now every object has a well-defined spatial relationship with each other object. We express this relationship by the transformation in space that would be needed to align the two coordinate systems [200]. Note also that this transformation might be continuously changing over time for some of the objects.

The graph consisting of all these spatial relations is called the *spatial relationship graph*. Note that this is not a new concept; informal drawings of such spatial relationships are well known in the literature. A few examples can be seen in figure 2.1. However, we propose to use a unified syntax [141] in these drawings to make them easier to understand on first sight.

In reality however we do not know the exact transformations; all we know are measurements from tracking systems or calibration algorithms or from other models. These measurements contain errors and only exist at discrete points in time. Some transformations are modeled or measured only once and



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

Fig. 2.1. Informal drawings of Spatial Relationship Graphs are well-known from the Literature: (a) Tuceryan et.al. [188], (b) Tsai et.al. [185], (c) Sielhorst et.al. [169], (d) Hoff [84]

assumed to be static after that, because they belong to the same rigid object or their relationship does not change during the usage of the system. Some other transformations are continuously measured by some kind of tracking system, where the update rates of different tracking systems range from few milliseconds (some optical tracking systems) to seconds (GPS).

Usually, the transformation needed for visualization is not among the transformations that can be directly measured. To compute the transformation we need to infer it from our measurements. Any inferred transformation can be considered as a measurement itself; by subsequently applying inference algorithms we can compute the desired transformation [141].

Figure 2.2 shows a classification of the edges in a spatial relationship graph. The spatial relationship graph contains all spatial relationships in the system, not only the ones that were actually measured.

Static Measurements

As stated above, there are spatial relationships in the graph that do not change over time, either because the two coordinate systems belong to the same rigid object, or because their relationship can be assumed to stay constant during runtime. This might be the case for example for the base coordinate systems of two independent tracking systems that are setup in a room and the respective relationship is once calibrated and then assumed to stay constant as long as the hardware is not moved around.

The fact that an edge is static might be used later by inference algorithms as seen in section 7. Note also that even edges that are not explicitly modeled as static edges might over some period of time actually stay static when for example a tracked object is not moved during some stage of the application.

Dynamic Measurements

Dynamic measurements on the other hand are measurements of the spatial relationship of two objects that are not connected to each other and are repeated with some frequency. This applies also to measurements of a transformation

	direct	indirect
static	**modelled edges** virtual objects CAD data medical imaging data manually measured	**calibrated edges** static relationships tracker alignement
variable	**measured edges** external tracking systems sensors	**inferred edges** interpolation filter (smoothing) sensor fusion

Fig. 2.2. Classification of edges in the spatial relationship graph

where the underlying objects are not really moving relatie to each other, but due to the errors in the measurements the measured transformation is changing constantly.

Direct Measurements

Direct Measurements are measurements that are either known, modeled or come from external tracking systems, i.e. they do not depend on other measurements. These measurements form the basic *spatial relationship graph*.

The measurements are disturbed by different kinds of errors. For static edges, the error is systematic since it does not change over time. Such error can come from wrong calibration of the external measurement tools, from wrong modeling or from changes of the actual spatial relationship that was assumed to be static, like a physical movement of a sensor.

Inferred Measurements

Inferred measurements are measurements that are derived from other measurements. These measuremnts can be direct measurements or already derived measurements, leading to a hierarchy of derivations, which is called the data-flow graph for that measurement.

The data flow graph consist in a set of standard components that are linked together according to the spatial relationship graph [127]. The data flow graph can also be used to describe how an edge can be computed from other edges. The data flow graph has in addition to deal with error propagation according to chapter 4.2.

2.1.2 Visual Programming Toolbox

If we use the formal description to uniquely describe all algorithms that are used in the data flow graph, we can see this library of algorithms a s a toolbox for building data flow graphs.

We could develop a visual editor where it is possible to instantiate and connect data flow algorithms without writing a line of code. The resulting graph could be saved in a textual description and could be instantiated even without the editor. An overview of software frameworks to support this process is given in section 2.2. Some of the calibration algorithms are described later in section 4.2 and 7.

2.1.3 Automatic Pattern Search and Data-Flow Generation

With tracking setups becoming increasingly complex, it gets more difficult to find suitable algorithms for tracking, calibration and sensor fusion. Even in small setups, several possibilities exist how a particular calibration problem can be solved.

Pustka et.al. [141] describe a method how the formal description of a spatial relationship graph can be used to automatically derive all possible solutions, given a library of potential algorithms. The algorithms are organized as so-called *spatial relationship patterns*. A spatial relationship pattern is used to identify parts of the overall spatial relationship graph for which a known algorithm exists.

An algorithm is proposed which searches the graph recursively for potential patterns to identify again solvable subproblems until a solution for all required transformations is found. The corresponding data flow components then can be instatiated and execute the algorithms to provides the result again as an edge in the spatial relationship graph [141].

This enable the fully automatic setup of data flow graphs from a description of the setup alone. The choice between different potential solutions can be done either automatically using a pre-specified error function [200], or manually from a list of potential solutions.

2.2 Supporting Middleware

To support the ideas of the ubiquitous tracking framework, a middleware is necessary to connect the various data flow components with teach other and to allow the execution of predefined data flows.

2.2.1 Studierstube / OpenTracker

In 2001, Reitmyar et.al. [142] presented an XML-based framework for the flexible configuration of different tracking devices and intermediate dat flow components. It has been used as tracking layer in the *Studierstube* project [142, 178]. In this framework, every device defined an interface to a query for the current position and orientation. The components are formed into a directed acyclic graph (DAG) according to the instructions in a user-supplied configuration file.

The framework did not support dynamic reconfiguration of the specified graph, as it i is required for the automatic setup of data flow graphs using spatial relationship patterns.

2.2.2 DWARF

The *Distributed Wearable Augmented Reality Framework (DWARF)* [17] is a framework to build and deploy augmented reality applications [114]. The main focus was on mobile, highly flexible and continuously changing systems. Therefore the middleware had to be able to dynamically connect independent components rather than instantiating a complete predefined graph.

Several academic applications have been built using the DWARF framework between the years 2000 and 2005, and finally lead to the design of the *Ubitrack* middleware.

2.2.3 OSGAR

OSGAR [39] is an extended scene-graph API to support uncertain transformations. Initial errors from the tracking systems get propagated to the final application error [113]. The scene-graph is in fact a tree, not a general directed acyclic graph.

The tracking of objects is tightly coupled with the rendering, and the data-flow is explicitly modeled into the rendering tree. Automatic data-flow generation is not possible using this approach.

2.2.4 Ubitrack

The *Ubitrack* framework [140, 127] is a conceptual enhancement of the DWARF framework to overcome the implementational limitations of its predecessor. It additionally will support more features like the automated graph search described above and error propagation methods as described later in this thesis. In contrast to both DWARF [114] and OpenTracker [142], the ubitrack framework defines an XML description language to describe bith the dataflow and the spatial relationship graph at the same time. The development of the Ubitrack framework is an ongoing research effort at the Technische Universität München in the group of Prof. Klinker.

2.2.5 CampAR

CampAR [170] is another framework for augmented reality applications developed at the Technische Universität München, in the group of Prof. Navab. It is mainly targeted at medical augmented reality with their special requirements on high-precision tracking and reliability.

Like in the Ubitrack framework, the data flow is specified in an XML file which is used to instantiate and execute data flow components at runtime.

Part II

Modeling and Estimating Uncertain Transformations

3

Error Representations

The error in a tracking system can be described as a statistical process [28]. At any given time the pose estimate is the the result of the true position, affected by some static repeatable noise due to wrong calibration or static field distortion plus additional random noise, which are non-repeatable errors made by the measurement system. Such errors are inevitable in any kind of physical measurement and therefore exist in any tracking system. Since it is non-repeatable, it can also not be removed by calibration but is an error which is intrinsic to a specific tracking system setup.

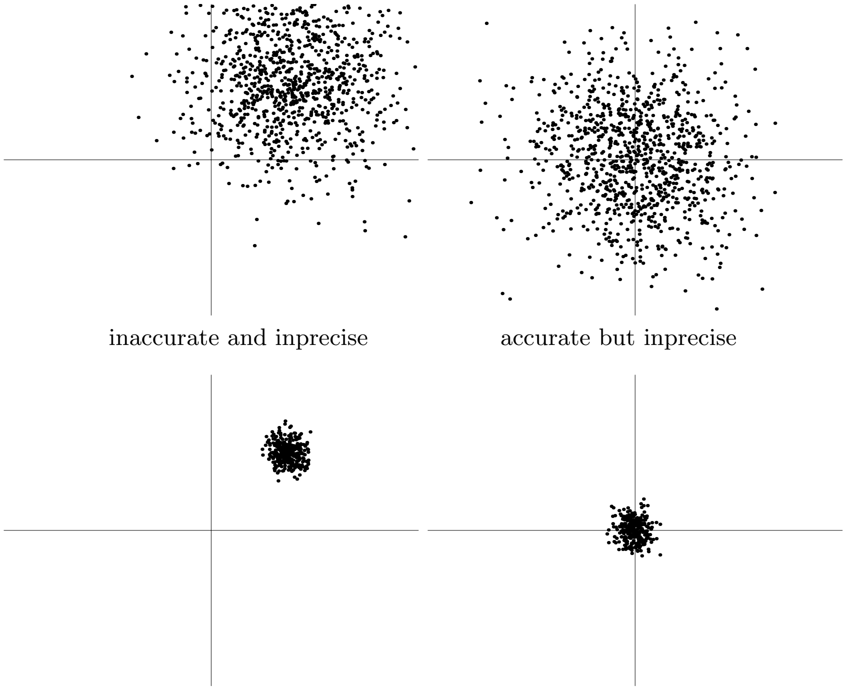
Since all calibrations are also based on measurements, the magnitude of this error limits also the achievable accuracy for calibration.

3.1 Accuracy and Precision

There are two concepts describing a statistical error that need to be distinguished: The accuracy and the precision of the measurement. The accuracy of a measurement is defined as the degree to which a measured value corresponds to the true value of the measured quantity, or in case of distributions the degree to which the distribution corresponds to the actual value.

The precision of a measurement indicates how close the measured values are together. A precise measuring instrument will give repeatable results, although they might be inaccurate if not properly calibrated. Figure 3.1 gives a graphical depiction of the concepts.

In general we can say that the accuracy of a measuring instrument can be increased by better calibration, while the precision of the measurements usually depends on the setup and the used hardware in the measuring instrument. However, it is not possible to get reliable accuracy for single measurements without a good precision. While the mean of a set of measurements with low precision might still represent the actual value, every single measurement can be way off.



inaccurate and inprecise

accurate but inprecise

inaccurate but precise

accurate and precise

Fig. 3.1. The concept of Accuracy and Precision

Still, we can say that after diligent calibration, i.e. when we can use the assumption that the estimated mean is in fact the correct value, the precision of the measuring device is a good estimate for the overall error of the measurements.

3.2 Gaussian Error Distributions

This section provides an overview how errors in tracking systems, represented by multidimensional Gaussian distributions, can be visualized to make them understandable even by inexperienced users of such tracking systems. One reason why the Gaussian distribution is of particular importance can be found in the *central limit theorem* [69] which states that under certain conditions the distribution of a sum of a large number of independent variables is approximately normal distributed. For this reason we can sufficiently model errors using Gaussian distributions, although the actual error sources have other error characteristics.

The chapter starts with probability density functions in several dimensions and continues with isocontours or confidence levels. After that, different matrix norms to reduce the dimension of the error are presented. To visualize rotational errors, different kinds of error propagations are explored.

3.2.1 Probability Density Functions

A probability function [69] $f(x)$ is a function which is non-negative and the integral

$$\int_{-\infty}^{+\infty} f(x)\,dx = 1$$

The probability for a random variable x to be in an interval $[a, b]$ is then given by the integral

$$\int_{a}^{b} f(x)\,dx$$

One-dimensional Gaussian distribution

For the special case of a one-dimensional Gaussian or Normal distribution (fig. 3.2), the probability density function is defined as

$$f_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

in the one-dimensional case where μ denotes the mean and σ the standard deviation (which is the square root of the variance).

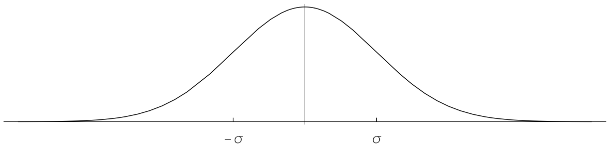


Fig. 3.2. Probability density function of a one-dimensional Gaussian distribution

Multi-dimensional Gaussian Distributions

In the n dimensional case (fig. 3.3), the probability density function is

$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{\frac{N}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma (x-\mu)} \tag{3.1}$$

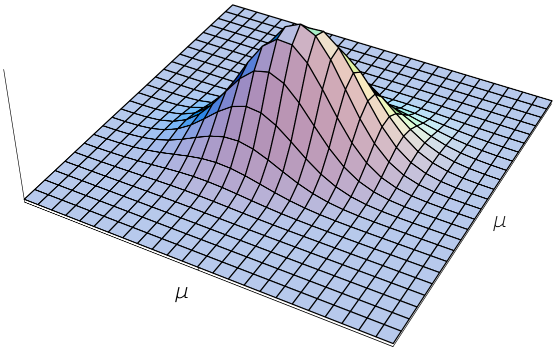


Fig. 3.3. Probability density function of a two-dimensional Gaussian Distribution

The covariance matrix Σ is an n-dimensional square matrix where the entry in the i-th column and j-th row denotes the covariance between the i-th and j-th entry in the random vector x. This implies that the covariance matrix is always symmetric and positive semi-definite, and therefore also all eigenvalues are non-negative.

$$\Sigma = \begin{bmatrix} \mathrm{Var}(x_1) & \mathrm{Cov}(x_1, x_2) & \cdots & \mathrm{Cov}(x_1, x_n) \\ \mathrm{Cov}(x_1, x_2) & \mathrm{Var}(x_2) & \cdots & \mathrm{Cov}(x_n, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(x_1, xn) & \mathrm{Cov}(x_2, x_n) & \cdots & \mathrm{Var}(x_n) \end{bmatrix}$$

For independent entries in the random vector, the covariance is 0, so a diagonal covariance matrix denotes a random vector with only independent entries.

3.2.2 Isocontours

Using the equations for the probability density functions of a normal distribution, we are going to show that the set of vectors with equal probability form a quadric in $\mathbb{R}^n$. This quadric is called the *surface of constant probability*, or *isocontour*. Additionally, the volume that is contained in this surface is the set of points in which the random variable x lies with some cumulative probability.

If we solve equation 3.1 for x, we get the set of vectors x which have the same probability p by the implicit equation

$$x^T \Sigma^{-1} x = -2 \ln(p\sqrt{2\pi}|\Sigma|^{\frac{1}{2}}) = z^2 \tag{3.2}$$

where we assume without loss of generality that the mean μ of the random variable is zero.

Then the cumulative probability for x being inside the surface corresponding to this implicit equation is given by [69]:

$$1 - P = \frac{n}{2^{n/2}\,\Gamma(\frac{n}{2}+1)} \int_z^\infty x^{n-1} e^{x^2/2} dx \tag{3.3}$$

where $\Gamma(x)$ is the Gamma function, an extension of the factorial function to real numbers. This function gives the probability of x being inside of the surface defined by $z^2 = x^T \Sigma^{-1} x$. The inverse of this function is called the quantile function because it returns the corresponding value for z for a given cumulative probability P. It can be shown that there exists no closed term of the quantile function. Solving the equation numerically for different values of P gives the values for z for some confidence levels as shown in table 3.1 in various dimensions. Figure 3.4 shows the cumulative probability function with the values for the confidence level of 95% indicated.

	Confidence Level					
Dimension	25%	50%	75%	95%	97%	99%
$n = 1$	0.10153	0.45494	1.3233	3.84146	4.70929	6.6349
$n = 2$	0.57536	1.38629	2.77259	5.99146	7.01312	9.2103
$n = 3$	1.21253	2.36597	4.10834	7.81473	8.94729	11.3449

Table 3.1. $z^2 = x^T \Sigma^{-1} x$ for different confidence levels and dimensions

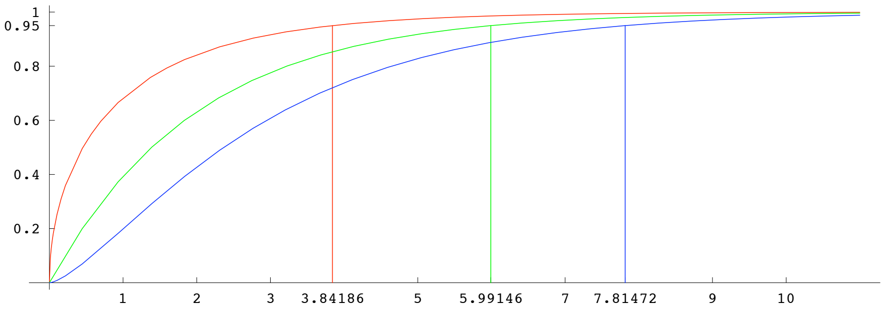


Fig. 3.4. Cumulative probability of Gaussian distributions; confidence levels of 95% shown for dimension $n = 1$ (red), $n = 2$ (green), and $n = 3$ (blue).

Looking again at equation 3.2 we can see that this equation defines a quadric for any given z. A quadric is a a special polynomial form which is defined by the equation

$$\mathrm{q} : \mathbb{R}^n \to \mathbb{R}, x \mapsto \mathrm{q}(x) = x^T A x$$

for a symetric matrix $A \in \mathbb{R}^{n \times n}$. We have already seen that Σ is in fact positive semi-definite, so the quadric has always the special form of an ellipsoid. An ellipsoid with the three principal axes aligned with the coordinate axes is defined by

$$\frac{x_1^2}{a} + \frac{x_2^2}{b} + \frac{x_3^2}{c} = 1 \quad \text{where} \quad a, b, c > 0$$

where a, b, c are the length of the three coordinate axes. This equation can also be rewritten as

$$\frac{x_1^2}{a} + \frac{x_2^2}{b} + \frac{x_3^2}{c} = x^T \begin{bmatrix} \frac{1}{a} & & \\ & \frac{1}{b} & \\ & & \frac{1}{c} \end{bmatrix} x = x^T A x$$

It is also easy to see that an ellipsoid that is rotated by a rotation matrix R fulfills the equation

$$1 = x^t (R^t A R) x = (Rx)^t A (Rx)$$

The Eigenvalue decompositiion of $\Sigma = RDR^T$ gives a diagonal matrix D containing the eigenvalues in the diagonal entries and a rotation matrix R consisting in the corresponding eigenvectors. Together with equation 3.2 we get for a certain value for z the equation for the quadric as

$$1 = x^T \left(\frac{1}{z^2} \Sigma^{-1} \right) x = x^T \left(\frac{1}{z^2} R^T D^{-1} R \right) x$$

From this equation it can be seen that the principle axes of the ellipsoid are along the columns of R which are the eigenvectors of the covariance matrix and have the length $l_i = z\sqrt{\lambda_i}$ $(i = 1, 2, 3)$.

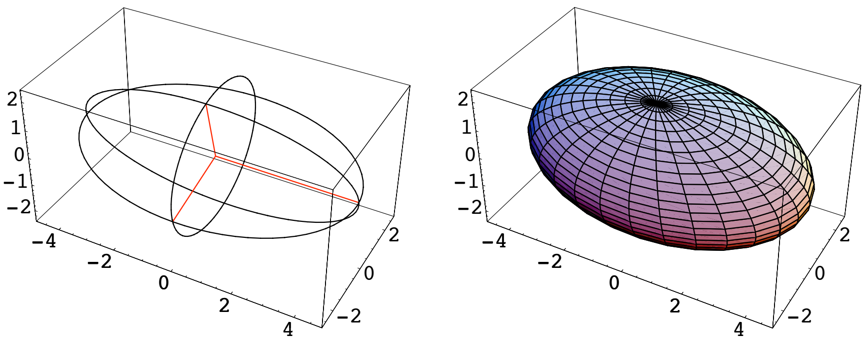


Fig. 3.5. Three-dimensional Gaussian distributions. Two different visualizations of the covariance ellipsoid for a certain confidence level.

This can be used now to plot the covariance ellipsoid for a certain confidence level as seen in figure 3.5. The corresponding value for z can be found by numerically solving equation 3.3 or using lookup-tables like table 3.1 or figure 3.4.

3.2.3 Matrix Norms

Higher dimensional Gaussian distributions are sometimes hard to visualize. A simple projection to a subspace loses all information orthogonal to the subspace, e.g. in figure 6.5 later in this thesis, the vertical component of the covariance is lost. A matrix norm $p(A) = \|A\|$ should fulfill the following requirements:

Positivity

The accuracy is defined as a confidence interval around the expected value and therefore should be always positive.

$$p(A) > 0 \qquad \forall A$$

Positive Homogeneity

A scaling of the accuracy in the original space should result in the same scaling in the matrix norm. This requirement also includes linearity of the norm.

$$p(\alpha \cdot A) = |\alpha| \cdot p(A) \qquad \forall A, \alpha$$

Triangle Inequality

When combining two covariance matrices, the norm of the combined covariance should not be smaller than the combined norms of the single covariances.

$$p(A_1 + A_2) \leq p(A_1) + p(A_2) \qquad \forall A_1, A_2$$

where the matrix addition is an element-wise addition of two covariances in the same reference frame as seen in section 3.4.3.

Positive Definiteness

The norm of the covariance should be only zero when there is no error, i.e. the covariance itself is zero.

$$p(A) = 0 \Leftrightarrow A = 0 \qquad \forall A$$

In the following we present two matrix norms and their geometric meaning for covariance matrices.

Spectral Norm

The spectral norm of a positive-semidefinite matrix A is the square root of the largest eigenvalue of A. As we have seen in section 3.2.2, the square root of the eigenvalues define the main axes of the surfaces of constant probability. Therefore the spectral norm of the covariance matrix can be seen as the absolute value of the maximum error (with some probability) Figure 3.6 shows a visualization of three-dimensional covariances in a plane by their spectral norm.

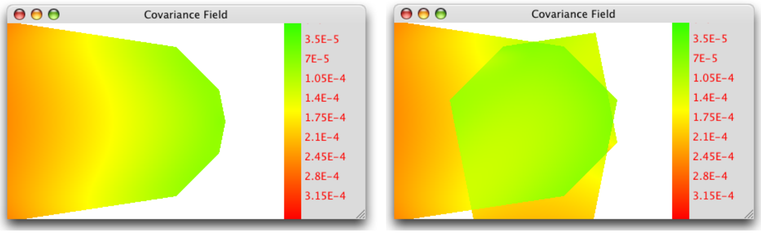


Fig. 3.6. Spectral norm in a two-camera (left) and three-camera (right) setup

The spectral norm can be of special interest in cases where the error is considerably larger in one direction than in the other directions. The spectral norm, in particular when compared with the trace norm, can cover this non-isotropic behavior by showing the maximum error.

Trace Norm

The trace norm, also called *Frobenius norm*, is defined as the square root of the sum of the diagonal elements. This is – as we will see in section 3.3 – equivalent to the root mean square error which is also widely used. Figure 3.7 shows a visualization of three-dimensional covariances in a plane by their trace norm.

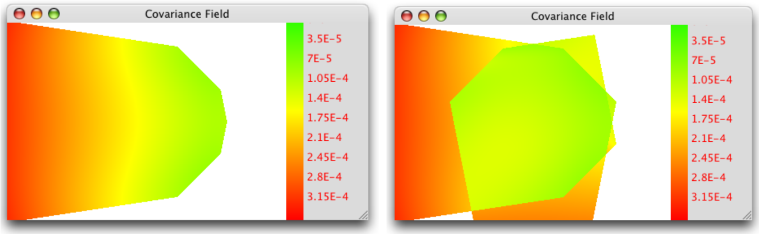


Fig. 3.7. Trace norm in a two-camera (left) and three-camera (right) setup

The trace norm is always smaller than or equal to the spectral norm; the quotient of trace norm and spectral norm can be used as a measure for the isotropy of the covariance.

3.3 Root Mean Square Error

Another useful representation of errors in measurement systems is the so called *root mean square (RMS)* error, which is defined as the root of the mean of the squared absolute errors. The RMS error is an absolute error measure. It calculates the error relative to a known ground truth value. Let x_i be a set of the measurements with $i \in \{1 \ldots N\}$ and x the real value. Then the RMS error e_{rms} is computed as

$$e_{\mathrm{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |x - x_i|^2}$$

This computation depends obviously on the used norm. In most cases, arithmetic distance is used, but other vector norms are possible depending on the application. The RMS error is a one-dimensional error measure even for multidimensional random variables.

If the underlying distribution is a one-dimensional Gaussian distribution with known mean $\bar{x}$ and standard deviation σ_x, we have an easy way of calculating the RMS error for that distribution:

$$e_{\mathrm{rms}} = \sqrt{\bar{x}^2 + \sigma_x^2}$$

since

$$\begin{aligned}
\bar{x}^2 + \sigma_x^2 &= \bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \\
&= \bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N}(x_i^2 + \bar{x}^2 - 2x_i\bar{x}) \\
&= 2\bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \frac{2}{N}\bar{x}\sum_{i=1}^{N} x_i \\
&= 2\bar{x}^2 + e_{\mathrm{rms}}^2 - 2\bar{x}^2 \\
&= e_{\mathrm{rms}}^2
\end{aligned}$$

The RMS of a zero-mean Gaussian distribution is therefore equivalent to the variance of the distribution. For a multidimensional Gaussian distribution with mean μ and covariance matrix Σ we can get a similar result [98]:

$$e_{\mathrm{rms}} = \sqrt{tr(\varSigma) + |\mu|^2}$$

Since the RMS error is only a single value even for multidimensional distributions, there exist many distributions that correspond to the same RMS value. However, if we assume unbiased, independent, and isotropic errors, we know that the trace of the covariance matrix should be

$$tr(\varSigma) = \sum_{i=1}^{n} \sigma_i^2 = e_{\mathrm{rms}}^2$$

and therefore due to the isotropy and independence $\sigma_i^2 = \frac{1}{n} e_{\mathrm{rms}}^2$. So we can get the corresponding Gaussian distribution with mean $\mu = 0$ and

$$\varSigma = \frac{1}{n} \begin{pmatrix} e_{\mathrm{rms}}^2 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & e_{\mathrm{rms}}^2 \end{pmatrix}$$

Section 6.4 shows an example of such a covariance matrix corresponding to a certain RMS value.

3.4 Error Propagation for Gaussian Distributions

Error propagation is the problem of finding a distribution of a function of random variables. It can be applied in all cases where we have a mathematical model of the function as well as knowledge about the distribution of the random variables. The distribution does not necessarily need to be Gaussian, but most error propagation rules assume Gaussian distributions.

In this case, the distribution can be fully specified by their first two statistical moments, the mean and the variance of the distribution. Therefore it is enough to provide propagation rules for these two parameters [5].

3.4.1 Forward Propagation

A nice property of Gaussian distributions is the fact that a linear function applied to a Gaussian distribution results again in a Gaussian distribution. A linear function is described by a matrix A, and it can be shown [69] that the mean of the transformed distribution is obtained by applying the function to the mean of the original distribution, $f(\mu)$.

The *forward propagation of covariance* rule for an affine function f is defined as [78]: Let v be a random vector in $\mathbb{R}^M$ with mean $\bar{v}$ and covariance matrix $\varSigma$, and suppose that $f : \mathbb{R}^M \mapsto \mathbb{R}^N$ is an affine mapping defined by $f(v) = f(\bar{v}) + A(v - \bar{v})$. Then $f(v)$ is a random variable with mean $f(\bar{v})$ and covariance matrix $\varSigma_f$:

$$\Sigma_f = A\Sigma A^T \tag{3.4}$$

If the function f is nonlinear, the propagated distribution can easily become significantly different from a Gaussian distribution and therefore hard to handle. However, for small errors the Gaussian distribution still serves as a good approximation. To be able to apply the Gaussian forward propagation to a general nonlinear function, we need to build the first order approximation of the function f. The first order approximation can be found using the (multidimensional) Taylor expansion:

$$f(x) = f(x_0 + \Delta_x) = f(x_0) + J_f(x_0)\Delta_x + \mathcal{O}(x^2)$$

where $J_f(x_0)$ is equal to the Jacobian matrix evaluated at the point x_0:

$$J_f(x_0) = \left.\frac{\delta f}{\delta x}\right|_{x=x_0} = \left.\begin{bmatrix} \frac{\partial \mathrm{f}_1(x)}{\partial x_1} & \cdots & \frac{\partial \mathrm{f}_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{f}_m(x)}{\partial x_1} & \cdots & \frac{\partial \mathrm{f}_m(x)}{\partial x_n} \end{bmatrix}\right|_{x=x_0}$$

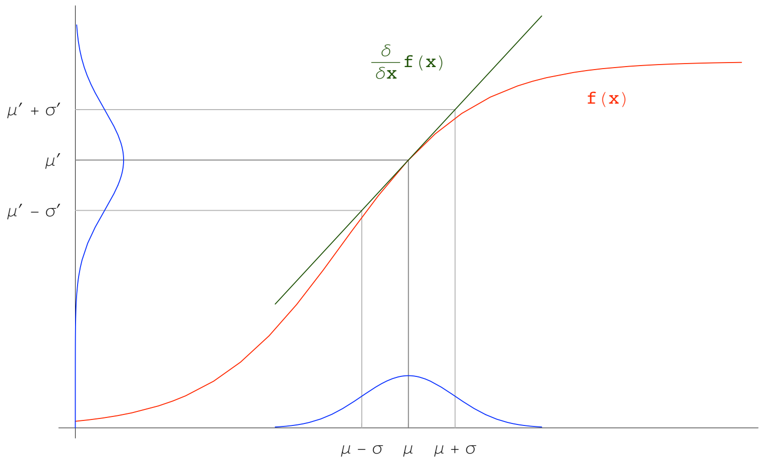


Fig. 3.8. One-dimensional case of a nonlinear error propagation of a Gaussian distribution The mapping function is approximated by the Jacobian and the distribution is transformed using the linearized function.

Now we can define the linearized error propagation rule for non-linear functions as: Let v be a random vector in $\mathbb{R}^M$ with mean $\bar{v}$ and covariance matrix Σ, and suppose that $f : \mathbb{R}^M \mapsto \mathbb{R}^N$ is differentiable and approximately linear in a neighborhood of $\bar{v}$. Then, up to a first order approximation, $f(v)$ is a random variable with mean $f(\bar{v})$ and covariance matrix Σ_f:

$$\Sigma_f = J_f \Sigma {J_f}^T \tag{3.5}$$

where J_f is the Jacobian matrix of f evaluated at $\bar{v}$. Figure 3.8 shows a one-dimensional visualization of this concept.

3.4.2 Backward Propagation

In some cases we do know the covariance of the function $f(v)$ and want to estimate the covariance of the parameters v. We could apply forward propagation on the inverse function f^{-1}, but what if the inverse function is hard to compute? Instead of computing $J_{f^{-1}} \Sigma {J_{f^{-1}}}^T$ we can show [78] that this is equal to computing

$$\Sigma_{f^{-1}} = ({J_f}^T \Sigma^{-1} J_f)^{-1}$$

which is easier to achieve. Again, this can be visualized intuitively using figure 3.8. Instead of computing the potentially complicated Jacobian of the inverse function f^{-1}, we simply invert the already linearized solution. This can be performed easily.

Additionally, we get a solution for the overparametrized case when we replace the matrix inversion by the *pseudoinverse*

$$\Sigma_{f^{-1}} = ({J_f}^T \Sigma^{-1} J_f)^{+} \tag{3.6}$$

This propagation rule can therefore be also applied for least-square solutions (cf. section 5.3.5).

3.4.3 Combination of Random Variables

In some cases it is necessary to combine two or more distributions into a single one. We have to distinguish two cases here: Either the two distributions describe independent parameters of the combination function, or we have a fusion of two distributions describing the same parameters [84].

Combination of Random Variables

A function f can have several random variables as input. The distribution of each of these random variables adds to the final error [85]. This is for example the case for the concatenation of two transformations as described in section 4.2.2.

We assume we have a function $f(x_1, x_2)$ where x_1, x_2 are random variables with covariances Σ_1 and Σ_2 and we are interested in the combined covariance Σ_c. Then f could as well be described as a function $f(x)$ where $x = (x_1, x_2)$ is the combination of both input variables. Since we know that the input variables are independent, the Jacobian is a block matrix

$$J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \quad , \quad J_1 = \frac{\delta f}{\delta x_1} \quad , \quad J_2 = \frac{\delta f}{\delta x_2} \tag{3.7}$$

where the upper left block is only derivatives of x_1 and the lower right only derivatives of x_2. Applying the error propagation from equation 3.8 this equation can be rewritten as

$$\Sigma_c = J_1 \Sigma_1 J_1^T + J_2 \Sigma_2 J_2^T \tag{3.8}$$

Literally this means that all covariances are transformed into the same target coordinate system and then added element by element [85].

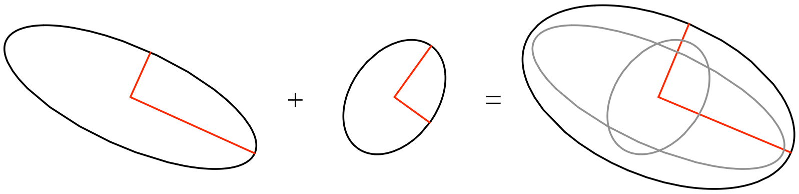


Fig. 3.9. Combination of two covariances. The resulting covariance is larger than the original covariances.

Fusion of Random Variables

The fusion of random variables is necessary when there are two different – but independent – estimates for the same parameters. This is for example the case in filtering applications like the Kalman filter [9, 28].

Let X_1 and X_2 be two independent random variables with their associated covariance matrices Σ_1 and Σ_2, then the combined random variable X_C is given by the weighted average of the two random variables using the covariance matrices as weights:

$$X_C = \Sigma_2(\Sigma_1 + \Sigma_2)^{-1} X_1 + \Sigma_1(\Sigma_1 + \Sigma_2)^{-1} X_2 \tag{3.9}$$

and the corresponding covariance matrix by

$$\Sigma_c = \Sigma_2(\Sigma_1 + \Sigma_2)^{-1} \Sigma_1 \tag{3.10}$$

The covariance matrix of the fused estimate is always smaller than the two original covariance matrices.

This equation can easily be extended to a fusion of more than two random variables.

3.4.4 Monte-Carlo Simulation

Monte-Carlo simulation is a method of simulating the behavior of complex systems. They are in particular well suited to simulate the error propagation

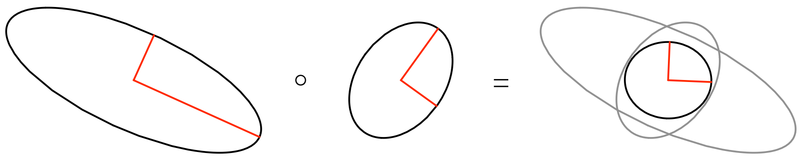


Fig. 3.10. Fusion of two covariances. The resulting covariance is smaller than the original covariances.

in functions where analytic error propagation would be hard to achieve. It also can handle arbitrary non-standard input distributions.

To compute the Monte-Carlo simulation of a function $f(x)$ where x is a random variable with some (not necessarily Gaussian) distribution, we draw random samples x_i from the distribution of x and compute the function value $f(x)$. Note that x can be a vector with several possibly independent random variables. From the resulting point set we can then compute again a distribution function. Figure 4.5 shows an example for the Monte-Carlo simulation of the concatenation of two rigid transformations with error.

As in all simulations, ground truth data can be used both to generate the input distributions and to validate the results in the end. This makes the Monte-Carlo simulation a very useful tool for offline accuracy analysis [219].

The main problem that arises when using Monte-Carlo simulations is the number of samples needed to get statistically significant results. This number increases exponentially with the number of dimensions in the random variable vector. Therefore, Monte-Carlo simulations have only limited use when the computing time is restricted, as in online error estimates.

3.4.5 Unscented Transform

The unscented transform is a second-order error propagation for gaussian distributions [95]. It has been proposed initially for nonlinear Kalman filtering [97, 94] but can be used for any kind of nonlinear error propagation.

The concept is similar to the Monte-Carlo simulation: A set of sample points are drawn from the distribution of the random variable and transformed using the function f. However, in contrast to the Monte-Carlo method, the sample points are not drawn randomly. A set of so-called *sigma points* is chosen deterministically such that the covariance matrix of the point set is equal to the original covariance matrix. These points are transformed using the function f and then the covariance of the resulting point cloud is computed as in Monte-Carlo simulation [96]. It can be shown that the points propagated through the true non-linear system capture the posterior mean and covariance accurately to the third order of the Taylor series expansion for any nonlinear function [202].

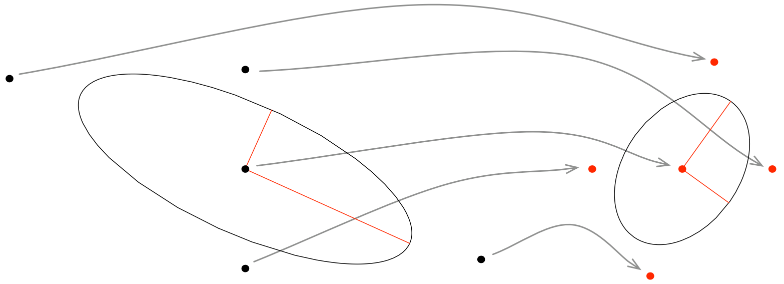


Fig. 3.11. Transformation of the sigma points

There are many different possibilities for choosing the set of sigma points. Any set of points where the covariance matrix of the point set is equal to the original covariance is a valid set of sigma points. One possibility is [95] to compute the Cholesky decomposition $Chol(\Sigma)$ of the covariance matrix and use the columns with positive and negative sign together with the mean as sigma points.

$$P_\sigma = \begin{Bmatrix} \pm \, Chol(\Sigma)_{1 \ldots n} \\ 0 \end{Bmatrix}$$

Figure 3.12 shows the sigma points for a three-dimensional Gaussian distribution using this construction.

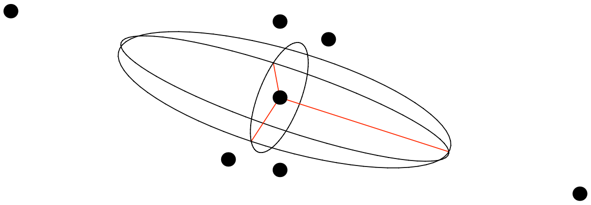


Fig. 3.12. Example sigma points for the error propagation with unscented transform of a three-dimensional covariance. The ellipsoid shows the standard deviation of the covariance matrix

One problem with the standard unscented transform is the fact that some of the sigma points (especially for higher dimensional random variables) tend to lie far away from the mean. Therefore, the approximation of the function f is not anymore local around the true value. Figure 4.7 later in this thesis shows this effect already for a three-dimensional covariance, and it is even more prevalent in higher dimensions [190]. To overcome this drawback, a variant of the unscented transform has been proposed where the sigma points are scaled towards the mean and this scaling is corrected after the transformation [93].

4

Representing Uncertain Transformations

Since almost all transformations in augmented reality systems are affected by errors, we need a consistent way to represent these errors [39] to allow a propagation of the errors. Uncertain transformations are given as a tupel (T, e) of the transformation T in some kind of representation and the associated error e in some other representation. Note that the representation of transformation and the representation of the error do not necessary need to be based on the same concept. We will see how for example a rotation can be represented by a matrix and at the same time the according error as a covariance matrix of the corresponding euler angles.

4.1 Representing Rigid Transformations

A rigid transformationis a mathematical way to describe the motion of a rigid object in n-dimensional space without changing its size or shape. Rigid transformations are also called isometries because they preserve the same measures for lengths and angles. The set of all isometries is called the *Euclidean Group*, which includes additionally reflections.

All rigid transformations are in fact affine, which means that the transformation T can be represented by

$$x' = R\,x + t \tag{4.1}$$

where $A \in \mathbb{R}^{n \times n}$ is a matrix and $t \in \mathbb{R}^n$ is a vector.

4.1.1 Homogeneous Matrix Representation

A linear transformation is a special case of an affine transformation which additionally fulfills the requirements of *additivity* and *homogenity.* Since equation 4.1 is not a pure linear transformation (the law of homogenity is violated), we move from n-dimensional euclidean space $\mathbb{R}^n$ to $n+1$-dimensional projective space $\mathbb{P}^n$ by defining equivalence classes of vectors αx with the *homogeneous vector* $x = (x_1, \ldots, x_{n-1}, 1)$ as its representative. This allows us to reformulate equation 4.1 as a linear equation using block matrix notation:

$$x' = \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} x \tag{4.2}$$

This matrix representation is called *homogeneous matrix* representation and is now a linear transformation. This matrix representation is easy to use; the concatenation of multiple rigid transformations can be computed by the matrix product of the respective homogeneous matrices, and the application of a transformation to a vector can be computed by the matrix vector product.

When we look at the rotation matrix R we can see that to be an isometry, the matrix needs to be orthonormal; i.e. the columns of the matrix need to be of unit length and be mutually orthogonal. There are $\binom{n}{2}$ different pairs of columns and n equations that need to be of unit length, leading to a total of $n(n+1)/2$ independent (nonlinear) equations which reduces the n^2 degrees of freedom of the general n-dimensional matrix to $n^2 - n(n+1)/2 = n(n-1)/2 = \binom{n}{2}$ degrees of freedom for the set of orthonormal matrices. It can be shown that the set of n-dimensional orthonormal matrices under matrix multiplication forms a group, which is called the *orthogonal group* $O(n)$.

In addition it is easy to see that all homogenous matrices A which represent rigid transformations need to have determinant $det(A) = 1$ to exclude reflections. Adding this requirement leads to the *special orthogonal group* $SO(n)$, the group of n-dimensional rotation matrices.

Now it is obvious to see that rotations in $\mathbb{R}^2$ have one degree of freedom, and rotations in $\mathbb{R}^3$ have three degrees of freedom.

Since orthogonal matrices have the useful property that $A^T A = I$, we can compute the inverse of a homogeneous matrix efficiently and numerically stable as

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^{-1} & -R^{-1}t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} \tag{4.3}$$

However, when we try to find such a matrix in nonlinear estimation (cf. section 7.2), $n(n+1)/2$ additional constraints would need to be added to ensure the matrix to be orthogonal [107]. Direct linear interpolation is also not possible using this matrix representation of rotations.

4.1.2 Euler Angles

To overcome the drawbacks of using homogeneous matrices to represent rotations, several other representations exist.

In three-dimensional euclidean space $\mathbb{R}^3$, a general rotation can be decomposed into three consecutive rotations around three distinct but fixed axes through the origin [193]. Since the arbitrary rotation has three degrees of freedom, three rotations are enough to specify the rotation. While arbitrary axes are possible here, it is most convenient to use rotations around the principal orthogonal axes since this gives the following simple formulas:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These can be used to parametrize arbitrary rotations as a triple (α, β, γ) of the respective rotation angles. This representation is called Euler angles since they have first been described by Leonhard Euler [53]. However, there is an ambiguity in this description when the rotation axes are not explicitly defined: There are twelve different possibilities to apply a sequence of three of these rotations with no two consecutive rotations having the same axis [91]. therefore the exact sequence of rotation axes needs to be defined, which may lead to confusion if only Euler angles are given. Rotation axes can be defined in world-fixed coordinates, or relative to the moving object. It can be shown that these two different points of view simply correspond to an inversion of the sequence of rotations.

One special case of Euler angles is the *yaw, pitch, roll* representation which is commonly used in ship and airplane navigation [28]. Here the three rotation axes are explicitly named with intuitive terms: *roll* is the rotation R_x around the x-axis (which is assumed to be aligned with the main axis of the airplane or ship) with angle α; *pitch* is the rotation R_y around the y-axis, and *yaw* is the rotation R_z around the vertical z-axis. The order is given by the order of words, and all coordinate axes are local since it is used from inside a ship or airplane.

$$x' = R_x R_y R_z x$$

There is one major drawback connected to the representation of rotations using Euler angles: For $\beta = \pm\frac{\pi}{2}$, the rotation matrix collapses to

$$R = \begin{bmatrix} 0 & \sin(\alpha - \gamma) & \cos(\alpha - \gamma) \\ 0 & \cos(\alpha - \gamma) & -\sin(\alpha - \gamma) \\ -1 & 0 & 0 \end{bmatrix}$$

which means the rotation axes of α and γ are actually aligned in space. While in all other situations smooth changes of the rotation in all directions are possible using smooth changes in the angles, in this case some rotations are not possible. Close to this situation, infinitesimally small changes in rotation require infinitely large changes in the angle parameters. This is called *gimbal lock* since it originates in an actual physical problem when a gimbal is used for example in a gyrocompass to determine orientation. There exist combinations of the gimbal joints where certain rotations can not be accomplished by the gimbal (it is 'locked' in this direction) and therefore could not be measured, which would lead to significant drift in the orientation determination. Spaceship engineers therefore avoided getting too close to the locking configuration [82]. The avoidance of gimbal lock was the reason why the quaternion representation of orientation started to find extensive use in the aerospace community.

Conversion from Euler Angles to Rotation Matrix

Whenever in the context of this thesis we use Euler angles, we use the following conversion to a rotation matrix:

$$R(\alpha, \beta, \gamma) = R_x(\alpha) R_y(\beta) R_z(\gamma)$$

which is the *fixed angle* x, y, z representation.

Conversion from Rotation Matrix to Euler Angles

The back-conversion to the same representation can be found when inspecting the full rotation matrix in analytical form:

$$\alpha = \tan^{-1}\left(\frac{R_{3,2}}{R_{3,3}}\right) \qquad \beta = -\sin^{-1}(R_{3,1}) \qquad \gamma = \tan^{-1}\left(\frac{R_{2,1}}{R_{1,1}}\right)$$

where $R_{i,j}$ is the entry in the i-th column and the j-th row of the matrix R.

We denote this conversion function by $Eu(.)$ with

$$Eu\big(R(\alpha, \beta, \gamma)\big) = (\alpha, \beta, \gamma)$$

when we refer to it later.

4.1.3 Axis Angle Representation

The fundamental theorem of Leonhard Euler states that every finite three-dimesional rotation of a rigid body with respect to a fixed reference frame can be described as a simple rotation by some angle θ around some axis v [52]. In contrast to the representation of rotation as a orthonormal rotation matrix using euler angles, we represent now the rotation as a tupel (a, θ) of the rotation axis a and the rotation angle θ. This tupel can be used to represent any kind of rotations. Since only the direction of the rotation axis a is needed to specify the rotation, a has only two degrees of freedom and we can use the length of the vector to encode the rotation angle into a single vector $v \in \mathbb{R}^3$:

$$\theta = |v| \qquad a = \frac{v}{|v|}$$

Conversion from Axis-Angle to Rotation Matrix

Olinde Rodrigues gave the following formula to compute the rotation matrix $R \in \mathbb{R}^{3\times 3}$ corresponding to an axis-angle rotation v [175]:

$$R = I_3 + \frac{\sin\theta}{\theta} Skew(v) + \frac{1-\cos\theta}{\theta^2} Skew(v)^2$$

where $Skew(v)$ is the skew-symetric matrix

$$Skew(v) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{4.4}$$

Note that the rotation angle θ can be directly used as a metric for the distance between rotations [91]. It is however important to consider the fact that the angle θ is cyclic modulo 2π and usually constrained between for example $-\pi < \theta < \pi$, which needs to be taken into account for measuring distances and for interpolation.

Conversion from Rotation Matrix to Axis-Angle

The axis of rotation of a rotation matrix is the set of points that do not move when the matrix is applied. According to Euler's theorem, this axis exists for every rotation. By definition of an eigenvector, the axis of rotation a is exactly the eigenvector corresponding to a unit eigenvalue [91].

The angle θ can be found by using the remaining two eigenvalues which are at $\cos\theta \pm i \sin\theta$ [155].

Composition of Rotations in Axis-Angle Representation

A composition of rigid transformations as well as the inverse of a rigid transformation is again a rigid transformation. This implies that the combination of rotations around distinct axes is again a rotation [52]. However, it is not easy to compute the rotation axis of the composition of two rotation from the axis-angle representation directly without going through the matrix representation and back – a numerically not always recommended process.

4.1.4 Quaternion Representation

Quaternions were discovered in 1843 [42] by William Hamilton as an extension of complex numbers to four dimensions [75, 76] and independently by Olinde Rodrigues [42]. A quaternion is a four-tupel $q = (q_x, q_y, q_z, q_w) \in \mathbb{R}^4$ which can be seen as a real number w with three imaginary parts,

$$(q_x, q_y, q_z, q_w) = q_w + \mathbf{i}q_x + \mathbf{j}q_y + \mathbf{k}q_z$$

or in vector notation $q = (q_v, q_w)$ where $q_v \in \mathbb{R}^3$ and $q_w \in \mathbb{R}$. Quaternions can be used to either represent real numbers ($q_x = q_y = q_z = 0$), three-dimensional vectors ($q_w = 0$) or rotations as we will see later.

Quaternion Addition

Quaternion addition $\oplus$ is defined as standard vector addition [136]

$$(q_x, q_y, q_z, q_w) \oplus (p_x, p_y, p_z, p_w) = (q_x + p_x, q_y + p_y, q_z + p_z, q_w + p_w)$$

which is obviously associative and commutative and preserves the rules for the addition of vectors and real numbers.

Quaternion Multiplication

Quaternion multiplication $\otimes$ is defined similar to polynomial multiplication and together with the rules $i^2 = j^2 = k^2 = ijk = -1$ we get in vector notation

$$(q_v, q_w) \otimes (p_v, p_w) = (q_w p_w - q_v \cdot p_v, \quad q_v \times p_v + p_w q_v + q_w p_v)$$

which is not commutative, but associative and distributive over addition.

$$q \otimes (x \oplus y) = (q \otimes x) + (q \otimes y)$$

Quaternion multiplication also preserves the rules for the multiplication of real numbers.

Unit Quaternions

The norm of a quaternion q is defined as $\sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2}$. A quaternion with unit norm is called a unit quaternion. It can be shown that the set of unit quaternions under quaternion multiplication form a subgroup of the full quaternion group [91]. This means that the multiplication of two unit quaternions is again a unit quaternion; a useful property that we will need to represent rotations later.

Conjugate Quaternion

Similar to complex numbers, the *conjugate* q^* of a quaternion q is computed by negating the vectorial part,

$$q^* = (-q_v, q_w) = (-q_x, -q_y, -q_z, q_w)$$

Unit quaternions have the useful property that their inverse is simply their conjugate: It is easy to see that with this equation, for unit quaternions

$$q^* \otimes q = q \otimes q^* = 1$$

The conjugate operation is distributive over quaternion addition, i.e. $(q \oplus p)^* = q^* \oplus p^*$ but for multiplication we get $(q \otimes p)^* = p^* \otimes q^*$, similar to the inverse of the product of two matrices.

Representing Rotations

One important property of quaternions is the fact that they can be used to represent rotation [16, 86, 194, 175]. Using unit quaternions and the multiplication rules of the quaternion algebra we can now define the quaternion based on the axis angle notation from section 4.1.3 with rotation axis a and rotation angle θ.

$$q = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}a\right)$$

To rotate a point x we write x as a pure quaternion $\hat{x} = (0, x)$ with zero rotation angle. The rotation is then defined as

$$\hat{x}' = q \otimes \hat{x} \otimes q^*$$

where q^* is the conjugate of q. The concatenation of two rotations p and q is possible by quaternion multiplication $p \otimes q$.

$$\hat{x}' = (p \otimes q) \otimes \hat{x} \otimes (p \otimes q)^* = p \otimes (q \otimes \hat{x} \otimes q^*) \otimes p^*$$

Representing Six-dimensional Transformations

A six-dimensional rigid transformation is represented by a tupel (p, q) of a rotation quaternion q and a pure quaternion p representing the translation. Applying a six-dimensional transformation (p, q) to a single point $\hat{x} = (0, x)$ is achieved by first applying the rotation and then the translation:

$$\hat{x}' = q \otimes \hat{x} \otimes q^* \oplus p$$

The composition (p_c, q_c) of two such transformations (p_1, q_1) and (p_2, q_2) is then given accordingly:

$$(p_c, q_c) = (p_1, q_1) \circ (p_2, q_2) = ((q_1 \otimes p_2 \otimes q_1^*) \oplus p_1, q_1 \otimes q_2)$$

which can be shown by expanding the equation.

Interpolation

Since (unit) quaternions are spherical parametrizations, it is not straightforward to use euclidean metrics for the difference of two rotations, or to interpolate between two rotations. There is a special quaternion interpolation algorithm called *Slerp* which provides a smooth way to interpolate between rotations [43, 100, 164, 91]. This provides a useful tool for animations as well as optimization and filtering applications [139].

4.1.5 Dual Quaternions

Chasle's theorem [32, 124] states that any rigid transformation can be described by a screw motion, i.e. a rotation about an axis not through the origin and a translation parallel to this axis [45]. The screw axis is a line in space which requires four parameters, and the rotation and translation can be described by two single parameters, which results in the six degrees of freedom we would expect for a rigid transformation [34]. The screw parametrization has been used frequently to represent arbitrary rigid transformations [42].

In contrast to the previous parametrizations of rotations around the origin, the screw notion gives us the possibility to provide a closed representation of an arbitrary rigid transformation, but with less redundancy than homogeneous transformation matrices. One algebraic possibility of representing screw motions is through the use of unit dual quaternions. Dual quaternions are defined like quaternions with dual numbers. Unit dual quaternions have six degrees of freedom and therefore can be used to uniquely represent a rigid transformation [45]. Using this representation it is possible to define non-linear optimization functions that optimize rotation and translation simultaneously.

4.1.6 Exponential Maps

A variety of other parametrizations of the rotational group $SO(3)$ exist, most notably *exponential maps* [73, 16] which have been used in several applications [50], but in general they have only slight advantages for special purposes and therefore have only limited prevalence throughout the community.

4.2 Error Propagation

In this section we combine the representation of transformations with the gaussian error propagation methods to provide the tools that are needed to propagate gaussian errors through general augmented reality tracking systems.

4.2.1 Representing Gaussian Errors

The representation of errors in the measurement of rigid transformations is closely connected with the representation of the transformation itself. While a redundant representation, as for example homogeneous transformation matrices, only consume more memory and sometimes more computing time when used to represent the actual rotation, it is even more important to reduce the number of parameters when representing errors.

The assumption of statistically not independent parameters leads to large covariance matrices with even more redundancy to represent the Gaussian distributions. In the following we analyze two frequently used representations, one using small quaternions [139] and the other one using small rotations [47].

However, there is actually not much difference in the two representations since the formulations discussed above all yield the same first order approximation [107]. As long as only first order error propagation is used, all representations give the same result.

In the vector quaternion representation, a measurement for a rigid transformation is parametrized using a vector t for the translation and a rotation quaternion r for the orientation [139]. The full transformation can be applied to a vector x by first applying the rotation and then the translation. We use from now on whenever it is not ambiguous the following abbreviated notation:

$$\begin{aligned} x' &= (r \otimes x \otimes r^*) \oplus t \\ &:= rxr^* + t \end{aligned}$$

Using Euler angles α, β, γ for the rotation [47], the same equation reads

$$x' = \begin{bmatrix} R(\alpha,\beta,\gamma) & t \\ 0 & 0 \end{bmatrix} x = R(\alpha,\beta,\gamma)x + t \tag{4.5}$$

Both representations are valid and lead in first order approximation to the same result [107].

Choice of Error Representation

Since we assume the measurement is affected by some error, we need to add the rotational error Δ_r and the positional error Δ_t to the equation. We assume that the error (Δ_r, Δ_t) is normally distributed with a covariance matrix $\Sigma \in \mathbb{R}^{6\times 6}$ and a mean of 0.:

$$(\Delta_r, \Delta_t) \sim N(\Sigma, 0) \tag{4.6}$$

For demonstration we visualize this concept in $\mathbb{R}^2$ (figure 4.1). The translation (green dots) is distributed around a mean value and the new orientation (green lines) has also a gaussian distribution around the mean rotation.

Now we have basically four different possibilities in which order to apply the error in the transformation:

$$\begin{aligned}
(1) \quad x' &= \Delta_r(rxr^*)\Delta_r^* + t + \Delta_t \\
(2) \quad x' &= r(\Delta_r x \Delta_r^*)r^* + \Delta_t + t \\
(3) \quad x' &= \Delta_r(rxr^* + t)\Delta_r^* + \Delta_t \\
&= \Delta_r(rxr^*)\Delta_r^* + \Delta_r t \Delta_r^* + \Delta_t \\
(4) \quad x' &= r(\Delta_r x \Delta_r^* + \Delta_t)r^* + t \\
&= r(\Delta_r x \Delta_r^*)r^* + r\Delta_t r^* + t
\end{aligned}$$

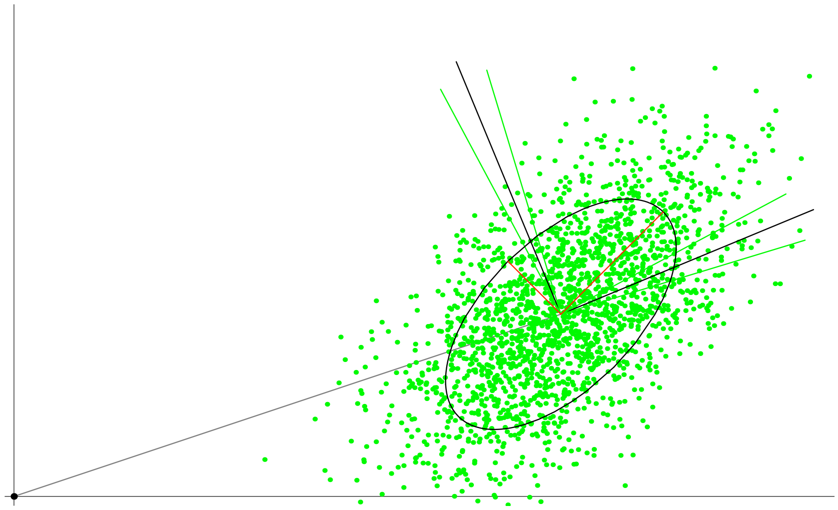


Fig. 4.1. Rigid body transformation in $\mathbb{R}^2$ with Gaussian distribution

Using homogeneous matrices, the same equations read:

$$
\begin{aligned}
(1)\quad x' &= (\Delta_R\, R)x + t + \Delta_t \\
(2)\quad x' &= (R\, \Delta_R)x + \Delta_t + t \\
(3)\quad x' &= (\Delta_R\, R)x + (\Delta_R t) + \Delta_t \\
&= \begin{bmatrix} \Delta_R & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} x \\
(4)\quad x' &= (R\, \Delta_R)x + (R\Delta_t) + t \\
&= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_R & \Delta_t \\ 0 & 1 \end{bmatrix} x
\end{aligned}
$$

Note that due to the missing commutativity of the quaternion multiplication or matrix multiplication, there is a difference between first applying the rotational error and then the actual rotation, and the other way around. The reason for this difference is the reference coordinate frame in which the error is expressed. Equations (3) and (4) can be decomposed into two consecutive rigid transformations, one with the actual transformation and another one with the transformation error. Figure 4.4 shows an explanation of the concept.

For all further analysis we choose equation (4), which implies that both the rotational and the translational error are expressed in already transformed coordinates, but also (2) has been used [139].

We denote a transformation with some error by the tupel (T, Σ), where T is the expected value of the transformation and Σ the covariance matrix of the error Δ_T with

$$\Delta_T \sim N(0, \Sigma)$$

Likewise, a transformation $T = [R|t]$ consists of a rotation R and a translation t with corresponding errors Δ_R and Δ_t. For the error Δ_R we use sometimes the matrix representation, but for the covariance we need a representation with less redundancy, therefore the fixed Euler angle representation is used.

$$\Delta_r = Eu(\Delta_R)$$

For Euler angles, we assume that the rotational error $(\Delta_\alpha, \Delta_\beta, \Delta_\gamma)$ is distributed normally around the mean of $(0,0,0)$ (no rotation), for the quaternion we assume also normal distribution of 'almost-unit' quaternions $(\Delta_{q_x}, \Delta_{q_y}, \Delta_{q_z}, 1)$ around $(0,0,0,1)$ [139]

Using this error representation, the covariance matrix Σ is a six-dimensional covariance matrix with three parameters for the translation and another three parameters for the rotation expressed in Euler angles.

4.2.2 Concatenation

Concatenation of two spatial measurements is a common procedure in computer graphics as well as in tracking networks [85]. The error propagation under concatenation has been analyzed by West et.al. [217] for isotropic root mean square errors and by Coelho et.al [39] using the unscented transform. In our work we use standard linear error propagation.

In the spatial relationship graph of figure 4.2, concatenation means we construct a new edge $T_c = [R_c|t_c]$ from A to C by combining the edges $T_1 = [R_1|t_1]$ from A to B and $T_2 = [R_2|t_2]$ from B to C.

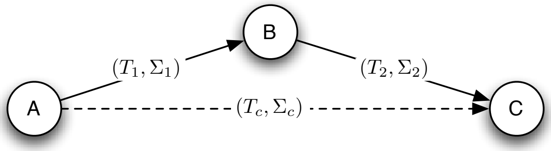


Fig. 4.2. Concatenation of two spatial transformations (T_1, Σ_1) and (T_2, Σ_2) into the combined transformation (T_c, Σ_c)

In homogeneous matrix representation, concatenation is simply defined by matrix multiplication

$$\begin{aligned} x' &= [R_1|t_1][\Delta_{R_1}|\Delta_{t_1}]\ [R_2|t_2][\Delta_{R_2}|\Delta_{t_2}]x \\ &= [R_c|t_c][\Delta_{R_c}|\Delta_{t_c}]x \end{aligned}$$

with

$$
\begin{aligned}
t_c &= R_1 t_2 + t_1 \\
R_c &= R_1 R_2 \\
\Delta_{t_c} &= R_1(\Delta_{R_1}(R_2 \Delta_{t_2}) + \Delta_{t_1} - t_2) \\
\Delta_{R_c} &= R_2^T \Delta_{R_1} R_2 \Delta_{R_2}
\end{aligned}
$$

The magnitude of the new translational error Δ_{t_c} depends now not anymore only on the error of (T_2, Σ_2) and the translational error of (T_1, Σ_1), but additionally on the rotational error of (T_1, Σ_1), proportional to the distance t_2. The rotational error on the other hand consists in the combination of both rotational errors where Δ_{R_1} first gets rotated into the coordinate system of T_2.

From this we build again the derivative of $(\Delta_{t_c}, \Delta_{R_c})$ with respect to Δ_{t_1}, Δ_{t_2}, $\Delta_{r_1} = Eu(\Delta_{R_1})$, and $\Delta_{r_2} = Eu(\Delta_{R_2})$ evaluated at 0 to get the error propagation formula for Σ_c. This results in rather huge equations [139] which are not easy to handle. Therefore we are going to analyze first some simplified special cases, which we will then use to get a generalized analysis for arbitrary transformations.

Coordinate System Change

We can perform a coordinate system change to a transformation (T_2, Σ_2) by applying an error-free transformation $(T_1, 0)$ to the transformation (T_2, Σ_2). Figure 4.3 shows the concept of this coordinate system change.

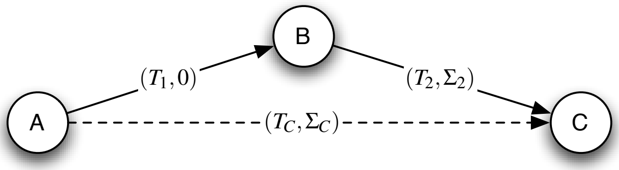


Fig. 4.3. Changing coordinate system by applying an error-free transformation $(T_1, 0)$ to a transformation (T_2, Σ_2)

We will show that in this case the rotational error does not change while the translational error is rotated into the new coordinate system.

A transformation with errors (T_2, Σ_2) is transformed into a new coordinate system by an error-free transformation T_1. This results into the following equation for the resulting transformation:

$$
\begin{aligned}
x' &= T_1 T_2 \Delta_{T_2} x \\
&= R_1(R_2(\Delta_{R_2} x + \Delta_{t_2}) + t_2) + t_1
\end{aligned}
$$

which can be rewrittten as

$$R_c(\Delta_{R_c} x + \Delta_{t_c}) + t_c$$

with

$$\begin{aligned} t_c &= R_1 t_2 + t_1 \\ R_c &= R_1 R_2 \\ \Delta_{t_c} &= R_1 \Delta_{t_2} \\ \Delta_{R_c} &= \Delta_{R_2} \end{aligned}$$

To get the error propagation for the random variable $\Delta_{T_2} = [\Delta_{R_2} | \Delta_{t_2}]$ we need to build the derivative of $(\Delta_{t_c}, \Delta_{R_c})$ with respect to the error Δ_{t_2} and $\Delta_{r_2} = Eu(\Delta_{R_2})$, evaluated at 0.

$$\frac{\delta}{\delta(\Delta_{t_2}, \Delta_{r_2})} \begin{bmatrix} R_1 \Delta_{t_2} \\ Eu(\Delta_{R_2}) \end{bmatrix} \bigg|_{\Delta_{t_2} = Eu(\Delta_{R_2}) = 0} = \begin{bmatrix} R_1 & 0 \\ 0 & I \end{bmatrix} \tag{4.7}$$

So the new covariance matrix Σ_c is obtained by

$$\Sigma_c = \begin{bmatrix} R_1 & 0 \\ 0 & I \end{bmatrix} \Sigma_2 \begin{bmatrix} R_1 & 0 \\ 0 & I \end{bmatrix}^T$$

The translational part of the covariance matrix is rotated by a similarity transform into the new coordinate system, while the rotational part of the covariance stays unchanged [85]. In the case of three-dimensional positional covariances (cf. section 5.3.3), we can also write

$$\Sigma_c = R_1 \Sigma_2 R_1^T$$

by using only the upper left part of the Jacobian matrix from equation 4.7. We can use this error propagation to derive formulas for the combination of two transformations with error

$$(T_1, 0) \circ (I, \Sigma_2) = (T_1, R_1 \Sigma_2 R_1^T) \tag{4.8}$$

$$(T_1, 0) \circ (T_2, \Sigma_2) = (T_1 T_2, R_1 \Sigma_2 R_1^T) \tag{4.9}$$

or vice versa for decomposing a transformation with error into a pure transformation and the pure error:

$$(T, \Sigma) = (T, 0) \circ (I, R^T \Sigma R) \tag{4.10}$$

This will enable us later to analyze the error propagation under the assumption of an identity transformation affected by the error.

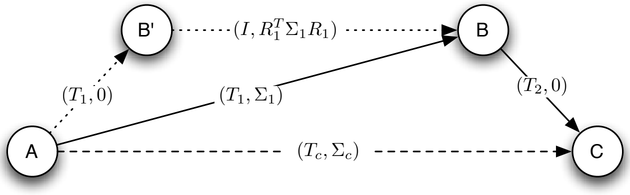


Fig. 4.4. Propagation of an error-free transformation $(T_2, 0)$ through a transformation (T_1, Σ_1). The decomposition of equation 4.10 is shown with dotted lines.

Propagation of an Error-free Transformation

Now we assume that $(T_2, 0)$ is a pure transformation with no error and (T_1, Σ_1) contains error.

Again the equations for the resulting translational and rotational error will be much simpler:

$$\begin{aligned} x' &= [R_1|t_1][\Delta_{R_1}|\Delta_{t_1}]\ [R_2|t_2]x \\ &= [R_c|t_c][\Delta_{R_c}|\Delta_{t_c}]x \end{aligned}$$

with

$$\begin{aligned} t_c &= R_1 t_2 + t_1 \\ R_c &= R_1 R_2 \\ \Delta_{t_c} &= R_1(\Delta_{R_1} t_2) + \Delta_{t_1} \\ \Delta_{R_c} &= R_2^T \Delta_{R_1} R_2 \end{aligned}$$

Without loss of generality we assume additionally that $T_1 = I$. We can always use the decomposition of equation 4.10 to accomplish this. The equation then reads

$$\begin{aligned} \Delta_{t_c} &= \Delta_{R_1} t_2 + \Delta_{t_1} \\ \Delta_{R_c} &= R_2^T \Delta_{R_1} R_2 \end{aligned}$$

We can now compute the derivative

$$\frac{\delta}{\delta(\Delta_{t_1}, \Delta_{r_1})} \begin{bmatrix} \Delta_{t_c} \\ \Delta_{r_c} \end{bmatrix} \Bigg|_{\Delta_{t_1} = \Delta_{r_1} = 0} = \begin{bmatrix} 1 & & & & z_{t_2} & -y_{t_2} \\ & 1 & & -z_{t_2} & & x_{t_2} \\ & & 1 & y_{t_2} & -x_{t_2} & \\ & & & & & \\ & & & & R_2^T & \\ & & & & & \end{bmatrix}$$

Since we are going to need this Jacobian a few more times, we define

$$J_T := J_{[R|t]} = \begin{bmatrix} I & Skew(t) \\ 0 & R^T \end{bmatrix} \tag{4.11}$$

where $Skew(t)$ is the skew-symetric matrix

$$Skew(t) = Skew\big((x_t, y_t, z_t,)^T\big) = \begin{bmatrix} & z_t & -y_t \\ -z_t & & x_t \\ y_t & -x_t & \end{bmatrix}$$

So the resulting error Σ_c of a identity transformation with error (I, Σ_1) by a pure transformation $(T_2, 0)$ is given by

$$\Sigma_c = J_{t_2} \Sigma_2 J_{t_2}^T$$

and therefore

$$(I, \Sigma_1) \circ (T_2, 0) = (T_2, J_{T_2} \Sigma_1 J_{T_2}^T) \tag{4.12}$$

Concatenation of Two Transformations with Error

Using the results of the previous sections, we can now derive a simpler term for the combined transformation with error (T_c, Σ_c) as a concatenation of the transformation (T_1, Σ_1) and (T_2, Σ_2) as in figure 4.2.

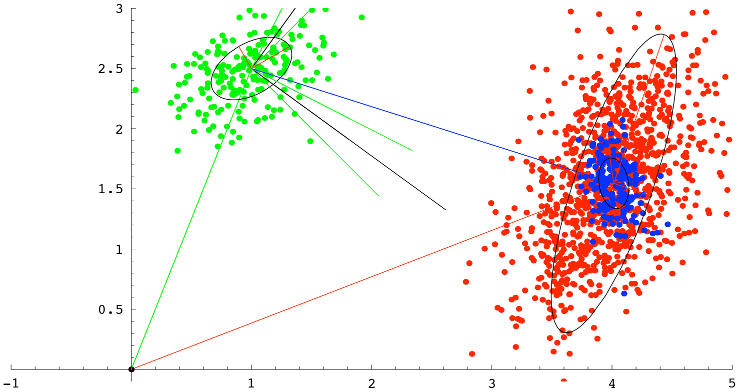


Fig. 4.5. Error propagation for the concatenation (red dots) of two rigid transformations (green and blue dots) with some error. For small rotational errors the linear error propagation fits with the actual error from a Monte-Carlo simulation

We apply the error propagation for general rigid transformations to the covariance matrix Σ_1 to get the propagated covariance in the same coordinate system as Σ_2. After combining them to a joint covariance matrix (cf. section 3.4.3 and [85]) we retransform them back to the original coordinate frame.

$$\begin{aligned}
(T_c, \Sigma_c) &= (T_1, \Sigma_1) \circ (T_2, \Sigma_2) \\
&\overset{(4.10)}{=} (T_1, 0) \circ (I, R_1^T \Sigma_1 R_1) \circ (T_2, 0) \circ (I, R_2^T \Sigma_2 R_2) \\
&\overset{(4.12)}{=} (T_1, 0) \circ (T_2, J_{T_2} R_1^T \Sigma_1 R_1 J_{T_2}^T) \circ (I, R_2^T \Sigma_2 R_2) \\
&\overset{(4.10)}{=} (T_1, 0) \circ (T_2, 0) \circ (I, R_2^T J_{T_2} R_1^T \Sigma_1 R_1 J_{T_2}^T R_2) \circ (I, R_2^T \Sigma_2 R_2) \\
&\overset{(3.8)}{=} (T_1 T_2, 0) \circ (I, R_2^T (J_{T_2} R_1^T \Sigma_1 R_1 J_{T_2}^T + \Sigma_2) R_2) \\
&\overset{(4.8)}{=} (T_1 T_2, R_1 (J_{T_2} R_1^T \Sigma_1 R_1 J_{T_2}^T + \Sigma_2) R_1^T) \qquad (4.13)
\end{aligned}$$

Note that applying a rotation to a six-dimensional covariance matrix in this case means applying the rotation to the positional part of the covariance only (cf. equation 4.7).

Similar results can be derived using quaternion notation [139] or using absolute bounding errors [85].

Nonlinear Error Propagation

The first-order error propagation using linear forward propagation with Jacobi matrices is only an approximation of the actual error propagation. To analyze the effect of this linearization, we use a Monte-Carlo experiment for the concatenation of two transformations. Figure 4.6 shows the result of this simulation for two transformations in $\mathbb{R}^2$ with rather large angular error compared to the positional error.

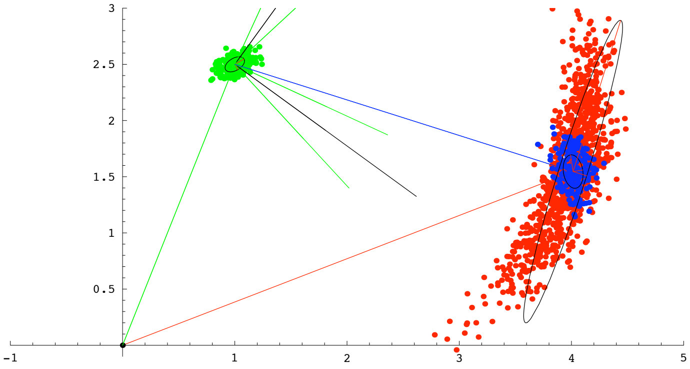


Fig. 4.6. Error propagation for the concatenation of two rigid transformations (green and blue dots) with error. The Monte-Carlo simulation of the concatenation (red dots) shows a 'banana shape' in the distribution for large angular errors.

The error distribution of the red points was computed using a Monte-Carlo simulation. It is obvious that the linear error propagation can not cover this error correctly anymore. This effect gets bigger for larger angular errors in Σ_1. Using a second order approximation, for example the unscented transform (cf. section 3.4.5 and figure 4.7) could reduce this problem. However, in many setups the actual angular uncertainty is not too large, so linear error propagation is a valid choice.

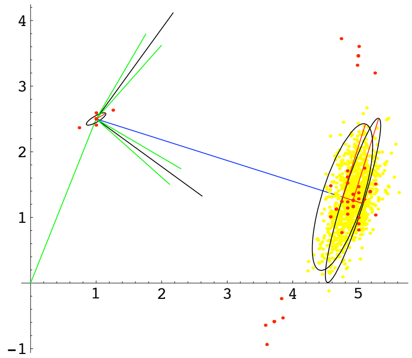


Fig. 4.7. Linear error propagation (yellow) compared with second order propagation using the unscented transform (sigma points shown in red). For large angular errors the difference in the two propagations is noticable.

Especially in augmented reality scenarios, large angular errors and their propagation would lead to tremendous alignment errors and would make the system unusable even before the nonlinearity effects become significant.

4.2.3 Inversion

In a similar way we can now develop the error propagation formula for the inversion of an edge. Given a transformation $T = [R|t]$ with the corresponding covariance matrix Σ, the inverse of the full transformation with error is given by

$$x' = \left(\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_R & \Delta_t \\ 0 & 1 \end{bmatrix} \right)^{-1} x \quad = \begin{bmatrix} \Delta_R & \Delta_t \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} x \qquad (4.14)$$

This shows that the inversion is nothing else than a propagation with a transformation $(T^{-1}, 0)$ of the inverse of the error. We can apply the decomposition of equation 4.10 to get

$$(T^{-1}, \Sigma_I) = (T^{-1}, 0) \circ (I, R\Sigma_I R^T)$$

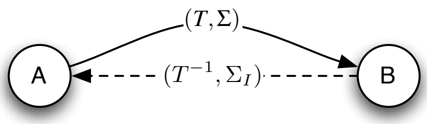


Fig. 4.8. Inversion of a transformation (T, Σ) results in a transformation (T^{-1}, Σ_I)

While we do have the propagation formula for the concatenation with the pure transform already, we still need to analyze the error propagation in case of the inversion.

The corresponding equation is given by

$$\begin{aligned} x' &= [\Delta_R | \Delta_t]^{-1} x \\ &= [\Delta_R^T | - \Delta_R \Delta_t] x \\ &= R_I(\Delta_{R_I} x + \Delta_{t_I}) + t_I \end{aligned}$$

with

$$\begin{aligned} t_I &= 0 \\ R_I &= I \\ \Delta_{t_I} &= -\Delta_R \Delta_t \\ \Delta_{R_I} &= \Delta_R^T \end{aligned}$$

The derivative of $(\Delta_{t_I}, \Delta_{R_I})$ with respect to Δ_{t_I}, Δ_{R_I} evaluated at 0 is the negative identity matrix, so applying the error propagation formula does not change the covariance.

Going back to the original equation 4.14 we see that we can ignore the inversion of the error for the propagation and apply the error propagation (cf. section 4.2.2.) for the inverse transformation $T^{-1} = [R^T | - R^T t]$. This leads to a new covariance matrix Σ_I with

$$\Sigma_I = J_{(-R^T t)} R \Sigma R^T J_{(-R^T t)}^T \tag{4.15}$$

Again, the rotation is only applied to the positional part of the covariance matrix here (cf. equation 4.7).

Figure 4.9 shows a visualization of this equation in $\mathbb{R}^2$. The transformation T from A to B is inverted, and the corresponding covariance Σ_I of the inverted edge is computed using the error propagation formula.

Similar results can be derived using quaternion notation [139]. Also, non-linear error propagation using the unscented transform (section 3.4.5) could be employed.

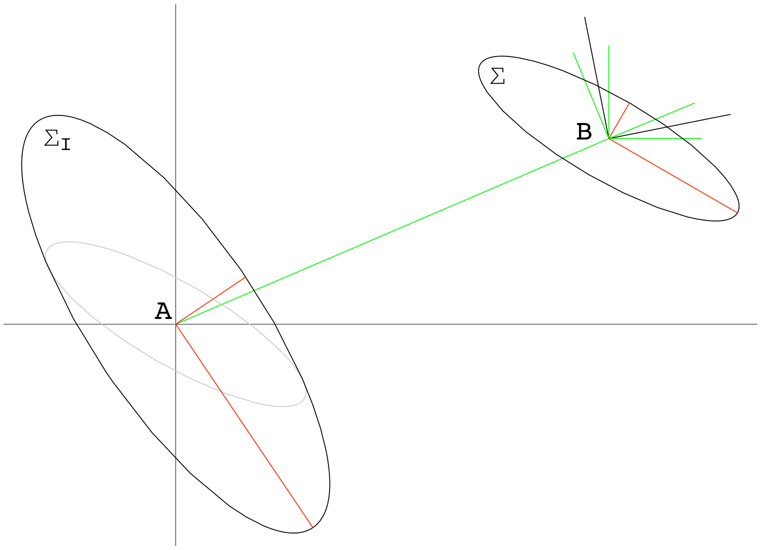


Fig. 4.9. Example error propagation for the inversion of a transformation. The transformation $(T_{\mathsf{AB}}, \Sigma)$ from A to B with error Σ is inverted to $(T_{\mathsf{BA}}, \Sigma_I)$.

5

Statistical Models for the Accuracy of Optical Tracking Systems

Every physical measurement is affected by errors. There are many different kinds of errors, which we are going to present in the following section. We then show methods and guidelines on how to estimate the errors in a tracking system. After that we go into details of a specific class of tracking systems: We show how error estimation can be performed for both n-ocular and monocular optical tracking systems.

5.1 Classification of Errors

Many different sources of errors contribute to the final error of a tracking system. Some of the errors presented here are specific for optical tracking systems, while others apply to different tracking systems as well.

The most important distinction is between dynamic and static errors [12]. We also distinguish between measurement errors that cause only wrong position measurements, and detection errors when the whole object can not be detected, due to occlusions or misidentification. Finally we have to differentiate between actual random errors and systematic errors, usually resulting from a wrong model of the measurement system. We are also going to show how these errors can be all treated as random errors for simplification.

5.1.1 Dynamic Errors

Dynamic errors are errors that are only present when the objects in the scene are moving [13]. The reason for this is the end-to-end delay [85] from the time of the measurement until the application displays or uses the data.

First-order dynamic error [139] is the registration error caused by the motion of the tracked object and the end-to-end system delay and can be estimated by the product of the motion speed and the delay. Other dynamic errors are introduced in intermediate steps when several unsynchronized measurements are used together.

In both cases, the sampling rate of the sensor is of importance. Longer intervals between measurements lead to larger errors [3] since the measurements tend to be outdated when they are used. Additionally, according to the sampling theorem, motions with high frequency can not be detected by slow measurements [139]. It is important to note that dynamic errors usually are much larger than static errors [205] in many setups.

There are several possible solutions to reduce the influence of dynamic errors besides the obvious way of increasing the sampling rate and speed of the measurements. To reduce the synchronization delay, we can physically assure that all the measurements are made at the same time. In particular this means for video see-through augmented reality to synchronize the tracking system with the video camera [170] and therefore delaying the whole display process. Using this trick, the tracking system measurements always fits to the displayed image, and small delays in the whole image in the head-mounted display are much more tolerable than discrepancies in the overlay.

Another way of synchronizing measurements is the use of predictive filters [9] and an appropriate motion model for the measurement data. This process needs accurate synchronized clocks on all machines to get useful estimates from the filters. The success of this method highly depends on the choice of a good motion model and therefore is hard to apply in a general context. Predictive filters can be also used to estimate the current error [3, 28] of a measurement at any time later than the measurement.

In the further context of this thesis however we are only interested in the static measurement error. We want to estimate the static error at the time of the measurement. This error can then additionally be processed through predictive models to get estimates for other points in time.

5.1.2 Identification and Visibility

Detecting and identifying features in the camera image is an important subtask for optical tracking systems. Also in this step, errors can occur. Features can be hidden due to occlusions from other objects, or parts of an object can be misidentified as a feature point.

Usually the pose reconstruction is performed using a set of correspondences between points on the two-dimensional image plane of the camera and three-dimensional feature points on the tracking target. The pose estimation problem is harder when the corresponding points are not known [46]. Identification of corresponding points can be established by uniquely identifiable patterns [102] (see section 5.4.1) at the fiducials, or by analyzing the marker constellation, a technique common with passive infrared systems [152, 167, 77]. Active infrared systems have the additional advantage that they can use time-division multiplexing to identify the fiducials [80].

It is important to design marker targets in a way such that they provide not only optimal accuracy but also optimal visibility. This has an influence on the marker design [48, 49] and on the camera placement [37, 36, 31]. In particular the case of *merging markers*, this means fiducials that are so close together such that they cannot be distinguished, needs special consideration (see section 5.3.2). Some fiducials can be only detected under a certain viewing angle. This is not only true for printed flat markers (see section 5.4.1) but also for active markers that show a significant anisotropic behavior in the center detection for different viewing angles [92] and therefore are only used for the pose estimation when the viewing angle is in a certain range, although they could be detected otherwise.

In our error analysis we assume that all used features are correctly detected and already identified. Although we can not generally assume that this will be actually the case, we have at runtime inside the tracking system all this information. Therefore we can perform the analysis with all actually used features instead of all possible features.

5.1.3 Systematic Errors

Experience shows that many of the measurement errors in optical tracking systems show a significant systematic behavior. Besides the occlusion and identification problems mentioned above, there are many different sources of error here.

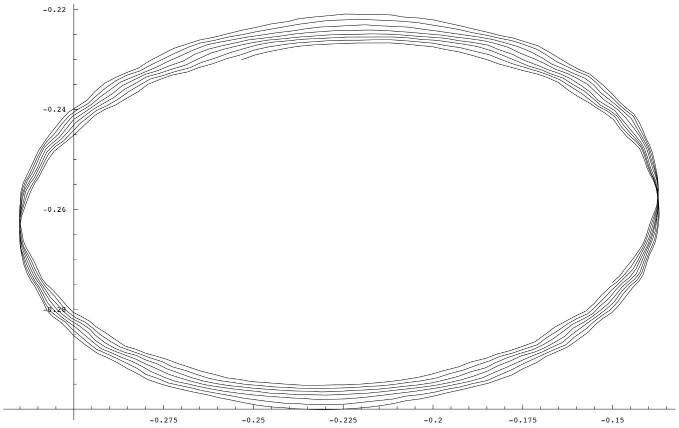


Fig. 5.1. Position of a single fiducial rotating on a helical path on the camera image plane. Systematic errors in the two-dimensional feature detection will result in interference patterns in the three-dimensional pose estimation in a two-camera setup.

The most important influence on the accuracy in a multiple camera system comes from the epipolar geometry. This relationship between the different cameras has to be estimated initially. We have observed significant errors (cf. figure 5.11) resulting from a wrong calibration of the extrinsic camera parameters in our experiments [154, 183]. Thermal drift due to the heating of the cameras [183] as well as vibrations in the mounting of the cameras also changes the extrinsic camera parameters. Some of these influences can be minimized by using only measurements relative to a reference target.

For every camera, the intrinsic camera parameters have to be determined as well [184] for the pose estimation. The accuracy of the intrinsic camera parameters [72, 209] can have a large systematic influence on the overall accuracy [162], since this is usually calibrated once and used thereafter. Also residual errors in the nonlinear camera model [110, 109] can lead to systematic errors [61].

A last source of systematic errors potentially comes from the algorithms used in the overall process (see figure 5.1) as well as from the quantization on the camera chip (cf. figure 5.6)

5.1.4 Random Noise

In addition to the systematic errors described above, there is also random noise resulting from two-dimensional marker jitter [198] in the cameras. But our experiments (cf. figure 5.5) show that also the systematic errors are coming from so many different sources that under certain circumstances they will result in quasi-random error that can be approximated using a gaussian distribution.

5.2 Measuring the Error

Measuring the error in augmented reality systems is not an easy task. To get absolute error numbers, we need to compare the measurements with some kind of ground truth. Alternatively, we can use simulations with a known ground truth to estimate the accuracy. The last option is to provide an analytical function to compute the current accuracy. To get an international comparable way of determining and expressing uncertainties in measurements, the *International Standards Organisation (ISO)* has published a *Guide to the Expression of Uncertainty in Measurement (GUM)* [89, 88].

5.2.1 Empirical Evaluations

Empirical evaluations are the best option to measure the accuracy of a tracking system. This means evaluating the uncertainty by the statistical analysis of a series of observations. In this case the uncertainty is the experimental standard deviation of the mean that follows from an averaging procedure or an appropriate regression analysis [89]. Therefore we have to make sure that the measurement process itself, that is used as a ground truth, has significantly less error than the process to be measured [146].

The uncertainty is most commonly given as an average RMS error inside the whole working volume. This is also well accepted and regularly performed by tracking system manufacturers [223, 222]. Other empirical evaluations show that the errors have in fact large differences inside the tracking volume [62]

The standard process [191] recommends the measurement of relative distances rather than absolute values when no reliable ground truth is available.

5.2.2 Monte-Carlo Simulations

If no ground truth data is available for a system, it is sometimes possible to generate artificial input data for predefined parameters with some statistical variation to compare the result with the original values. This kind of Monte-Carlo simulation has been used frequently. Florou et.al. [61] use a Monte-Carlo simulation to estimate the quality of camera calibration and resulting position accuracy. Weyrich et.al. [220] and Pentenrieder et.al. [134] use computer-generated images to test an optical tracking system.

The main difficulty here is the need for an inverse function of the process to estimate. In the case of an optical tracking system, a renderer has to be written that generates input images. This inverse function has to include all kinds of errors that will occur in reality to be a good model of the actual runtime system [219].

Additionally, Monte-Carlo simulations have the property that they need large amounts of simulation runs for higher-dimensional input vectors to give statistically significant results. Even simple models of optical tracking systems have already quite a lot of input parameters. This is usually no problem, since the simulation can be done offline. However, runtime estimates of the current accuracy are not possible using this model.

5.2.3 Analytical Models

The advantage of analytical models for accuracy estimates to compute error estimates at runtime for the actual current situation. An analytical error model is a function of a defined set of input parameters that returns some measure for the accuracy of the particular configuration of the system given by the parameters. Again, this is only a simplification of the actual process, therefore the validity of the error function has to be proved using either empirical measurements or simulations first.

Abawi et.al. [1] provide an accuracy function for the *ARToolkit* which they have extracted from a series of empirical measurements. They return an abstract value from *good* to *bad* depending on the viewing angle and distance of the camera to the marker. Pentenrieder et.al. [134] propose an accuracy function for a monocular flat marker tracker based on a database of simulation results. They use a six-dimensional error covariance matrix that can be used for error propagation as in chapter 4.2. We will give in section 5.3 and section 5.4 an analytical error function based on the geometry of the cameras and the fiducial layout. This error function is applicable for a variety of optical tracking systems

A similar method of estimating the accuracy based on the geometric constellation only is used in satellite positioning systems [149]. Here the pose estimation is based on range measurements from the satellite sender to the mobile receiver. The receiver estimates from the difference in the received time-stamp signals both the exact local time and the three-dimensional position in space. The *dilution of precision* [105] denotes the expected accuracy of the pose and time estimation based on the satellite constellation in space.

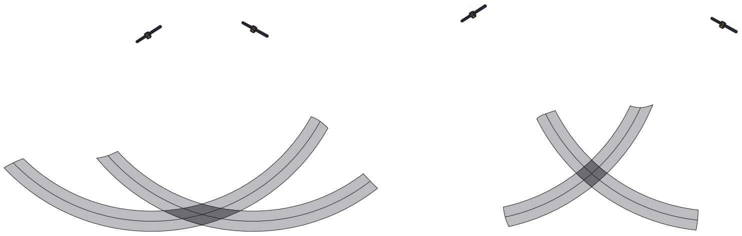


Fig. 5.2. *Dilution of Precision* in range measurements for satellite positioning systems: A small baseline gives bad dilution of precision (left) while a large baseline gives good dilution of precision (right)

The range measurements are affected by errors coming from a variety of different sources, mainly atmospheric refraction and different signal runtime in different layers of the atmosphere. Although mathematically incorrect [225, 226], it is assumed that the wrong measurements in fact form a statistically gaussian distributed deviation of the pseudo-range measurements. This statistical deviation is called the *user equivalent range error* and is estimated globally for every satellite by the control segment and transmitted from the satellites to the receiver. Geometrically, the effect can be seen in figure 5.2.

From this error it is possible to compute a four-dimensional covariance matrix [116] for the three position variables and the single time variable. From this covariance matrix, four different values are computed by means of the matrix trace norm which is equivalent to a root mean square error (see 3.2.3). The *positional dilution of precision (PDOP)* is the trace norm of the positional part, which is the upper left 3×3 matrix. The *time dilution of precision (TDOP)* is the lower right diagonal element, the time variance. For the next two measures the covariance matrix is first transformed from the original world-fixed coordinate system into a local coordinate system at the estimated point (cf. section 4.2.2). Then the *horizontal dilution of precision (HDOP)* is the trace of the 2×2 horizontal part of the transformed covariance matrix, which is the expected horizontal root mean square error, and the *vertical dilution of precision (VDOP)* is the variance in the vertical direction.

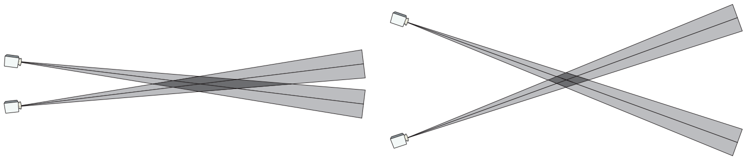


Fig. 5.3. *Dilution of Precision* for angular measurements in pose estimation using triangulation from two cameras. Again, large baseline (right) gives better dilution of precision than small baseline (left)

In a similar way, we compute the predicted accuracy of an optical tracking system based on covariances on the image plane and the geometric constellation of cameras and fiducials in chapter 5 (see figure 5.3).

The accuracy of an augmented reality system can be most easily defined in terms of the discrepancy of the real and the virtual objects. However, while a good alignment of the fiducials is desirable in a visual augmented reality application, it has been noted [60] that good fiducial alignment is in general a poor indicator of the overall accuracy. This means that the analysis of errors is highly application specific and can not be performed without the application in mind [113]. All general building blocks of an augmented reality system, such as tracking systems, calibration algorithms and other algorithms need to apply the error analysis in a general way such that the application can finally apply the general error to the actual application specific error. This is also the reason why we propose using full six-dimensional covariance matrix propagation instead of root-mean-square errors in section 6.4.

The overall error is the result of all individual errors in the tracking, calibration and projections and the propagation and combination of these errors to the error in the final augmentation.

5.3 *n*-Ocular Vision

An *n*-ocular optical tracking system usually contains of a set of *n* infrared cameras which are rigidly connected with each other, such that the epipolar geometry is constant. The object to be tracked is rigidly connected to a tracking target which contains several fiducials, which are retroreflective balls or infrared LEDs that can be seen by the infrared camera system. From the precalibrated extrinsic camera geometry the position of the fiducials in space and therefore the pose of the instrument can be computed [87]. A typical camera configuration is a rigid bar which contains one camera at each end, but more flexible camera arrangements are also possible.

The accuracy of n-ocular optical tracking systems has been analyzed already in the context of intraoperative navigation systems. In most common setups with two or more cameras the fiducial location error is not isotropic and independent. We present a refined estimation of the target registration error based on the propagation of covariances from the image plane up to the point of interest. We show that there are significant differences in the expected accuracy compared with the current state of the art.

In 1998, Fitzpatrick et.al. have presented a simple formula [60] for the estimation of the target registration error from a given fiducial location error. The fiducial location error is the error in detecting every single fiducial and is assumed to be unbiased, isotropic and independent. However, in real setups the error tends to be non-isotropic [218, 101] as we will also demonstrate in section 5.3.2

Hoff et.al. [83] have used arbitrary covariances for the fiducial location error to predict the marker target error. Our derivation in section 5.3.3 is based on their work. Ma et.al. [112] use spatial stiffness analysis to predict the registration error depending on the fiducial configuration. Davis et. al. [47] have also presented a method to predict the accuracy of a tracking target for optimized target design.

Sun et.al. [176] have introduced the concept of imaginary markers which they use to express the point of interest on which the target registration error is computed. From this they can obtain the accuracy distribution in a volume around the tracked rigid body. They use a Monte-Carlo simulation to estimate the resulting error at a point of interest given the error in the pose estimation of the target. Instead of randomized simulation we provide a closed form solution for this estimation in section 5.3.5.

Taking all together, we present a closed-form solution that estimates the target registration error only from the geometric setup of the cameras and the fiducials on the tracked tool and an estimate for the image plane error that can be derived from manufacturers data or estimated experimentally for each camera. The proposed approach provides more accurate results than the currently known methods.

5.3.1 Terminology of Errors

First we describe the terminology of errors we are using and relate them to the errors used in literature. In particular, we are looking at the following kinds of errors (cf. figure 5.4):

Image Plane Error (IPE)

The tracking algorithm detects the point features in the image plane. The accuracy of this feature detection is limited by factors such as the image noise

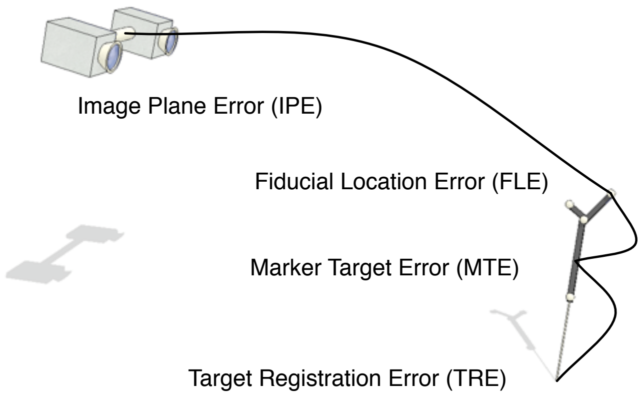


Image Plane Error (IPE)
Fiducial Location Error (FLE)
Marker Target Error (MTE)
Target Registration Error (TRE)

Fig. 5.4. Terminology of Errors involved in the estimation of the target registration error (TRE)

or the algorithm used and can be well approximated by a two-dimensional Gaussian error distribution, as we show later. Mistakes made in this step are propagated and amplified in the following steps and accumulate in the TRE. Also any other error made in the pose estimation process could be reprojected onto the image plane. Therefore we will show later that we can consider the IPE as the source of all the other errors we are dealing with.

Fiducial Location Error (FLE)

After the 2D locations of feature points in the image plane of two or more cameras are known, the 3D position of these features can be computed in the world. The way errors in the image plane propagate to the FLE is influenced by the position of the feature and the arrangement of the cameras. We will investigate this further in section 5.3.2. Allen et. al. [3] present a general framework for predicting the estimated performance of arbitrary tracking systems. Their work is similar to what we propose in section 5.3.2 and will be discussed in more detail there. Mitschke et. al. [122] showed that it is crucial to know about the general shape of the error covariances in a given camera setup at design time.

Fiducial Registration Error (FRE)

This error is the root-mean-square distance between the measured fiducial locations after registration and their theoretically known locations. It can be easily measured but provides only a one-dimensional estimate which is under certain assumptions related to the fiducial location error [60].

Marker Target Error (MTE)

In most cases, rigid arrangements of three or more fiducials are used to construct a marker target of which both position and orientation can be determined. In this case the FLE error distributions of the individual fiducials influence the error in the 6D pose, but also the way they are arranged.

Target Registration Error (TRE)

In most applications, the pose of the marker target is not used directly, but instead the position of some point of interest in the target coordinate frame is determined, such as the tip of a pointing device. By propagating the MTE errors in position and orientation of the marker target to this point of interest, the TRE error relevant for the application can be estimated.

By combining the error propagations of all four steps, we can give a correct estimate of the target registration error for every measurement at runtime. Accurate runtime estimates can only be given by the tracking system itself since information is needed on how many cameras and how many feature points were used for the particular measurement.

However, at design time, the single error propagation steps are useful on their own, when used with reasonable default distributions of the previous step, in order to design optimal marker targets for particular applications or to experiment with different camera setups.

For our work, we assume that the actual positions of features, fiducials or targets are computed independently and provided to the accuracy estimation. This makes the approach suitable as an add-on to existing trackers or, provided with hypothetical input, as an off-line analysis tool.

5.3.2 Estimating the FLE from IPE

The fiducial location error is the error between the true position of a fiducial and the measured position. This error is usually given by the tracking system manufacturer as a result of standardization procedures [191] or can be measured experimentally [101, 33, 62, 223] as an average error inside a specified tracking volume, or predicted using Monte-Carlo simulation [61]. Typically, this error is given as a one-dimensional root-mean-square error. Fitzpatrick et.al. [60] base their analysis on this kind of unbiased, isotropic, and independent errors. However, looking at the experimental evaluations in the literature [101, 33, 62, 223, 153] shows that the errors in common two-camera setup are actually not isotropic.

Sanders-Reed [150] shows the derivation of error propagation formulas for a two sensor setup using first order statistics and shows the validity of the model using Monte-Carlo simulations. Our work [19, 154] goes beyond this by providing a simple matrix equation that can be used to estimate the accuracy of a tracking system with any number of cameras. The Gaussian error propagation has been used for the accuracy of mechanical stereotactic localizing before [234].

Experimental Setup

To assess the fiducial location error for a specific tracking setup, we have conducted a set of experiments with a two-camera tracking system.

We placed retroreflective marker balls as features in a regular grid on a table in our tracking volume. Both the cameras and the markers on the table were fixed throughout the experiment. We captured the measured locations of the features in space for several minutes at 60 Hz resulting in a total of 71553 sample points for each feature. We used a rather small baseline of about 20 cm to emphasize the effects in the images, see figure 5.5.

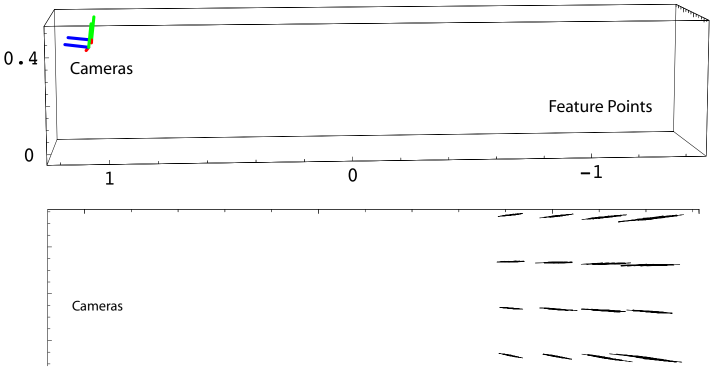


Fig. 5.5. Error in position estimation for a regular grid (side view and top view of the setup; error for visualization exaggerated by a factor 50)

The results of this experiment are shown in figure 5.6, where the measured 3D locations of a single feature point are plotted as black dots. We also calculated the covariance for each single feature point from this dataset, which is displayed as an ellipse at a 75% confidence level around the centroid. The particular regular pattern in the 3D reconstruction of the point results from discretization in the camera coordinate system. However, the general shape of the measurements in space is in fact approximated by the covariance ellipsoid.

To validate the result, we first map this covariance onto the image plane. In other words, we are searching for a covariance on the image plane of the camera that would result in this observed covariance at the position of the marker. To this end, we first need to develop a model of the camera equations, such that we can apply the error propagation from chapter 4.2 to these equations.

Pinhole Camera Model

The *pinhole camera model* [78] is a simplified model of the projection in a camera. The pinhole camera is a linear projection of a point x in space onto a point u on the image plane.

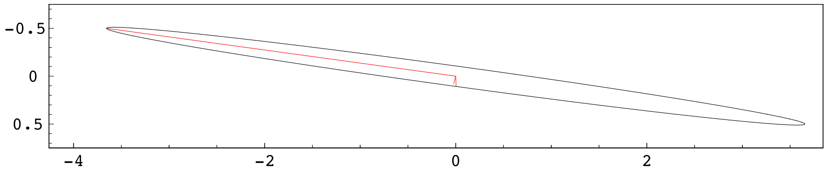


Fig. 5.6. Error in position estimation together with estimated covariance (95% confidence level) for a single marker from figure 5.5

$$p : \mathbb{R}^3 \to \mathbb{R}^2, \qquad x \mapsto \rho \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KT}x \tag{5.1}$$

where $\mathbf{T} \in \mathbb{R}^{4\times4}$ is a rigid transformation matrix that transforms the point p from world coordinates to camera coordinates. This transformation is also called the extrinsic camera matrix since it describes the position and orientation of the camera in space. The matrix $\mathbf{K} \in \mathbb{R}^{3\times4}$ is the intrinsic camera projection matrix. There are different models for the intrinsic camera matrix, describing parameters like the principle point offset, the skew and the pixel aspect ratio. For our further analysis however it does not matter which parameters are used inside the camera matrix.

The parameter ρ is the normalization factor, i.e. the inverse of the third row of the camera matrix equation. This maps all points located on the projection line through the camera center and the point x to the same point u.

Derivation of Covariance Formulas

We want to estimate the 3D covariance $\Sigma_{\mathbf{p}} \in \mathbb{R}^{3\times3}$ of the detection of a single feature (FLE) at position $\mathbf{p}$ in a given multi-camera setup, consisting in n pinhole cameras. We assume for now that the intrinsic and extrinsic camera parameters are known without error [162] and we have an estimate $\Sigma_{\mathbf{u}} \in \mathbb{R}^{2\times2}$ for the 2D IPE detection covariance including image noise, algorithm artifacts, and quantization errors on the image plane. For simplicity of the model we assume the same covariance in each camera (which is acceptable for commercial tracking systems).

If we use an n-ocular stereo system detecting the same point, we get the measurement function for the *Triangulation*, a set of nonlinear camera equations p:

$$\begin{aligned} p: \qquad \mathbf{u_1} &= \frac{1}{\rho_1}\mathbf{K_1 T_1 x} \\ \vdots \qquad & \qquad \vdots \\ \mathbf{u_n} &= \frac{1}{\rho_n}\mathbf{K_n T_n x} \end{aligned}$$

as the projection function, with $\mathbf{K_i}$ and $\mathbf{T_i}$ being the respective parameters of the i-th camera.

In order to compute the FLE, we build the Jacobian $J_p = \frac{\delta p}{\delta \mathbf{x}}$ and apply the backward propagation formula (3.6).

$$\Sigma_x = \left({J_p}^T \begin{bmatrix} \Sigma_{u_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Sigma_{u_n} \end{bmatrix}^{-1} J_p \right)^+ \tag{5.2}$$

The resulting equations are analytically computed using a computer algebra system and then evaluated for each position in space. Details can be found in Appendix A.1.

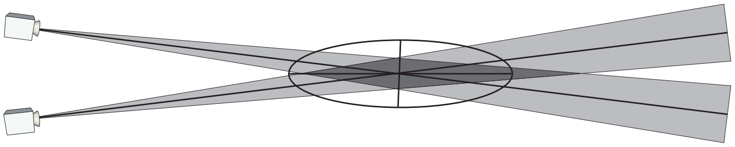


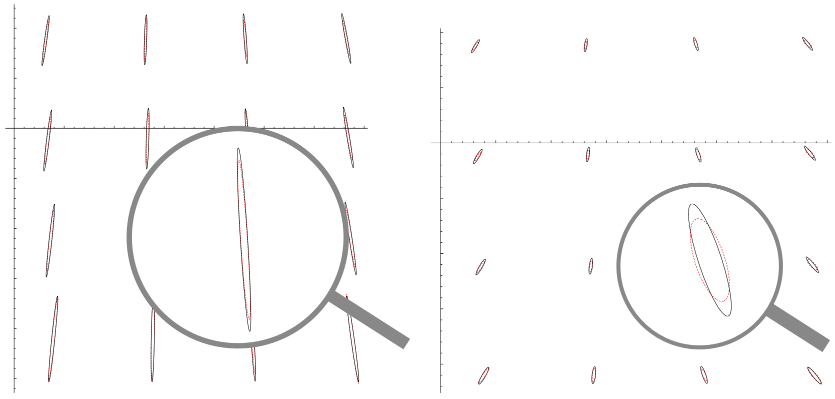
Fig. 5.7. The projected mean of the fiducial centers does not coincide exactly with the statistical mean for small baselines

From figure 5.7 it can be see that the projected mean of the fiducials on the image plane does not coincide exactly with the statistical mean of the estimated points for small baselines and large angular errors. Therefore the estimation of the gaussian error assumption is violated here [157, 158]. However, it can be shown using Monte-Carlo simulation that this difference is in fact negligible for common setups.

Estimation of the Image Plane Error

We now estimated the actual error covariance $\Sigma_{\mathbf{u}}$ on the image plane using the experimental measurements for a single feature point and applied the model to predict the parameters for the other fifteen points; we then compared the prediction with the measurements for these points.

To estimate the error in the image plane, we evaluated our model with symbolic parameters for the covariances on the image plane and used Newton's method to fit the parameters to the measurements. We assumed an unbiased Gaussian distributed error around the true two-dimensional marker locations [198]. Figure 5.8 shows the measured errors in red together with the computed errors in black first horizontally on the grid (x-z-plane) and then



x-*z*-plane *x*-*y*-plane

Fig. 5.8. Measured errors (red dotted line) vs. predicted errors (black line)

parallel to the image plane (*x*-*y*-plane, projected onto the grid). The parameter estimation was done for the lower leftmost point and then applied to all other points. The predicted values fit the measured values quite well, as can be seen in figure 5.8. In our experiment, we estimated a standard deviation of $\frac{1}{115}$ pixel on the image plane for the detection of the center of the fiducials.

Prediction for Common Setups

Using this image plane error, we can now apply equation 5.2 to estimate the fiducial location error for every single fiducial. Figure 5.9 shows the expected error in a four camera setup. The error is shown as covariance ellipsoids as well as RMS error. Figure 6.5 shows visualizations for several common setups with three and two cameras. The error covariances are magnified for better visibility.

In the same figure it is possible to see the effect of the occlusion of single cameras onto the fiducial location error. In the example, all configurations are subsets of a four camera setup. Therefore the effect of occluding a single camera or of occluding two cameras can be observed in the figure. Note the significant difference between large baseline and small baseline in the two camera examples.

Validation of the Model for Image Plane Error

To estimate the accuracy of optical coordinate measurement systems, VDI/VDE 2634/1 [191, 192] recommends measuring differences between single features in several directions [63]. We have implemented this measurement with a rotating two-ball target (cf. figure 5.10) for which we have measured the distance between the balls [19].

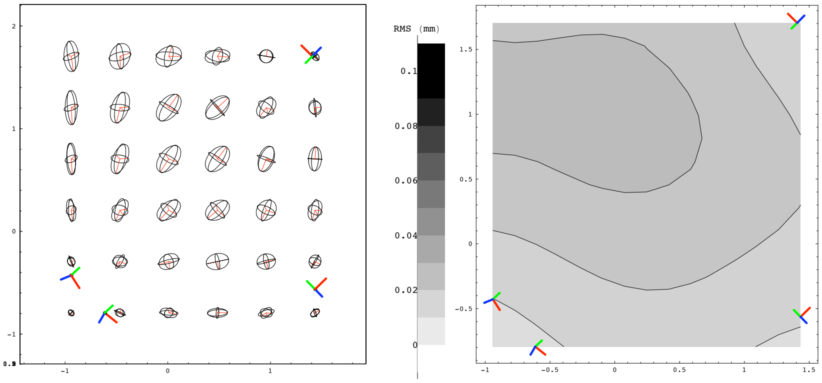


Fig. 5.9. Predicted accuracy in a four camera setup. Covariance ellipsoids (left) and RMS error (right). See also figure 6.5

Applying our theoretical error prediction model to this kind of test, we need to build the Jacobian J_d from the distance function,

$$J_d = \frac{\partial}{\partial(\mathbf{x}_1, \mathbf{x}_2)} \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)}$$

which we use to propagate the two 3D covariances $\Sigma_{\mathbf{x}_1}$ and $\Sigma_{\mathbf{x}_2}$ to a one-dimensional variance σ_d of the distance with

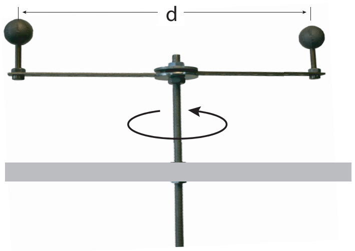


Fig. 5.10. Two rotating balls, rigidly connected

$$\sigma_d = {J_d}^T \begin{bmatrix} \Sigma_{p_1} & \mathbf{0} \\ \mathbf{0} & \Sigma_{p_2} \end{bmatrix} J_d$$

We compare the variance σ_d with our measurements, as shown in figure 5.11. The horizontal axis shows the angle of the two balls and the vertical axis the respective measured distance.

In the errors we first note a large sinusoidal error in the angular data. This error comes from a wrong scaling of the three room axes during the calibra-

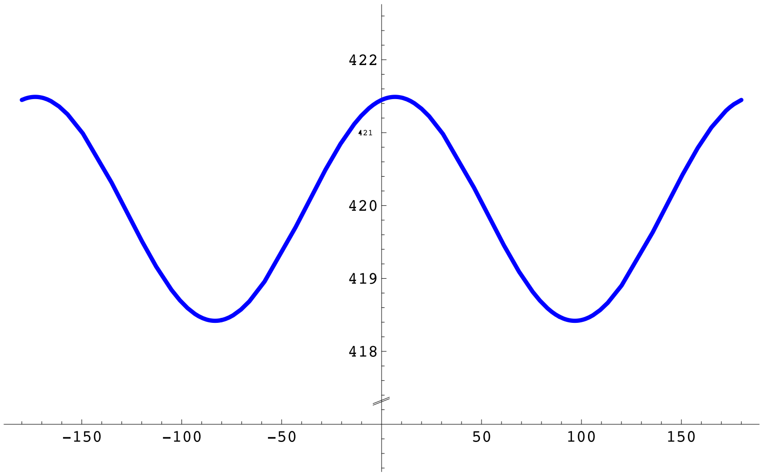


Fig. 5.11. Error in length estimation for a rotating two-ball target, blue line shows assumed room calibration error

tion of the system. Different calibration runs showed this effect in different intensity. This systematic error from the calibration process provided by the manufacturer of the tracking system needs to get eliminated independently, as it cannot well be modeled as Gaussian noise. For our analysis, we have removed that error manually from the data by assuming independent scaling on the three axes.

The remaining error consists of a random part coming from system noise as analyzed above, and an additional irregular error coming from subpixel effects in the 2D detection algorithms. Although these errors are in fact systematic, we assume for now that we can approximate them with a Gaussian distribution and model them as a zero-mean noise in the image plane. This is justified by the relatively high frequency of the artifacts.

Figure 5.12 shows a plot of the predicted one-dimensional standard deviation of the error covariance for the respective angle in green together with the distance measurements, already corrected for the wrong room calibration. While the corrected error obviously is not Gaussian, the predicted covariance fits the measurements still well.

This experiment shows that already for two fiducials in a single target the systematic errors begin to appear more and more random. Using typical targets with four or more fiducials, it is safe to assume that the resulting error is in fact gaussian distributed.

Additional Influences on the Image Plane Error

There are additional factors that influence the detection accuracy on the image plane [41]. The fiducial detection algorithms are more accurate when the fidu-

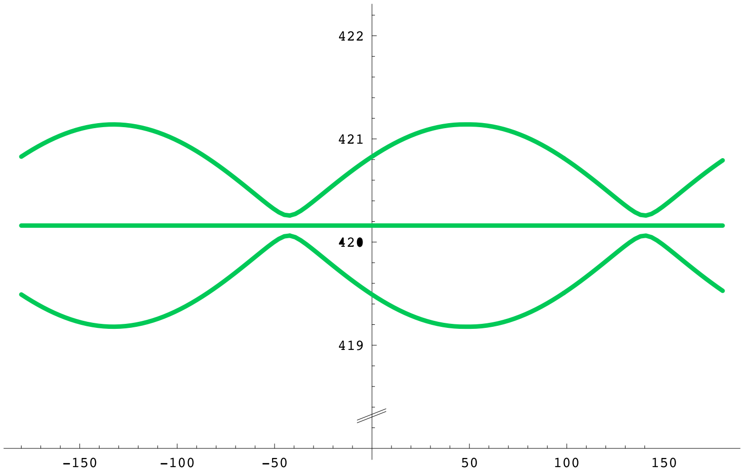


Fig. 5.12. Measured errors corrected for room scaling (blue dots) vs. predicted error standard deviation (green line)

cial fills a larger area on the image plane. This depends both on the distance from the camera to the fiducial as well as on the fiducial size.

Partial occlusions of fiducials by other fiducials or different objects as well as anisotropic lighting behavior of the fiducials lead to a shift in the estimated fiducial center. Finally, weak lighting conditions also influence the accuracy.

Distance and Marker Size

Different kinds of fiducials [38, 182] need different algorithms for the subpixel accurate determination of the fiducial location. Many systems use elliptic shapes [165, 20, 229, 59, 180] for fiducials, others use black square markers [173, 58, 24]. All these methods have the property that they get more accurate when the size of the fiducial on the image plane is larger. Table 5.3.2 shows the diameter in pixels of a circular fiducial depending on the diameter of the fiducial and the distance to the camera.

	Distance from Camera to Marker											
Size	0.25m	0.5m	0.75m	1.0m	1.25m	1.5m	1.75m	2.0m	2.25m	2.5m	2.75m	3.0m
12mm	22.6	11.3	7.5	5.6	4.5	3.7	3.2	2.8	2.5	2.2	2.0	1.8
14mm	26.4	13.2	8.8	6.6	5.2	4.4	3.7	3.3	2.9	2.6	2.4	2.2
20mm	37.8	18.9	12.6	9.4	7.5	6.3	5.4	4.7	4.2	3.7	3.4	3.1
30mm	56.7	28.3	18.9	14.1	11.3	9.4	8.1	7.0	6.3	5.6	5.1	4.7

Table 5.1. Relationship between marker ball diameter and distance to the camera and the resulting pixel-size diameter on the image plane. The focal-length is 3.5mm and chip-size is 4.87mm × 3.67mm for a resolution of 640 × 480 pixels.

It is obvious that in a certain distance from the camera the size differences due to the distance are becoming smaller. Additionally, the influence of a larger radius on the accuracy of the whole fiducial gets smaller with increasing size of the fiducial. Even using larger and larger fiducials, the achievable accuracy is limited.

Merging Markers and Partial Occlusions

One important situation to consider is the case of *merging markers.* This occurs when two markers are almost aligned in the camera image. The images of both markers merge together to a single fiducial on the image plane. The resulting estimate using standard algorithms like weighted averaging for the center of the fiducial is therefore biased into the direction of the merged fiducial as seen in figure 5.13. This case can be detected (but not necessarily corrected) using algorithms that determine the roundness of the fiducial. After the pose estimation is finished, a backprojection of the fiducial locations onto the image plane can be used as well to detect potential merging markers. Systems using uniquely identifiable fiducials are not affected by this problem.

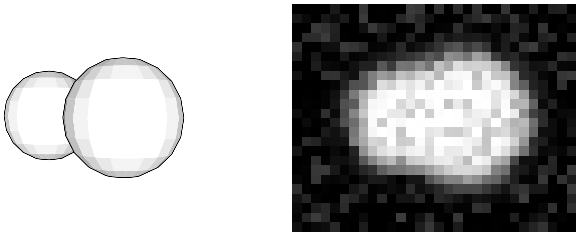


Fig. 5.13. Merging markers. Ellipse center detection in this case results in a deviation of the estimated fiducial center.

Fiducials can also be partially occluded by other objects in the scenery. Under certain circumstances the detection of the fiducial is still possible, while the estimated center is biased away from the occlusion.

Some tracking systems detect partially occluded markers and do not give pose estimates [23], others try to deal with partial occlusions as much as possible [57, 56].

Anisotropic Behavior of Fiducials

Depending on the type of fiducial, the pose estimation process can be affected by considerably large systematic effects. Active infrared emitting diodes have an angle-dependent anisotropic lighting behavior [92]. Circular shaped flat disk features have the property that they will under projection always be seen as ellipses; the center of the ellipse however does not coincide with the projected center of the circle (fig. 5.14) [167, 144].

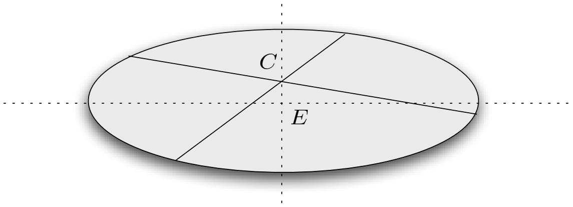


Fig. 5.14. Difference between the projected center C of a circle and the center E of the projection of the circle.

This error however can be corrected under certain circumstances when the target geometry is known [2]. Another approach is the definition of a maximum viewing angle for every fiducial [131]. Both methods necessarily lead to an iterative pose estimation process. The affected fiducials are removed from the fiducial list and the pose estimation process is repeated.

Using Residuals for Error Estimation

After the pose estimation process we can compute for every fiducial P and every camera C_i the backprojected residual (fig. 5.15). This is the distance between the projection P_i of the estimated three-dimensional location of P and the original two-dimensional position P_i' of the fiducial on the image plane [183].

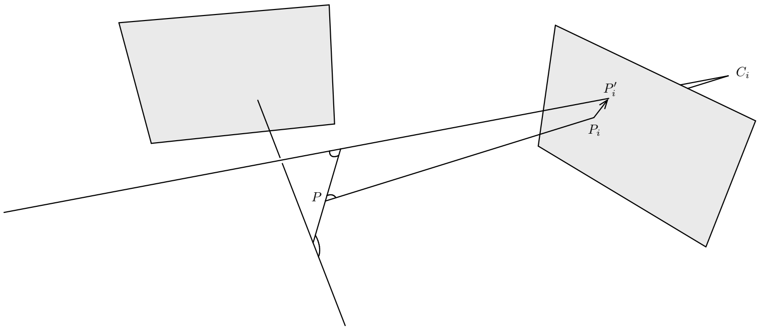


Fig. 5.15. The backprojection of the three-dimensional residual error onto the image plane can be used to adjust the image plane covariance and to detect systematic errors

This distance is a measure for the correlation between the measurement of this camera and the overall estimate. Therefore this could be used to adjust the image plane covariance accordingly. Additionally, an average of the distance vectors over time could be used to detect systematic errors resulting from errors in the extrinsic camera parameters from wrong calibration or accidental change of the orientations of the cameras after calibration as a consistency check.

5.3.3 Estimating the MTE from FLE

The estimation of the error in position and orientation of a target with several fiducials from a given error distribution for every fiducial has been of interest in the medical communitiy for a while [60], mainly in the area of multimodal medical image registration [216, 215] and the measurement of human body movements [224, 123] but has been extended by West et.al. [218, 217] to the accuracy of optical tracking systems. In their work, only independent and isotropic errors are considered.

Davis et.al. [48, 49, 47] presents another method to predict the expected tracking error using a first-order propagation of the errors associated with the fiducials of the target. In his work, he allows both the fiducial configuration and the pose estimation to be affected by gaussian one-dimensional noise.

Ma et.al. [112] use spatial stiffness analysis to predict the registration error depending on the fiducial configuration.

In our work we follow the method presented by Hoff et.al. [84, 83]. We start with computing the 6D error covariance $\varSigma_c \in \mathbb{R}^{6\times 6}$ of the marker target in the centroid c, assuming that we know the fiducial location error (FLE) for every feature point p_i, not only as a single RMS value [60] but as an arbitrary covariance matrix $\varSigma_{\mathbf{p}_i} \in \mathbb{R}^{3\times 3}$.

Without loss of generality we consider a marker target defined as a set of feature point coordinates $\{q_k\} \in \mathbb{R}^3$ in a local coordinate system with the origin in the centroid of the marker balls, and their respective counterparts $\{p_k\} \in \mathbb{R}^3$ in the tracker coordinate system, which are additively disturbed by zero-mean Gaussian errors $\{\varDelta_{p_k}\} \in \mathbb{R}^3$. The according covariance matrices are given as $\varSigma_{p_k}$, assuming independent error for every fiducial. A possible correlation between the two-dimensional coordinates of the jitter in the camera image would need to be investigated further [198].

Note that, for the propagation to work properly, the origin has to be defined at the centroid of the marker target. Otherwise, the positional error would be artificially increased due to its distance from the origin. We will take this effect into account separately when we propagate MTE errors to points of interest.

From the corresponding point sets we estimate the pose of the target by solving the *3D/3D Pose Estimation* problem using any kind of algorithm, for example [87, 106]. This estimation leads to a homogeneous transformation $[R|t]$ which maps

$$R(\varDelta_R\, q_k + \varDelta_t) + t = p_k + \varDelta_{p_k}$$

with some unknown error $[\varDelta_R|\varDelta_t]$ of the whole transformation and the residual $\varDelta_{p_k}$ for every fiducial k. Our goal is to estimate the error $[\varDelta_R|\varDelta_t]$.

To this end we first split with the use of equation 4.10 the transformation into a error-free transformation and a transformation that contains only the error. This means we transform the coordinate system such that all coordinates p_k and the respective covariances are given in the marker target coordinate system by propagating the individual covariances with $R^T \Sigma_{p_k} R$.

Using Euler angle representation for $[\Delta_R | \Delta_t]$, we can treat the orientation and position of the centroid $\mathbf{c} = (x, y, z, \alpha, \beta, \gamma)^T$ as a random variable that represents the MTE. To apply the error propagation, we linearize the error function $f(\mathbf{p}, \mathbf{q}) = R\mathbf{q} + \mathbf{t} - \mathbf{p}$ by computing the Jacobian

$$J_f(\mathbf{q}) = \left. \frac{\partial f(\mathbf{p}, \mathbf{q})}{\partial \mathbf{c}} \right|_{\mathbf{c}=\mathbf{0}} \tag{5.3}$$

with respect to the 6D pose $\mathbf{c}$ at the target centroid and evaluate the Jacobian at the pose $\mathbf{c} = \mathbf{0}$. The resulting Jacobian is the same as the upper part of equation 4.11. This is because we consider only the three-dimensional position of the fiducials here since they do not have an orientation.

$$J_f(q) = \begin{bmatrix} 1 & 0 & 0 & 0 & q_z & -q_y \\ 0 & 1 & 0 & -q_z & 0 & q_x \\ 0 & 0 & 1 & q_y & -q_x & 0 \end{bmatrix}$$

This Jacobian maps the 6D pose error Δx of the target to the respective 3D feature errors, and we can stack the equations for all fiducials together in a single matrix $M \in \mathbb{R}^{6 \times 3n}$:

$$\begin{bmatrix} \Delta_{p_1} \\ \vdots \\ \Delta_{p_n} \end{bmatrix} = \begin{bmatrix} J_f(q_1) \\ \vdots \\ J_f(q_n) \end{bmatrix} \Delta x = M \Delta x \tag{5.4}$$

Using the backward propagation formula (3.6) we get

$$\Sigma_{\mathbf{c}} = \left(M^T \begin{bmatrix} R^T \Sigma_{p_1} R & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & R^T \Sigma_{p_n} R \end{bmatrix}^{-1} M \right)^{-1} \tag{5.5}$$

for the MTE covariance $\Sigma_{\mathbf{c}}$ in the centroid of the marker target (where $R^T \Sigma_{p_k} R$ are the FLE covariances in the target coordinate system).

Note that to visualize this error covariance in the original world coordinate system, we should again retransform the resulting covariance matrix covariance matrix by computing $R \Sigma_{\mathbf{c}} R^T$ according to equation 4.9.

5.3.4 Estimating the TRE directly from the IPE

The two steps from section 5.3.2 and section 5.3.3 could also be integrated into a single computation step by constructing a Jacobian matrix of a function that maps the six-dimensional marker target error to the respective two-dimensional image plane errors. In that case we do not compute individual three-dimensional fiducial location error covariances, but rather the six-dimensional marker target error covariance directly. A similar method is used in section 5.4.2 to estimate the accuracy of monocular tracking systems. Depending on the algorithms used in the pose estimation process, this function could also include additional algorithmic details that go beyond a simple least-squares estimation as in the presented case.

Since this function is highly dependent on the actual implementation of the pose estimation process in the tracking software, we do not consider this in our generalized description here. However, it should be included in the accuracy estimation for a specific tracking system.

5.3.5 Estimating the TRE from MTE

From this 6D MTE covariance in the centroid we can compute the 3D TRE covariance $\Sigma_{\mathbf{p}}$ at a point other than the centroid by applying the forward propagation formula (3.5).

Again we use the Jacobian J_f from equation 5.3 evaluated at the point of interest $\mathbf{p}$. The target registration error at the point of interest is then given by forward propagation with

$$\Sigma_{\mathbf{p}} = J_f \Sigma_{\mathbf{c}} {J_f}^T \tag{5.6}$$

Assuming that the rotational part of the error in the centroid is independent of the positional part, it is then easy to see that the positional error at the point of interest is equal to the positional error at the centroid plus a positional error coming from the propagated rotational error [60], which is increasing proportionally to the distance from the centroid. Examples for this can be seen in chapter 6 and 8.

Here we have assumed that the position of the point of interest is known without errors. In general, this transformation might be affected by errors coming from a calibration process (see section 7.1) as well. In this case we would rather use the error propagation from section 4.2.2 here.

5.4 Monocular Vision

The estimation presented before can be only used for n-ocular tracking systems, since the three-dimensional position of the fiducials in space need to be known. There is however a simple extension of this analysis to monocular tracking systems, which we are going to develop in this section.

5.4.1 Monocular Tracking Systems

Monocular tracking systems are widely used in Augmented Reality and photogrammetric applications. In contrast to n-camera systems, not only less hardware and also no synchronization between the cameras is required. Also the calibration of the epipolar geometry between the different cameras can be omitted.

While there are many different kinds of monocular tracking systems [107], we consider in our analysis only those that are using point-based fiducials. A marker is therefore defined by a set of fiducials. By knowing its geometrical model it is possible to obtain all six spatial constraints from the correspondences of four or more points [227].

Flat Visible Light Markers

Flat visible light markers have been used in near-range photogrammetry for example in crash tests and for quality assessment in metalworking already for a long time. The identification of corresponding points is done here often with user interaction or using special application-specific heuristics.

To make the process automatic, markers have been proposed that consist of an easy detectable black square frame that provides at least the coordinates of the four corner points. Inside that frame, some kind of code is used to distinguish several markers from each other. Many systems use a binary code [143, 228, 57] for identifying the marker, but also two-dimensional (color) patterns [23] or fourier-encoded greylevel patterns [132] are used. Using this code, not only the identification of the whole marker is possible but also the identification of corresponding points. The image of the square is rectified using a homography H and then the pose is computed from this homography [107].

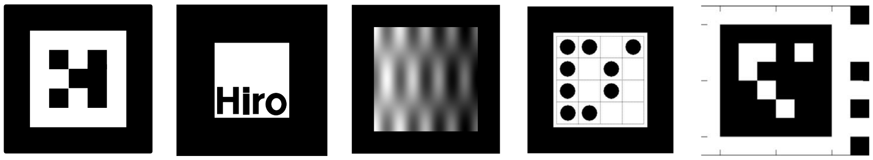


Fig. 5.16. Examples for Planar Tracking Markers [143, 23, 132, 228, 228]

Active and Passive Infrared Fiducials

Another different class of monocular tracking systems uses retroreflective marker targets as common in multicamera tracking setups. The identification of the corresponding corner points which is needed for the pose reconstruction is solved either by specially designed markers [152, 198] or generally by using an initial estimate of the pose based on depth estimations [167]. Wang et.al. [204] used active infrared emitters mounted on the ceiling. The emitters use different pulsating patterns for identification. Our error and accuracy analysis can be used for both kinds of tracking systems.

5.4.2 Accuracy of Monocular Tracking

Due to the widespread use of monocular tracking systems, several groups have worked on analyzing the accuracy of planar marker tracking systems. The work was mainly based on the widely used ARToolkit [23] but in general could be applied to any of these tracking systems. Malbezin et.al. [115] measured the accuracy in a distance of up to three meters and compared the pose estimation to manually measured values. Zhang et.al. [228] compared four different planar marker tracking systems with respect to processing time, accuracy and recognition performance. Abawi et.al. [1] provide a coarse accuracy function depending on the distance and angle between marker and camera. Pentenrieder et.al. [134] use a Monte-Carlo simulation to generate a more general accuracy function for a planar marker tracker. The main consensus of these evaluations is a decreasing accuracy with the distance and a viewing angle dependent behavior with maximum accuracy around a viewing angle of about 45°.

Our approach goes one step further and tries to analyze the *theoretically achievable* accuracy by analyzing the geometric setup of the marker and the camera and predicting the accuracy in terms of a error covariance matrix as we did in the n-camera case. Therefore we do not care about the detection of the marker and the marker features as well as the identification of the features.

Theoretical Model

From now on we assume that the topology of the target is known exactly and all the features from the target can be well recognized and that the 2D position estimation is unbiased but not noise free. So the 2D positons can be interpreted as two dimensional gaussian distributed error variables with zero-mean and a covariance matrix in two dimensions, which describes the non-isotropic jointly distributed error of position estimation.

Derivation of Error Propagation Formulas

In general, this analysis follows the ideas presented in section 5.3.2 and 5.3.3. Since no three-dimensional estimate of a single fiducial point is possible using only one camera, we need to join the two steps to a single step where we estimate the accuracy of the six-dimensional pose estimate of the whole target from the image plane errors directly.

Without loss of generality we use a spherical coordinate system for the following analysis of the accuracy to get results that are comparable with the published experimental results in the literature [1, 115, 228, 151]. The target is fixed in position and orientation and only the camera is moving on a sphere with radius r.

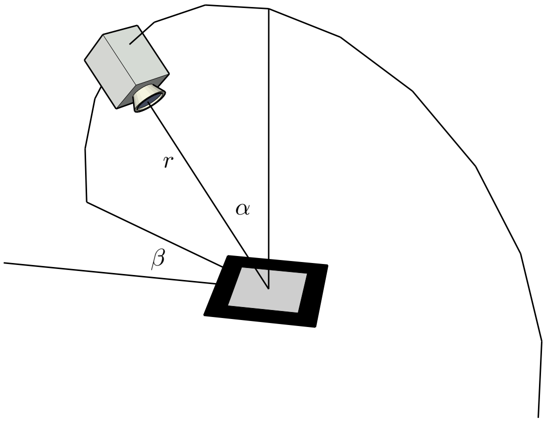


Fig. 5.17. Setup for analyzing the theoretical accuracy of a monocular tracking system with planar fiducials.

We use the same pinhole camera model as in section 5.3.2, and again assume that the camera parameters are known without error [162].

$$p : \mathbb{R}^3 \to \mathbb{R}^2, \qquad x \mapsto \rho \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KT}x$$

and the set of nonlinear camera equations

$$\begin{aligned} p: \qquad \mathbf{u_1} &= \frac{1}{\rho_1}\mathbf{KTx_1} \\ \vdots \quad &\qquad \vdots \\ \mathbf{u_n} &= \frac{1}{\rho_n}\mathbf{KTx_n} \end{aligned}$$

The same camera maps the different points x_i to different locations u_i on the image plane.

Without loss of generality we parametrize the extrinsic camera parameters (and therefore also the error) using three parameters α, β and r. This means we have no *roll* around the camera axis and assume that the camera is always centered towards the origin where the marker is placed. A parametrization using standard Euler angles could be used as well. In this case the Jacobian in the following section would be done with respect to all Euler parameters instead.

The transformation T are the extrinsic camera parameters which map the fiducial points x_i into camera coordinates. Again we decompose the transformation according to eq. 4.10 into the transformation and the unbiased error. Now we can compute analogous to equation 5.4 the Jacobian of p evaluated at the origin

$$J_{\mathrm{p}} = \left. \frac{\partial\, \mathrm{p}}{\partial(\alpha, \beta, r)} \right|_0 \in \mathbb{R}^{3\times 2n} \tag{5.7}$$

and apply the backward propagation formula (3.6).

Assuming points on the image plane can be detected with uncertainty given by the covariance matrices $\varSigma_{v_i} \in \mathbb{R}^{2\times 2}$. Then the covariance matrix for detecting the planar fiducial is

$$\varSigma_{\alpha,\beta,r} = \left(J_{\mathrm{f}}^{T} \begin{bmatrix} \varSigma_{v_1} & & \\ & \ddots & \\ & & \varSigma_{v_n} \end{bmatrix}^{-1} J_{\mathrm{f}} \right)^{-1} \in \mathbb{R}^{3\times 3} \tag{5.8}$$

Now the covariance of the spherical angles α, β and the distance r is known and can be calculated for any arbitrary position of the camera or of the target, respectively.

Accuracy Depending on the Viewing Angle

We can now use this theoretical model to derive error functions for monocular tracking systems. We do this on the example of planar markers.

First we analyze the influence of the viewing angle or marker tilt (α in fig. 5.4.2) on the accuracy in distance and angle. Figure 5.18 shows the predicted standard deviation of the depth estimation for r (left) and the tilt angle (α). The angle α changes from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ which is a full half circle at constant distance r around the marker which is placed in the origin.

We can see from the graph that the angular accuracy is worst around the pose where the camera is looking frontal onto the flat marker. This is reasonable since in this position, a small change in the tilt angle only changes the projected image slightly. Therefore, slight random noise in the image leads to larger errors in the estimated tilt angle. The distance error in the centroid in contrast is a bit smaller around this pose, since small errors do not affect the distance too much here. However, due to the error propagation, the large angular error will cancel this effect out when moving away from the center. For estimating the camera pose relative to the marker see figure 5.19. The presented derivation estimates the pose of the marker in camera coordinates,

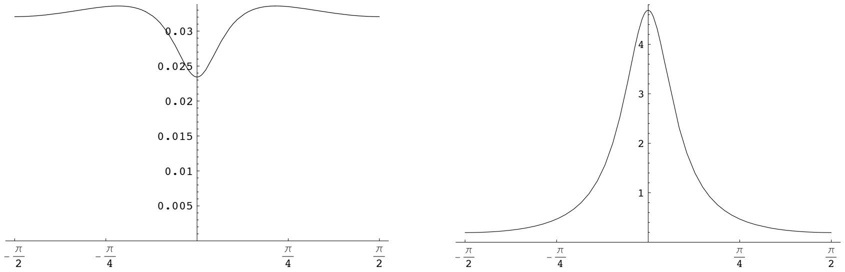


Fig. 5.18. Influence of the marker tilt to the accuracy in distance (left) and angle (right). Note that usually tracking is not possible at tilt angles larger than 80°

which is what most simple augmented reality applications actually need for displaying virtual objects in the camera image. If the pose of the camera in marker coordinates is required (inside-out tracking), this involves inverting the tracked pose and at the same time applying the error propagation for pose inversion from section 4.2.3. The result can be seen in figure 5.19: The ellipsoids show a large pose error when the camera is pointing frontal to the marker, and smaller errors at larger tilt angles.

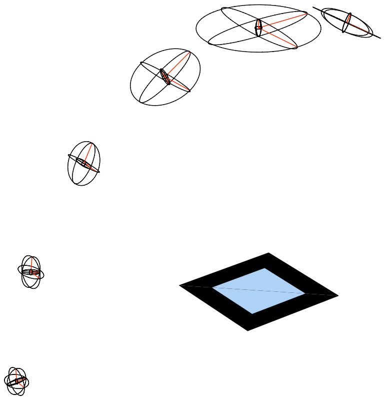


Fig. 5.19. Accuracy of the pose of the camera in marker coordinates (inside-out tracking) is rather weak due to the pose inversion and high angular uncertainty. The covariance ellipsoids are magnified for illustration of the difference only.

The same behavior can be found in the experimental evaluations [1, 151, 115]. From this result we can conclude that planar markers should be positioned always in such a way that the camera can look at the marker at a certain angle and not frontal. It is obvious that non-planar markers will have better performance also in moncular tracking systems [198].

We have verified these results using Monte-Carlo simulations. For every tilt angle we have computed the projected image of the marker and applied unbiased random noise to the two-dimensional locations. From these values we reconstruct again the pose of the camera which we can then compare to the ground truth.

Note that the errors might in reality not be unbiased due to the feature detection algorithms. Also the standard deviation of the fiducials might be different for different tilting angles due to the perspective distortion.

Accuracy Depending on the Distance

It has been noted earlier [1] that the positional error of a monocular tracking system increases more than linearly with the distance. This can be observed from figure 5.20 as well.

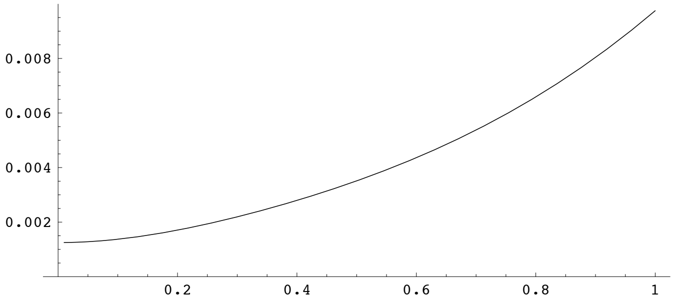


Fig. 5.20. Influence of the distance to the marker to the accuracy in distance measurements for a fixed tilt angle

Pentenrieder et.al. [134] have analyzed the accuracy of distance measurements in a series of simulation runs and come to the same result.

5.4.3 Natural Feature Tracking and Point-based Registration

The presented estimation was performed using an artificial fiducial tracking system. However, the same analysis could theoretically be applied to a natural feature tracking system as well. In fact in natural feature tracking systems, quite similar estimations [40] have been performed, but so far only to improve the tracking internally. The estimated accuracy values could additionally be used for further error estimation inside a larger tracking framework [200].

Another application of this theory is the field of three-dimensional image registration, where so far only isotropic and independent errors are considered [216]. However, in these applications there is usually not too much propagation of the errors since the final volume of interest is close to the centroid of the fiducial distribution, such that the difference to the currently used error models is not too large.

6

Designing Targets for Optical Tracking

In the previous chapter we have presented a method for the estimation of the expected accuracy in an optical tracking system. This estimation function returns the expected accuracy for a given camera configuration and a specific tracking target at a specified pose inside the tracking volume.

When designing applications involving optical tracking systems, this function can be used to evaluate the expected accuracy at typical scenarios. It can not directly be used to compute the ‘optimal’ setup of cameras and marker fiducials. Davis et.al. [49] use *simulated annealing* and a simpler error function to design optimal tracking targets under the assumption that the fiducials will be located on the surface of an existing CAD model. The CAD model enables them to consider visibility of the fiducials as well [48]. West et.al. [218] analyze the error for various fiducial configurations and give some conclusions for common configurations for the tracking of a pointing device and an endoscope.

In general, the problem of finding an optimal setup of cameras and fiducials has far too many parameters to be solved automatically. In particular, some of the most obvious ways of increasing the accuracy, as we will see later, are only bounded by the usability for the final application — a parameter that can hardly be fitted into a formula for some optimization process.We will rather give a set of rules and insights to show how various changes on the setup influence the final accuracy when designing a tracking system.

As a side effect, we will show that it is important for a useful runtime estimate to do the error estimation also online and use all available information on how many cameras and fiducials were actually used in a particular measurement (cf. section 8.4). These are crucial parameters for the accuracy, and this information is only available online and usually only inside the tracking system software. Therefore it would actually be the task of the tracking software to deliver these online error estimates.

6.1 General Rules

It is desirable to define some general rules of thumb for the construction of tracking targets with optimal accuracy. It can be shown [218] that the best accuracy is achieved when all fiducials are distributed with equal distance on a sphere around the point of interest. While this can be achieved in rigid three-dimensional data registration [216], is is usually not an option for optical tracking. Here we have to make sure that the tracked tool is still usable.

We give now an overview of potential rules and their applicability in practice.

6.1.1 Use More Fiducials

The most obvious way of improving the accuracy might be the use of more fiducials. The error decreases with n being the number of fiducials by $\frac{1}{\sqrt{n}}$ [60]. This can be derived from the quadratic form in equation 5.5. However, using more fiducials also increases the occurrence of occlusions between the fiducials. With too many mutual occlusions, no tracking is possible at all. This limits the amount of markers in realistic setups, since the increase in accuracy is rather slow with more than five or six fiducials, while the probability of occlusion increases when the diameter of the fiducial distribution is not increased at the same time. Additionally, too many fiducials might interfere with the originally intended use of the tracked object. In most cases, more than five fiducials are not required, since other effects have larger influence on the final accuracy.

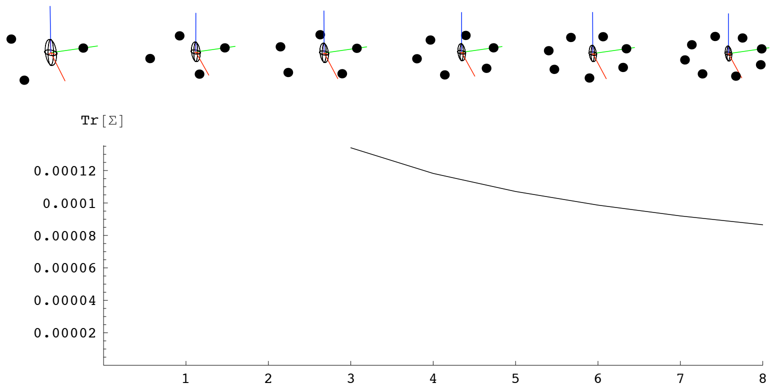


Fig. 6.1. Error covariance in the centroid for increasing number of fiducials

Figure 6.1 shows how adding fiducials influences the resulting accuracy in the centroid. Note that the arrangement of the fiducials does not affect the accuracy in the centroid, which is depicted in the graph.

6.1.2 Use a Larger Diameter for the Fiducial Distribution

A second obvious method to increase the accuracy is to use a more widespread fiducial distribution [166, 108, 60]. However, increasing the diameter of the fiducial distribution does not change the positional accuracy. This can be derived from equation 5.5 when we look at the expanded matrix. Due to the form of the Jacobian J_f and therefore the matrix M, the expanded covariance matrix will only contain terms from the positional covariances of the Σ_{p_i} and no mixed terms from the rotational covariances.

In contrast to that, the rotational accuracy does indeed increase inversely proportional with a larger distribution [224, 60]. Therefore it is a good rule to make the distribution of fiducials as large as possible, in particular when a propagation of the rotational error is expected. Unfortunately, the diameter can only be increased as long as the usability of the tracked object allows this (cf. section 8.1).

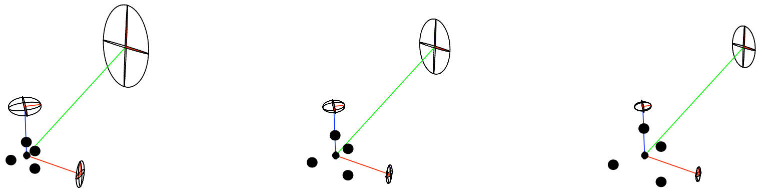


Fig. 6.2. Propagated error covariances along three coordinate axes for a tetrahedral fiducial configuration of different size (increasing from left to right)

Figure 6.2 shows how the propagated error decreases with a larger diameter of the fiducial distribution.

6.1.3 Put the Centroid Close to the Point of Interest

The largest part of the final tracking error is coming from propagated rotational errors. This propagation is smallest at the centroid of the fiducials [224]. Unfortunately it is usually not always possible to position the fiducials in a way such that the centroid lies at the point of interest. For example in a pointing device this would make pointing impossible. When tracking an endoscope, the point of interest actually lies inside the body of the patient and therefore all fiducials have to be rather far away from that point. The error increases linearly with the distance from the centroid (see figure 6.3).

If the point of interest is in fact a whole area or volume of interest, make sure that all points in the volume are as close as possible to the centroid. Note also that ‘close’ in this case means close relative to the radius of the fiducial distribution.

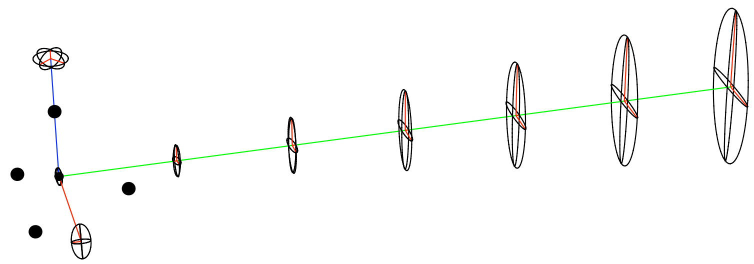


Fig. 6.3. Propagated error covariances at points of interest with different distance to the centroid of the fiducials

6.1.4 Improving the Image Plane Covariance

The resulting error at the point of interest depends directly on the detection covariance on the image plane. Therefore, all methods that are able to improve the image plane covariance directly reduce the resulting error. This includes methods that increase the size of a fiducial on the image plane, like the use of larger fiducials (cf. section 5.3.2), a higher resolution of the camera chip, or a longer focal length a as well as changes in the lighting conditions or the detection algorithms.

Usually however, many of these methods are unavailable when using an existing commercial tracking system.

6.1.5 Special Rule for Pointing Devices

A pointing device that is used only to identify three-dimensional points does not need to provide rotational accuracy around the pointing axis (cf. figure 6.9). In this case, accuracy around this screw axis does not matter. Therefore, the diameter of the fiducial distribution in this dimension can be really small. However, this is not valid for devices which need actual rotation information, like in an endoscope [218].

Some intraoperative navigation systems use pointing devices with collinear fiducials. With minor modifications, the error estimation can be applied to such targets as well. Since there is no information about the rotation around the pointing axis, this dimension simply needs to be excluded from the Jacobian matrix and the resulting covariance.

6.1.6 Planar Marker

For manufacturing reasons it is sometimes easier to produce targets with planar fiducial distribution. While the positional error in the centroid is not affected by the distribution, the propagated error depends on the rotational error. This error is in all dimensions larger for planar markers [218].

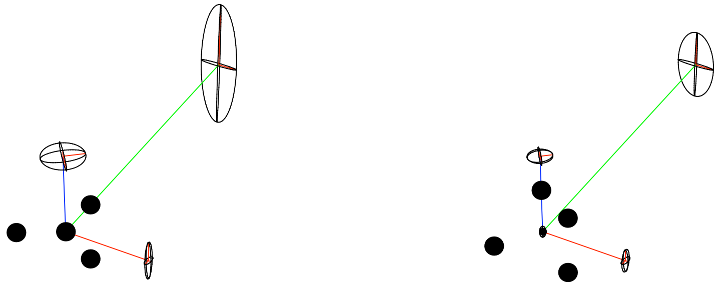


Fig. 6.4. Propagated error covariances for a flat fiducial distribution (left) compared to a tetrahedron configuration (right)

Figure 6.4 shows an example of the propagated error for a planar marker as compared to a volumetric one. This means that volumetric markers should be used whenever possible to increase the accuracy.

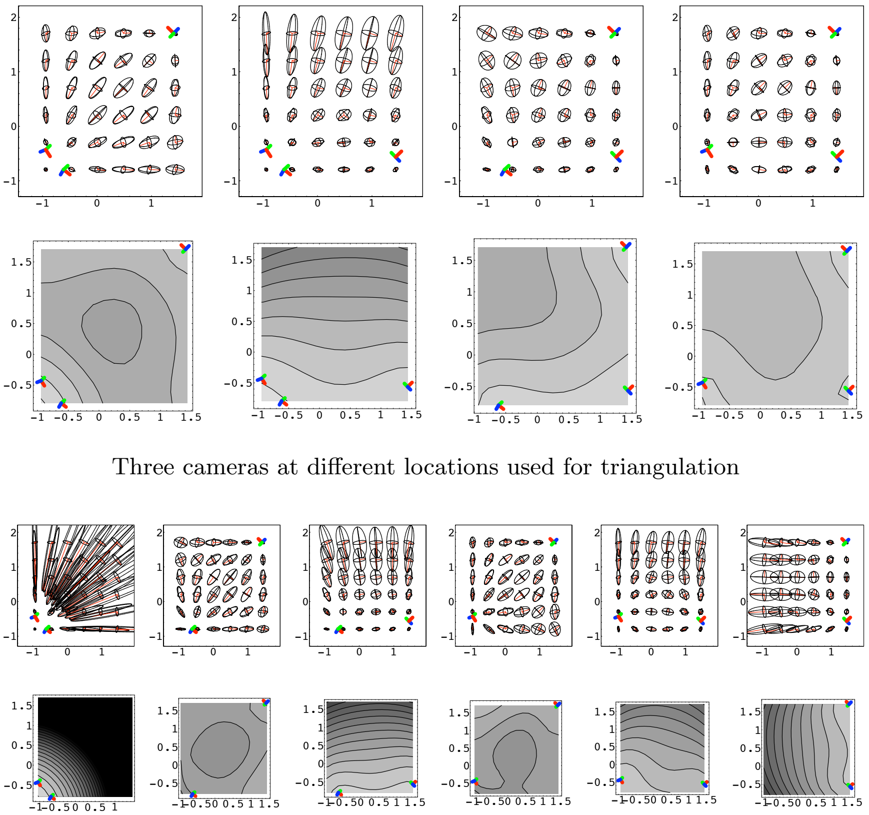
6.2 Dealing with Occlusions

When we started with the analysis of optical tracking systems in chapter 5, we have stated that we do not consider the visibility of the fiducials; our analysis is based on the fact that all fiducials we use for the error estimation were actually seen by all cameras. When we want to use this analysis for the design of optical tracking targets, we can not assume this any more.

6.2.1 Occluding Cameras

The optimal camera configuration in terms of errors is a configuration where all cameras are evenly distributed on a sphere around the point of interest [31]. In reality however, there are usually some restrictions on where cameras can be placed [37, 36].

Looking at figure 6.5, we can see that the effect of occluding cameras has a rather large influence on the resulting error. When designing tracking setups,



Three cameras at different locations used for triangulation

Two cameras at different locations used for triangulation

Fig. 6.5. Effect of the occlusion of different cameras on the resulting FLE error. Covariance ellipsoids and RMS errors.

this means that we do not only need to consider visibility under partial occlusions, but additionally need to make sure that the remaining configuration still provides useful pose estimates when some cameras are occluded.

In a runtime system this means that different fiducials might have quite different fiducial location error, since a different number of cameras has been involved in detecting each particular fiducial.

6.2.2 Occluding Single Fiducials

In a similar way we can analyze the effect of the occlusion of single fiducials. Again, it is important to consider how the fiducial layout changes when single fiducials are occluded. Figure 6.6 shows the difference between a rhombus layout of four fiducials and a square layout (cf. section 8.2).

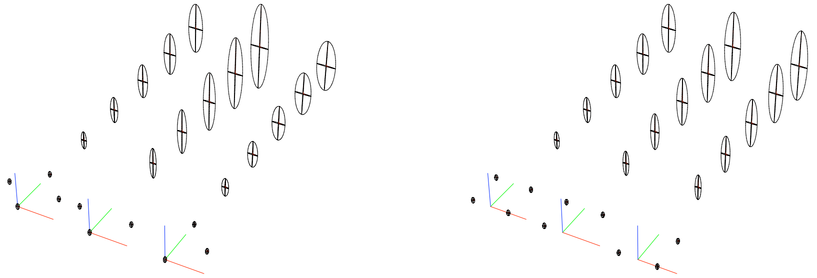


Fig. 6.6. Difference in error propagation in case of occlusion – Comparison of different target layouts: rhombus (left) and square (right)

While there is no difference in the propagated error when all four markers are visible, occluding single markers reveals a difference. This difference is mainly due to the fact that occluding a marker also shifts the centroid. Therefore it is useful to place redundant markers close to the point of interest, such that occluding one of these markers does not shift the centroid too far away from the point of interest. Additionally, for good angular accuracy it is important to maintain the maximum fiducial distribution radius even under occlusions as good as possible, especially along the main pointing direction.

6.3 Inside-out vs. Outside-in Tracking

It is a frequent practice to classify tracking systems into *inside-out* and *outside-in* systems. These terms were coined by Wang [204, 203] and then adopted by many others [148, 21, 28, 102].

Figure 6.7 explains the concept with an easy example. The outside-in configuration is common in many commercial tracking systems: The camera is mounted fixed in the environment and senses the object. The rotation of the object, the cube in this case, produces a small motion of the image on the CCD camera. In contrast to that, in the inside-out configuration, the rotation of the camera, which is movable in this case, produces a large motion of the image of the cube on the CCD camera [148]. Sometimes, the term *inside-in* is additionally used to denote tracking systems where both the sensor and the sensed properties are part of the object. One example are data gloves that track the internal state of the hand without providing information about its pose relative to other objects.

Due to this observation, it has been frequently stated [102, 21, 148] that an inside-out system can compute angular orientation of the target with higher accuracy than an outside-in system that uses comparable technology.

It is easy to see that the difference between both configurations is only the inversion of the pose estimate. An optical tracking system always senses the

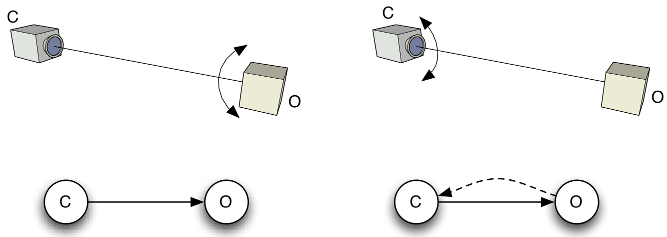


Fig. 6.7. Outside-in (left) and inside-out (right) configuration of the same tracking system. The difference is in the direction of the pose estimate.

position of the fiducials relative to the camera coordinate system and computes the pose from that. We have shown in section 4.2.3 that the inversion of an edge in the spatial relationship graph is nothing than a coordinate system change with subsequent error propagation to the new endpoint. Equation 4.15 shows that the magnitude of angular precision does not change under inversion.

And in fact what we have observed in the inside-out tracking system is not the rotation of the camera but rather the change in position of the tracked object. The relative orientation between the two objects depends on the orientation of the object relative to the camera, just as in the outside-in case.

The inside-out configuration does not yield higher resolution and accuracy in orientation than the outside-in. The angular accuracy depends only on the relative baseline of the cameras and the fiducials in the marker.

From an applications point of view, there can still be a large difference. The main error in common augmented reality setups comes from the propagation of initially small angular error through the whole spatial relationship graph. By making this propagation chain as short as possible, for example by mounting the tracking system in an inside-out configuration close to the actual application, even a cheap inside-out system can outperform an expensive stereo system [122]. The main drawback of outside-in systems as opposed to inside-out systems is the longer chain of error propagations, not larger errors by itself. All that matters for the error is the size and orientation of the baseline.

The main difference between inside-out and outside-in configurations actually is the fact that a movable inside-out system can not be as large or heavy as a fixed installation. Outside-in tracking systems typically use more than one camera while inside-out systems are monocular, but there are also lightweight inside-out tracking systems using several cameras [206], a panoramic camera [219, 220] or laser rays in different directions [199].

Nevertheless it is still a good idea to combine outside-in and inside-out approaches. What actually matters for good augmentation is positional accuracy after the propagation, and using a clever combination of complementary positional measurements [84], the resulting accuracy can be improved.

6.4 Influence of Anisotropic FLE on the Final TRE

Fitzpatrick et.al. [60] already mention that *'it is not always a good assumption that FLE is isotropically distributed'*. They have shown [214] that the theory is still valid for multimodal registration of computer generated medical images where the point of interest is not too far away from the centroid, or even inside the bounding box of the fiducial distribution. However, for the case of optical tracking of cameras or tools, the point of interest lies usually farther out. We will see that in this case the anisotropy of the fiducial location error does in fact matter.

6.4.1 Input Covariances (FLE)

To this end we compare both error prediction methods for the same fiducial configuration and input error. We use a realistic covariance matrix Σ from one of our lab setups as shown in figure 6.8.

$$\Sigma = \begin{pmatrix} 1.987 \cdot 10^{-9} & 7.129 \cdot 10^{-10} & 9.453 \cdot 10^{-11} \\ 7.129 \cdot 10^{-10} & 1.733 \cdot 10^{-9} & -2.103 \cdot 10^{-10} \\ 9.453 \cdot 10^{-11} & -2.103 \cdot 10^{-10} & 1.214 \cdot 10^{-8} \end{pmatrix}$$

This is an example covariance from a four camera setup. Note that in the quite common two camera setups, the anisotropy of the error is even more significant (cf. figure 5.6 and 6.5).

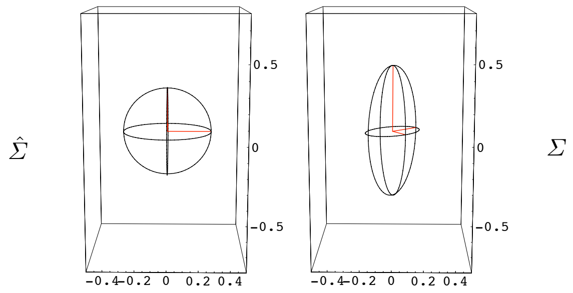


Fig. 6.8. Visualization of the input covariances (75% confidence level in mm) for the isotropic case (left) and the anisotropic case (right)

To get the according isotropic error for this slightly anisotropic covariance, we use a diagonal matrix with $\frac{1}{3}Tr(\Sigma) = 5.288 \cdot 10^{-9}$ as the diagonal entries:

$$\hat{\Sigma} = \begin{pmatrix} 5.288 \cdot 10^{-9} & 0 & 0 \\ 0 & 5.288 \cdot 10^{-9} & 0 \\ 0 & 0 & 5.288 \cdot 10^{-9} \end{pmatrix}$$

In terms of RMS error this corresponds to $\sqrt{Tr(\Sigma)} = 0.1259$ mm error. The two covariance matrices are displayed in figure 6.8.

The fiducial configuration we are using for this experiment is from a pointing device used in a commercial medical intraoperative navigation system. The pointer and the fiducial distribution can be seen in figure 6.9. It consists in four retroreflective fiducials and a point of interest at the tip of the tool which is about 15 cm away from the centroid of the fiducial distribution with a maximum radius of about 7 cm. This factor is quite common for a lot of tools and therefore serves as a good example.

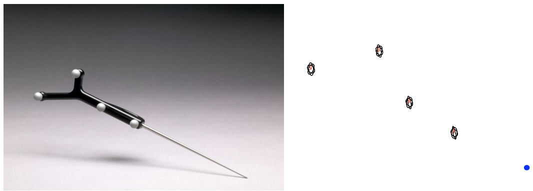


Fig. 6.9. Exemplary commercial pointing device (Northern Digital, Inc.)

6.4.2 Propagation of the Isotropic Error

If we now apply both the TRE estimation [60] and the covariance propagation (CP) [19] to this tool, we get the following results: The RMS target registration error for the isotropic case is 0.172 mm with both methods. For the TRE estimation we get

$$TRE(\hat{\Sigma}) = \begin{pmatrix} 9.905 \cdot 10^{-9} & 0 & 0 \\ 0 & 9.905 \cdot 10^{-9} & 0 \\ 0 & 0 & 9.905 \cdot 10^{-9} \end{pmatrix}$$

However, the shape of the resulting covariance from the proposed covariance propagation method is in fact a flattened disk (cf. figure 6.10) with a smaller error along the axis of propagation (from the centroid to the point of interest) and larger error orthogonal to that axis.

$$CP(\hat{\Sigma}) = \begin{pmatrix} 1.836 \cdot 10^{-9} & -1.959 \cdot 10^{-9} & 1.413 \cdot 10^{-9} \\ -1.959 \cdot 10^{-9} & 1.368 \cdot 10^{-8} & 1.246 \cdot 10^{-9} \\ 1.413 \cdot 10^{-9} & 1.246 \cdot 10^{-9} & 1.420 \cdot 10^{-8} \end{pmatrix}$$

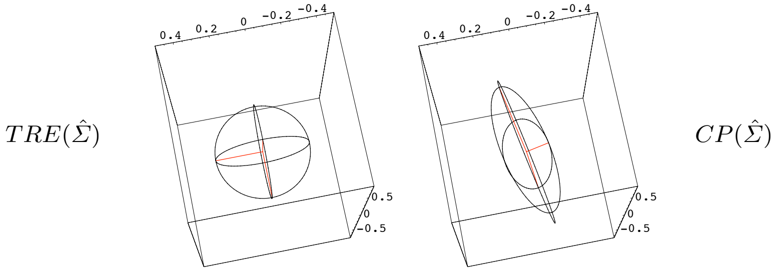


Fig. 6.10. Propagation of the isotropic fiducial location error using the TRE method (left) and covariance propagation (right)

The square roots of the eigenvalues of this matrix are $0.123, 0.115, 0.036$ mm as compared to three times 0.099 mm for the TRE estimation. Therefore the standard deviation of the error is already about 20% larger in two of the three dimension while all have the same RMS error of 0.172 mm (as expected).

6.4.3 Propagation of the Anisotropic Error

If we additionally consider the slightly anisotropic fiducial location error as input, the result is in fact different for different orientations of the tool due to the anisotropic propagation. Propagating the error from the already larger direction results in bigger errors than propagating the smaller part. In figure 6.11 this is done for an exemplary direction where this is visible. Now even the resulting RMS error differs from the RMS error estimated with the TRE method, 0.203 mm in this case.

The according covariance matrix looks as shown here:

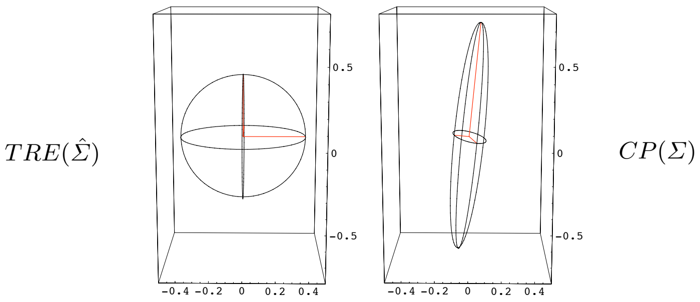


Fig. 6.11. Propagation of the isotropic fiducial location error using the TRE method (left) and propagation of the anisotropic covariance (right)

$$CP(\Sigma) = \begin{pmatrix} 1.0722 \cdot 10^{-9} & -0.732 \cdot 10^{-9} & 3.347 \cdot 10^{-9} \\ -0.732 \cdot 10^{-9} & 7.513 \cdot 10^{-9} & 2.346 \cdot 10^{-9} \\ 3.347 \cdot 10^{-9} & 2.346 \cdot 10^{-9} & 32.61 \cdot 10^{-9} \end{pmatrix}$$

with square root of eigenvalues of $0.182, 0.086, 0.024$ mm while for the TRE estimation we have still three times 0.099 mm.

6.4.4 Target Orientation

As already mentioned, the orientation of the propagation axis from the centroid of the fiducial distribution to the point of interest matters. When this axis is vertical to the bigger parts of the anisotropic FLE covariance, then this error gets propagated. In this case the resulting error is in fact larger than estimated using the TRE estimation. On the other hand when propagating the smaller error parts, the resulting estimate might be even smaller than the one from TRE, as can be seen in figure 6.12. The target is rotated by 90 degree from the worst orientation (in that case horizontal) to the best (in that case vertical).

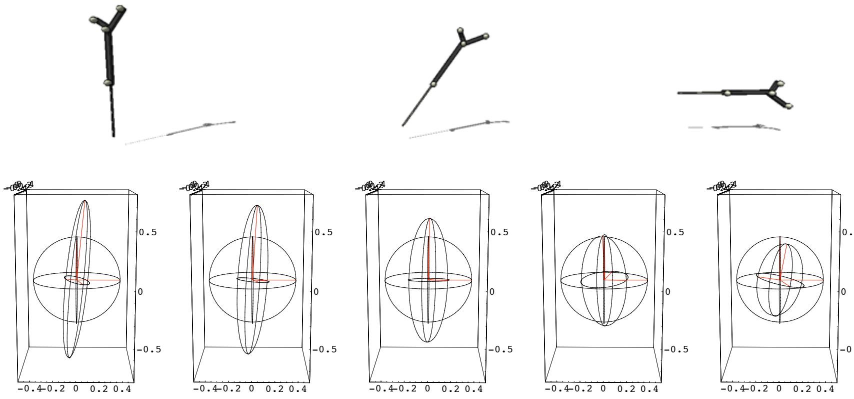


Fig. 6.12. Error propagation for different orientations: Propagating large errors (left) to small errors (right) when rotating by 90 degrees. For comparison, the according TRE estimation is shown additionaly.

One has to note though that for small camera baselines, the best orientation unfortunately coincides with the orientation that provides the worst visibility and most interfiducial occlusions. These occlusions are not considered in this model but still need some attention [48]. We can conclude therefore that it is in fact necessary to consider both the anisotropic error propagation as well as the anisotropic fiducial location errors since they result in larger errors then estimated using the TRE method alone. This gets more significant when the point of interest lies outside of the fiducial distribution which is usually the case for optical tracking of tools and less significant for multimodal registration of medical images where the point of interest usually is inside or close to the fiducial configuration.

6.5 Conclusion

To sum up the statements from the previous sections, we can define the following rules for the design of an optical tracking system setup:

- Try to minimize the error propagation distance by carefully positioning the cameras
- Avoid propagation of already large errors. Pay attention to large rotational errors, they have a huge influence on the final error.
- Use as many fiducials as you can afford without disturbing the usability and without creating too many merging markers.
- Make the diameter of the fiducial distribution as large as possible.
- Bring the centroid of the fiducial distribution as close to the point of interest as possible.
- Try to consider effects of occlusions of cameras and fiducials.
- Try to use the smaller error dimension of anisotropic errors.

After considering all these rules, the accuracy can be estimated and compared to the desired accuracy.

7

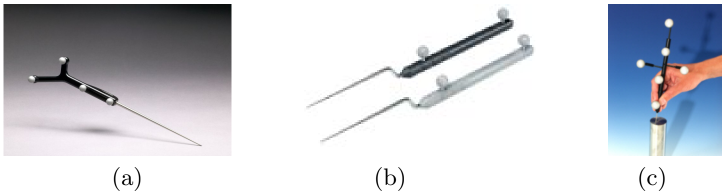
Calibration: Inferred Measurements

Many transformations in a tracking network can not be directly measured. They rather get inferred from other measurements [141]. In this section we give an overview over some of the frequently observed calibration problems and on algorithms used to solve them. In particular, we look at the calibration of pointing devices, the hand-eye calibration problem and the calibration of optical see-through head mounted displays.

7.1 Pointing Device Calibration

Pointing devices are used in many augmented reality applications. They are used for point-based object registration, and for measurements where the point of interest can not be measured directy, but can be touched with a pointer.

There exist a variety of commercial pointing devices for different purposes (cf. 7.1) by different manufacturers. One thing all these devices have in common is the fact that the pose of the pointing device is needed at a certain point of interest where it is not directly measured. The tip offset, the position of the point of interest in local marker coordinates needs to be calibrated unless it is known from manufacturing.



(a) (b) (c)

Fig. 7.1. Different designs of pointing devices for coordinate measurements and point-based registration: (a) Northern Digital, Inc., (b) BrainLab AG, (c) Advanced Realtime Tracking GmbH

7.1.1 Collecting Calibration Data

Pointer calibration is the process of computing the offset p_h of the tip of a pointing device ('caliper') in the coordinate system of the used marker target. To this end the user has to move the pointing device in a way such that the tip is always at the same place in the coordinate system of the tracking device. The pointer rotates around the tip and the positions p_i of the attached marker target, and if possible also the rotations R_i of the target are recorded.

This process is sometimes also called Hot-Spot calibration [221], stylus calibration [66] or pivoting [130].

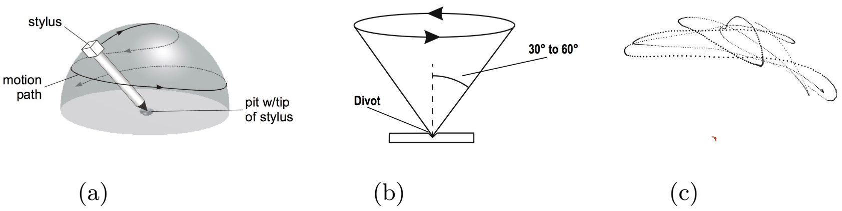


Fig. 7.2. Pivoting instructions from (a) Fuhrmann et.al. [66] and NDI [130]. The pointing device is rotated around the hot-spot and the data of the tracking system is recorded. Collected data points (c) from the pivoting process

The computation of the offset now depends on the fact if orientation data for the recorded positions is available, or not.

7.1.2 Using Orientation and Position

If we have for every measurement not only the position p_i but also the orientation R_i (as rotation matrix) we can set up a system of equations

$$\begin{pmatrix} p_w \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_i & p_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_h \\ 1 \end{pmatrix}, \tag{7.1}$$

Where p_w is the tip of the pointing device in world coordinates and p_h the tip in marker coordinates.

It is obvious that p_h and p_w are constant but unknown. We construct the following linear system of equations [188]:

$$\mathbf{A} \begin{pmatrix} p_h \\ p_w \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1 & -\mathbf{I} \\ \mathbf{R}_2 & -\mathbf{I} \\ \vdots & \vdots \\ \mathbf{R}_n & -\mathbf{I} \end{pmatrix} \begin{pmatrix} p_h \\ p_w \end{pmatrix} = \begin{pmatrix} -p_1 \\ -p_1 \\ \vdots \\ -p_n \end{pmatrix} \tag{7.2}$$

This system is highly overdetermined since we have only six unknowns but potentially hundreds of measurements. To find the least-square solution of the

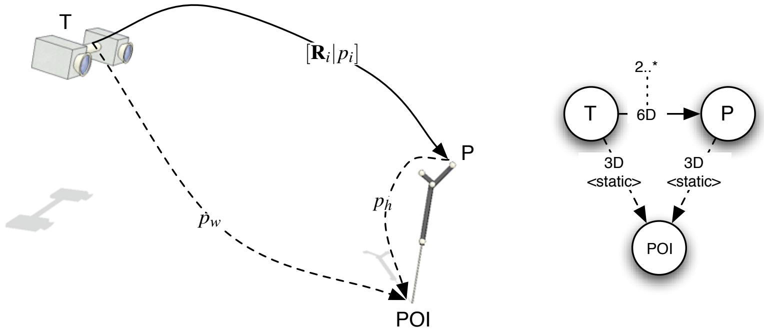


Fig. 7.3. Calibrating a pointing device. Setup and spatial relationship graph.

system, we build the singular value decomposition (SVD) of **A**, which gives us

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{7.3}$$

where Σ is a diagonal matrix with $\frac{1}{w_i}$ in the diagonal elements. Finally we get the following formula for the solution of the linear system of equations:

$$\begin{pmatrix} p_h \\ p_w \end{pmatrix} = \sum_{i=1}^{n} \left(\frac{\mathbf{U}_{(i)} \cdot \mathbf{b}}{w_i} \right) \mathbf{V}_{(i)} \quad \text{with} \quad \mathbf{b} = \begin{pmatrix} -p_1 \\ -p_2 \\ \vdots \\ -p_n \end{pmatrix}$$

and $\mathbf{U}_{(i)}, \mathbf{V}_{(i)}$ the ith column of **U** and **V**, respectively [138].

Additionally it is possible to use the numerical condition of the matrix **A** as a measure of the quality of the solution by building the ratio between the largest and the smallest singular value in Σ. Another way of estimating the quality of the solution is to compute the reprojection with $[R_i|p_i]\ p_h \approx p_w$ and get the covariance of the resulting point cloud. Figure 7.4 shows the reprojection covariance at a 75% confidence level together with the reprojected points (blue) and the measured points (red). In this case the standard deviation was between one and two millimeters.

This error needs to be additionally propagated when estimating the accuracy on the point of interest according to section 5.3.5.

7.1.3 Using only Position

If we have a tracking system that gives us only 3D measurements, we can not use the described algorithm to get the position of the point of interest. However, we can still use this tracking system to measure locations in 3D.

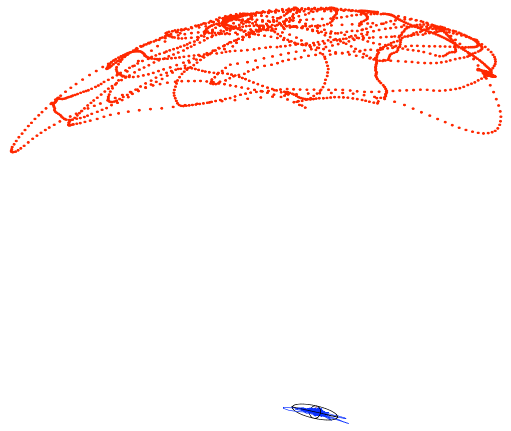


Fig. 7.4. Reprojection of the pointing device tip into world coordinates shows a standard deviation of one to two millimeters in this calibration run.

Four points are sufficient to uniquely define a sphere. Given at least four points p_i that lie on the surface of a sphere, it is possible to calculate the center and the radius of the sphere [137].

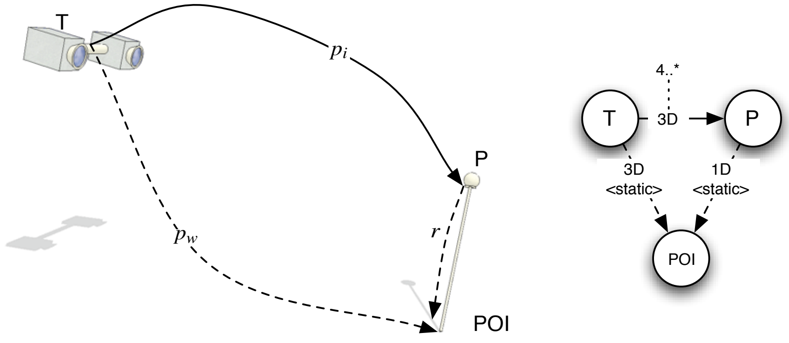


Fig. 7.5. Measurement of a single point. Setup and spatial relationship graph.

We can use this to compute the pivoting point in tracker coordinates. In the presence of measurement errors it is better to use more than just four points and least squares methods to fit the center of a circle to the data [137, 111].

However, this pivoting process needs to be repeated for every point we want to measure in that case, since we do not get the full pose of the tip in tool coordinates here.

7.2 Hand-Eye Calibration

Hand-eye calibration is the process of determining the relative transformation between the extrinsic coordinate system of a camera ('eye') and the coordinate system of an attached external tracking system ('hand'). The external tracking system can be a robot with the camera mounted rigidly onto the gripper. This is where the problem was originally described. More common is the case that a marker target of the external tracking system is attached to the camera. This is for example the case in all video augmented reality systems where the camera image in which the augmentation is performed can not be used for the tracking.

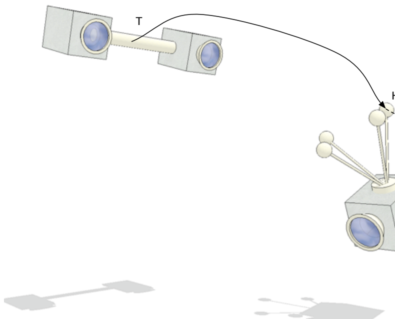


Fig. 7.6. The hand-eye calibration problem: Estimating the offset between the camera coordinate system and the target coordinate system of an attached external tracking marker

One has to note that the problem of computing the hand-eye transformation does under certain circumstances not require algorithms commonly referred to in the literature as hand-eye calibration algorithms. Depending on the kind of markers and tracking systems, several solutions are possible, from a trivial direct solution (7.2.1) to a class of algorithms that solve the most general hand-eye calibration problem (7.2.3). These algorithms can be additionally used to solve other kinds of problems, including the alignment of two independent coordinate systems.

The hand-eye calibration problem is stated as follows: A camera E is equipped with a marker H of an external tracking system T (see figure 7.6). While the transformation $T_{\mathsf{TH}} = [R_{\mathsf{TH}}|t_{\mathsf{TH}}]$ is given by the tracking system, we are searching for the transformation T_{HE} to compute for example poses of virtual objects in camera coordinates.

7.2.1 Direct Solution

If both tracking systems are able to find the same marker in 6D, there is a trivial solution to the problem [22]. We can compute directly

$$T_{\mathsf{HE}} = (T_{\mathsf{TH}})^{-1} T_{\mathsf{TM}} (T_{\mathsf{ME}})^{-1}$$

by matrix inversion and concatenation as seen in figure 7.7.

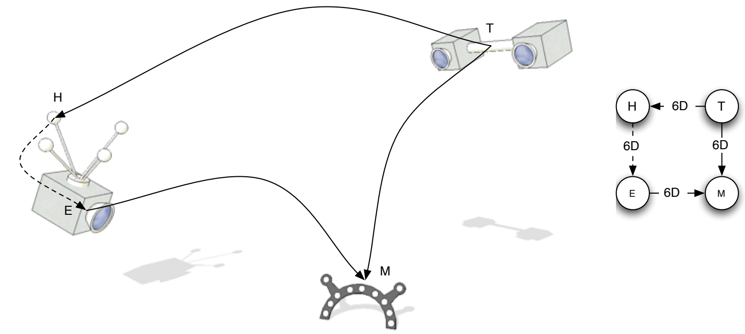


Fig. 7.7. Direct solution: The same marker can be seen in both tracking systems. The desired transformation can be computed directly by matrix inversion and concatenation

The error propagation in this case can be derived easily by applying the propagation rules for inversion (equation 4.15) and concatenation (equation 4.13).

7.2.2 Semi-Direct Solution

In many cases, the camera E is not able to detect the same kind of markers as the tracking system T. However, the relative position between the marker of the tracking system and a set of 3D points the camera can detect is either known by design (e.g. flat retroreflective markers for the external system, glued to well-defined locations on a printed calibration grid marker for the camera), or can be estimated (see fig. 7.8) using a calibrated pointing device (cf. section 7.1).

In that case we might first solve the 3D-3D pose estimation problem to get T_{PE} and then compute the required transformation T_{HE} as a combination $T_{\mathsf{TH}}^{-1} T_{\mathsf{TP}} T_{\mathsf{PE}}$. Other solutions are possible as well [141].

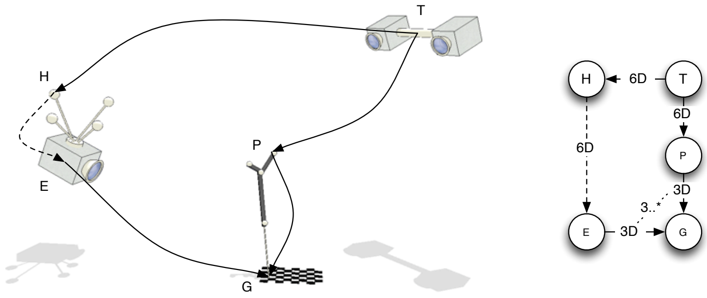


Fig. 7.8. Semi-direct solution: The coordinate systems are preregistered using a 3D-3D registration process with a calibrated pointer.

7.2.3 Indirect Solution

The most general solution to the problem is usually known in the literature as *hand-eye calibration algorithms*. Initial solutions to the problem in the robotics community used to employ multiparameter nonlinear optimization over the whole robot kinematics including the camera setup. This was not sufficient due to performance reasons and the need for a good initial guess [185]. The current state of the art is to solve the problem independent from the robot kinematic model.

From an abstract point of view we have two tracking systems T and E providing at any point in time the transformations $A_i = T_{\mathsf{TH}}$ and $B_i = T_{\mathsf{EP}}$, and we are looking for the relative transformations X and Y between their respective coordinate systems (cf. figure 7.9).

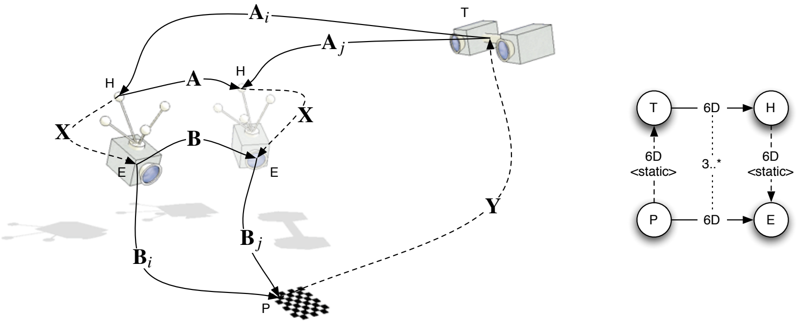


Fig. 7.9. Indirect computation of the hand-eye offset between a camera and an attached marker from an external tracking system. Setup and spatial relationship graph.

At every point in time, the two tracking systems provide measurements A_i and B_i for both transformations. To compute T_{HE} we use relative measurements between two poses by computing $A_{ij} = (A_i)^{-1}Aj$ and $B_{ij} = (B_i)^{-1}Bj$. Therefore we get for any arbitrary pair of measurements an equation with six degrees of freedom.

$$A_{ij}T_{\mathsf{HE}} = T_{\mathsf{HE}}B_{ij} \tag{7.4}$$

$$\iff \quad \begin{bmatrix} R_{A_{ij}} & t_{A_{ij}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} R_{\mathsf{HE}} & t_{\mathsf{HE}} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} R_{\mathsf{HE}} & t_{\mathsf{HE}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} R_{B_{ij}} & t_{B_{ij}} \\ \mathbf{0} & 1 \end{bmatrix} \tag{7.5}$$

We need at least two such equations [185, 34] to solve the system of equations, which is obviously not linear with respect to T_{HE}.

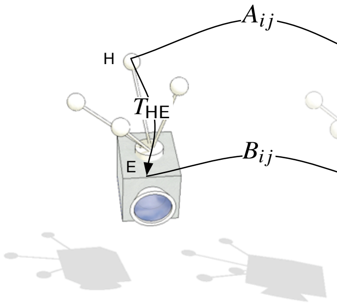


Fig. 7.10. The System of equations $A_{ij}T_{\mathsf{HE}} = T_{\mathsf{HE}}B_{ij}$ can be solved first for rotation, then for translation. At least two relative motions around different rotations axes are required

A Survey on Solving the Equation $AX = XB$

In 1987, Roger Tsai and Reimar Lenz [186, 185] proposed a solution to the general hand-eye calibration problem by first decomposing equation 7.5 into a rotational part of

$$R_{A_{ij}}R_{\mathsf{HE}} = R_{\mathsf{HE}}R_{B_{ij}}$$

and a translational part

$$(R_{A_{ij}} - I)t_{\mathsf{HE}} = R_{\mathsf{HE}}t_{B_{ij}} - t_{A_{ij}}$$

After first solving the rotational part for R_{HE}, they use this solution to find the translation t_{HE}.

To reach their goal, they represent the rotations by their main rotation axis $P_{A_{ij}}$ and $P_{B_{ij}}$ (which is in fact the vectorial part of the according quaternion) and find the linear least squares solution

$$P'_{\mathsf{HE}} = Skew(P_{A_{ij}} + P_{B_{ij}})^{+} \; (P_{C_{ij}} - P_{A_{ij}})$$

and after scaling

$$P_{\mathsf{HE}} = \frac{2P'_{\mathsf{HE}}}{\sqrt{1 + |P'_{\mathsf{HE}}|^2}}$$

which can be reconverted into a rotation matrix to get R_{HE}. Now this can be used to solve the translational part

$$t_{\mathsf{HE}} = (R_{A_{ij}} - I)^{+} \; (R_{\mathsf{HE}} t_{B_{ij}} - t_{A_{ij}})$$

again using a linear least-square technique.

In an independent work around the same time [163], Shiu and Ahmad present a closed-form solution to the problem, using two independent rotation pairs. Their approach was refined by Zhuang and Roth [231] by using again quaternion representation. Chen [34, 35] notes that under certain circumstances this closed-form solution does not provide unique solutions.

More recently, Daniilidis et.al. [44, 45] have presented a closed form solution based on dual quaternions (cf. section 4.1.5). Hirzinger et.al. [174] give a more detailed overview over the proposed methods together with a comparison of different algorithms based on simulation data.

Selection of Movement Pairs

It has been noted in the original work of Tsai and Lenz that not any pairs of the hand-eye poses give good numerical performance. In particular, when recording continuous data from two tracking systems instead of moving a robot arm to well-defined poses, the resulting small movements will not result in good movement pairs when used consecutively. The relative rotation axes of two movement pairs should be ideally vertical, while the relative rotation should be about 90° [186]. Additionally, the distance from the camera to the optical marker should be as small as possible – an obvious result of the error propagation in the pose inversion as seen in figure 5.19. The same applies to the relative translation t_{HE} in the hand-eye matrix, though this is usually given by the application.

Schmidt et.al. [156] present an efficient algorithms to select movement pairs automatically from a series of measurements. Shi et.al. present a similar algorithm [161] based again on the principal rules stated by Tsai and Lenz.

7.2.4 Iterative Solution

The hand-eye calibration problem can also be seen as a recursive matrix equation $X = A^{-1}XB$. Hirsh et.al. [81] have presented a solution based on this recursive definition. While the system does not converge for a single equation, they use a system of equations with one equation for each measurement, and average the value of X in every step using quaternion interpolation (cf. section 4.1.4). Using this method, the iterative estimation converges.

Another iterative solution was presented by Zhuang et.al. [230, 232, 233]. They provide a Jacobian relating the measurement residual errors to pose errors of the unknown matrix X. Based on this relationship, the system can be solved using nonlinear optimization. Additionally, this measurement Jacobian directly provides the possibility to apply linear error propagation here.

7.2.5 Error Propagation for the Hand-Eye Calibration

The error propagation during hand-eye calibration naturally depends on the algorithm that is used. If a direct computation (cf. section 7.2.1) is used, the error propagation from section 4.2.2 can be used. Aron et.al. [4] have given an error estimation for the rotation-only hand-eye calibration, which is the first step of the algorithms described in [185]. For simplification, they assume the vision tracking to be error-free. When using the translation as well, this is not a good assumption since the translational error in this case is rather large (cf. figure 5.19).

In this case we apply again the backward error propagation (section 3.4.2) by computing the Jacobian function J_f that maps the 6D pose error of the hand-eye transformation to the respective errors of the tracking systems, linearized at zero error [230, 211].

7.3 Tracker Alignment

Tracker alignment is the problem of aligning the base coordinate systems of two or more different tracking systems with each other [14]. This problem is quite similar to the problem of hand-eye calibration. When we want to use sensor fusion (cf. section 3.4.3) of two different tracking system, we need the tracking data given in the same coordinate system. Figure 7.11 shows an example of the fusion of an electromagnetic tracking system with the sensor mounted at the tip of a pointer, and an optical tracking system with the fiducials at the end of the same pointer. To apply the sensor fusion, the relative pose between the optical target and the magnetic sensor needs to be known.

Sometimes it is possible to use the same fiducials for both tracking systems [169]. The alignment of the two tracking systems is then equivalent to the direct computation of the hand-eye calibration (cf. section 7.2.1).

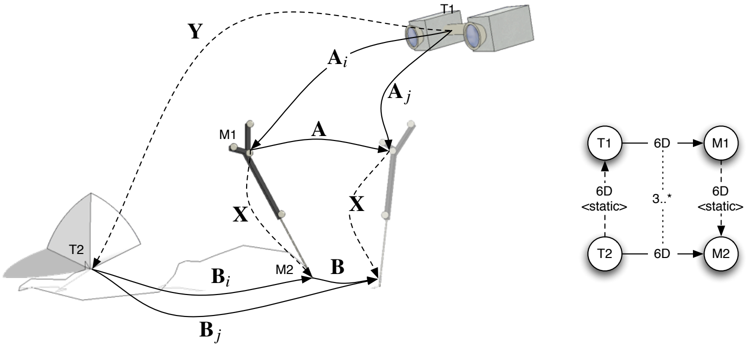


Fig. 7.11. Tracker alignment of two tracking systems tracking the same rigid object.

7.4 Optical See-through Head-mounted Display Calibration

For visual augmented reality, there are several different options for the display technology. We can classify the displays according to the location of the display relative to the user [25] in head-attached displays, hand-held displays and spatial displays. Head-attached displays either use optical see-through or video-feed-through [103, 147] technology. In addition, retinal displays [181, 195, 68] and head-mounted projectors have been proposed. Azuma et.al. [8] defined the tracking requirements for head-mounted reality.

The acceptance of an augmented reality system heavily depends on the correct alignment of virtual and real objects. To reach this goal we do not only need accurate tracking but also a correct determination of the parameters of the display system. The calibration of optical see-through displays always needs special attention, since in this case the augmented reality system has no access to an image of the reality. For calibration of such displays user input is therefore necessary, which introduces additional sources of error.

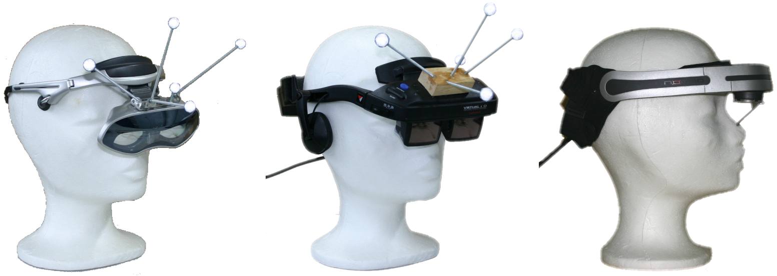
Since the emergence of optical see-through displays, several methods have been proposed for the determination of the projection parameters and the transformation from the display to the user's eye. At the moment, two methods are considered state of the art: The Single Point Active Alignment Method (SPAAM) and the Display Relative Calibration (DRC) which are both explained in more detail in the next section.

Tang et.al. [179] have already analyzed the performance of different variants of the SPAAM method. In this section we compare both methods with respect to their performance using over 1500 corresponding points from over 20 users and three different displays. Using different displays we can also draw conclusions independent from a single specific setup. The displays used for this study were the following three displays also shown in figure 7.12:

Sony Glasstron PLM-S700E This display with a resolution of 800 by 600 pixel is widely used and was also available in a stereo variant. However we used the mono display since we only analyzed monocular calibration. The second eye was covered by tape to avoid distraction.

Virtual i.O. i-glasses! This display uses a 640 by 480 resolution in which measuremets in this paper are made while only 320 by 240 pixel are actually displayed. Although this is a stereo display, we used again only a single eye for the calibration study.

Microvision Nomad This display is actually a 800 by 600 monochrome virtual retinal display which draws images by scanning low power laser light directly onto the retina.



Glasstron i-glasses Nomad

Fig. 7.12. Optical see-through HMDs

Similar products from competitors exist that have higher resolution and full color but were unfortunately not available for our studies.

7.4.1 Current Calibration Methods

Early work in the field of optical see-through head-mounted display calibration usually included the task of aligning some virtual object displayed in the HMD with its real counterpart. The alignment is achieved by holding the head still and manipulating the virtual object for example with the keyboard [90] or mouse [99]. The main problem with these approaches is the fact that it is not easy for novice users to align three-dimensional objects with its virtual counterparts and additionally holding the head still during such an alignment procedure.

SPAAM and EasySPAAM

The *Single Point Active Alignment Method (SPAAM)* [187] and its improvement, *EasySPAAM* [71] aim at making this calibration procedure more usable.

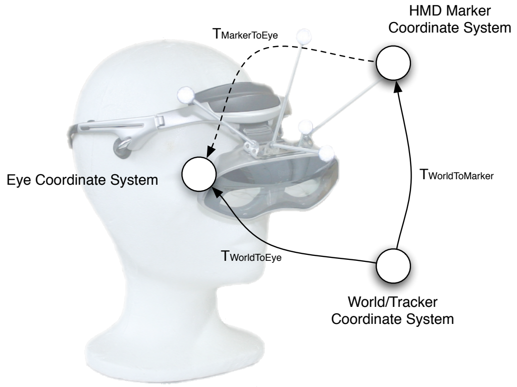


Fig. 7.13. Relevant coordinate systems and transformations for SPAAM calibration

This method estimates the projection matrix $\mathbf{P}$ of the HMD directly from a set of 3D-2D corrrespondences which are collected one after the other by a simple interaction procedure: A virtual crosshair is shown to the user wearing the HMD. The user then has to align this crosshair with a known point on a tracked object. As an easy model for the head-mounted display projection we can again use the pinhole camera model:

$$\rho \mathbf{X}_{image} = \mathbf{P}\,\mathbf{X}_{marker}$$

where

$$\mathbf{X}_{marker} = \mathbf{T}_{WorldToMarker}\,\mathbf{X}_{world}$$

Figure 7.4.1 shows the relevant coordinate systems and tranformations needed for this procedure.

If a sufficiant number of correspondences is collected, the least square solution for the transformation P can be computed directly using singular value decomposition [187]. This method can be extended to stereo displays as well [70, 189].

The *EasySPAAM* [71] method is an extension of the general method where the whole process is split into two phases. In the first phase, which is done once, the transformation from the tracking marker to the display coordinate system as well as the intrinsic parameters are determined as above. In the second phase, these transformations are only linearily shifted and scaled to match the users eye position using few point correspondences. The user can choose the initial calibration from several precomputed calibrations. Although this seems to be rather imprecise, the accuracy of the results approved this procedure.

Display Relative Calibration (DRC)

More recently, Owen et al. [133] proposed the *display-relative calibration (DRC)* method. This method is again a two stage method and aims at reducing the amount of user interaction by an offline calibration step done under laboratory conditions. In this stage the intrinsic camera parameters are estimated relative to the display coordinate system and the transformation between the display relative and the eye coordinate system is then adjusted in the second phase online (figure 7.4.1).

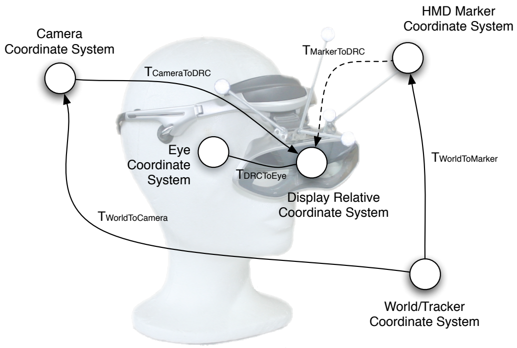


Fig. 7.14. Relevant coordinate systems and transformations for DRC calibration

The first phase does not require any user interaction; here the human eye is replaced by a digital camera. First the camera is calibrated using several planar fiducials on two calibration boards. After that, the display plane of the HMD is calibrated using this calibrated camera and fiducial markers displayed on the HMD screen. These fiducials are captured by the calibrated camera, which is moved horizontally and parallel to the image plane to different locations at about 3 millimeter distance, as seen in figure 7.15.

The corresponding points found in this stage are used to define the virtual display plane of the display. In the standard pinhole model this is assumed to be a flat plane, but for DRC a parabolic display plane is used to cover additional distortion errors. The resulting display planes for our displays are shown in figure 7.16.

During the online phase, only the exact center of projection has to be estimated. This is dependent from the position of the eye relative to the display coordinate system and therefore requires user interaction. Several methods were proposed to achieve this alignment, for example measuring the interpupilatory distance or a manual adjustment by the user; for our experiments, we used the collection of point correspondences similar to the second phase of

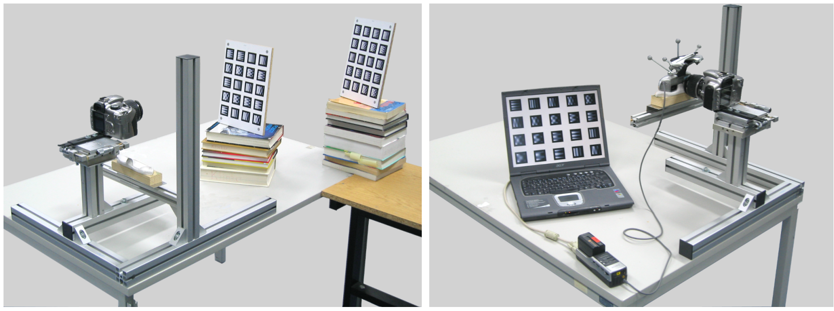


Fig. 7.15. Setup for the DRC Off-Line Phase: Calibration of the intrinsic camera parameters (left) and calibration of the display plane by taking several pictures with the HMD at a fixed position and the camera moved horizontally (right).

EasySPAAM. Using these correspondences, the eye position can be adjusted in three dimensions.

7.4.2 Comparative Evaluation

According to [13], the main sources for static registration errors are distortion in the HMD optics, mechanical misalignments in the HMD, errors in the tracking system and incorrect intrinsic and extrinsic parameters. The aim of HMD calibration is to estimate the intrinsic and extrinsic parameters; therefore it is necessary to use the same HMD and tracking system (and if possible also the same measurements) to compare only the influence of the calibration algorithm on the final result while all other parameters stay fixed.

Approaches for the Evaluation of Optical See-Through Calibration

Evaluating the accuracy of the calibration has because of the missing ground truth the same problems as the calibration itself. Since there is no digital image of the result available, again user interaction is needed for the evaluation. We present different approaches [79] for the evaluation in the following sections.

User Feedback

The simplest way of evaluating the accuracy is getting feedback from the user itself. However, problems here are not only the lack of quantitative values for the accuracy but primarily the subjectivity of the answers which are neither reproducable nor comparable.

However, some conclusions can be drawn even from user feedback, therefore we gathered qualitative user feedback for the calibrations in addition to our evaluation.

Video Camera

To circumvent the problem of not being able to access the augmented image, a video camera can be used which is placed at the estimated eye position [187, 70]. It is utilized during both the calibration and the evaluation process. Thus, a digital augmented image is obtained which can be analyzed for registration errors, measured in pixels. This method depends on a rather complicated setup which requires the camera being attached rigidly to the head-mounted display. Furthermore, it is questionable if the results really reflect the situation of a human using the augmented reality system since a video camera is only an approximation of the eye.

Evaluation Board

To get at least some quantitative number, a tracked evaluation board can be used where the user is asked to point the location where he percieves the virtual object [117, 126]. This can be done either by a tracked pointing device or by displaying a virtual mouse pointer in the head-mounted display. Note that this method could easily be integrated into the user's daily work, e.g. by using the buttons in his application software as target points. Thus, the AR system could be capable of providing feedback concerning the calibration accuracy, of warning the user if the accuracy drops below a certain threshold or even of utilizing the user inputs for calibration purposes.

Using Point Correspondences

The simplest but most effective way of getting user feedback is to collect again single point correspondences as desribed for SPAAM or the online phase of DRC. To determine pixel error, the 3D point in marker coordinates is projected onto its corresponding 2D location and the distance to the actual 2D crosshair location is computed. For metric errors in the real world, the inverse of the projection matrix is used to map 2D points into 3D. This gives a line in space which can be used to get the angular error in space or a metric error in terms of "millimeters per meter" at a certain distance.

A major advantage of this evaluation method is its simple user interaction. Moreover, the procedure of collecting point correspondences is exactly the same which the user encountered during the calibration phase with SPAAM, EasySPAAM or DRC and thus, he is accustomed to it. Even more important for comparing different calibration methods, however, is that it is possible to let the user collect many correspondences and to decide afterwards how to use them. For example, subsets with varying numbers of correspondences can be utilized for calibrating the HMD. This makes it possible not only to compare different calibration methods, but to draw conclusions concerning the required number of correspondences to reach the desired accuracy.

Evaluation Results

To be able to compare the calibration methods SPAAM, EasySPAAM and DRC we first did a DRC off-line calibration followed by a user study to collect the necessary point correspondences and after that an off-line analysis of the collected data [79].

Hardware Setup

To track the HMDs and the calibration objects we used an infrared optical tracking system with retroreflective marker targets, produced by Advanced Realtime Tracking GmbH, more specifically a three-camera setup looking at the calibration area from three different sides to get almost isomorphic error for the detection of the markers in space.

In the HMD we displayed a crosshair with concentric circles that had to be aligned with a well-defined point on an also tracked calibration object held by the user. Once alignment has been achieved, the user had to press a button with his other hand. The points were displayed on a regular grid in the HMD, but with random order. The collected point correspondences then were used for the various calibration methods as well as for the evaluation of the accuracy as described in 7.4.2.

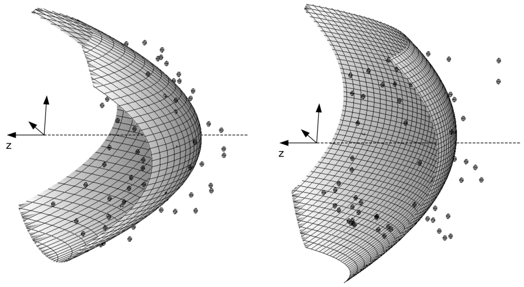
DRC Off-line Phase

For the DRC offline calibration we used a Canon EOS 300D Digital camera (due to size reasons the i-glasses were calibrated using a smaller Canon Digital Ixus II camera). Although it showed to be not too easy to gather the required number of pictures with all markers visible, finally the intrinsic and extrinsic parameters of the camera/display system could be determined: the focal length f in pixels, the projection center (c_x, c_y) and pixel aspect ratio s are displayed in table 7.4.2 [79]. From the corresponding points we additionally were able to determine the virtual display plane depth z_0 in meters, the spherical aberration parameter κ_2, the HMD apex (a_x, a_y) and the residual r from the optimization process.

	f [px]	(c_x, c_y) [px]	s	z_0 [m]	κ_2	a_x [m]	a_y [m]	r [m]
Glasstron	1745	(466, 409)	0.9803	-1.35	11.25	0.0078	-0.0238	0.015
i-glasses	1404	(371, 238)	1.0325	-2.16	9.64	0.0484	-0.0721	0.172

Table 7.1. Intrinsic parameters as a result of the DRC off-line phase

The computed virtual display planes of the Glasstron (a) and the i-glasses (b) are shown in figure 7.16. For the Nomad display, we were unfortunately not able to generate the required number of at least five pictures to perform the analysis. This was mainly due to the rather small viewing angle of the Nomad



(a) Glasstron (b) i-glasses

Fig. 7.16. Display planes as computed in the off-line phase of DRC

display, which did only allow images at a maximum horizontal distance of about 8 millimeters. For this reason, we had to exclude the Nomad display from our further analysis.

User Study

For further analysis, a user study with 14 participants was conducted [79]. Each user had to collect 25 point correspondences with a distance to the calibration object of 40 to 60cm. The points displayed in the HMD were distributed on a regular 5 × 5 grid, but were shown in a random order. To obtain the depth information required for the SPAAM calibration, the user was asked to capture another 25 correspondences at a distance of 90 to 110cm. The process was then repeated for the other displays.

After completing the point correspondences we asked for qualitative user feedback: We displayed a virtual overlay over the calibration object using different projection matrices calculated with the various calibration methods and let the user rate the quality on a scale of 1 to 6.

As mentioned above, calibration of optical see-through displays is highly user-specific due to different ways of wearing the HMD as well as varying anatomical properties such as the interpupillary distance. These differences are supposed to be reflected by the intrinsic and extrinsic parameters determined by the calibration procedure. In order to examine this, all 50 point correspondences of each user were utilized to compute the respective projection matrix. Since the correspondences are guaranteed to contain varying depth information, the resulting matrices should contain the actual parameters. Each matrix was then decomposed yielding the intrinsic and extrinsic parameters. Thus, the user-specific parameters f (focal length), (c_x, c_y) (projection center), s_x (pixel aspect ratio), s (skew) and (t_x, t_y, t_z) (eye position) shown in table 7.2 were obtained.

	Glasstron				i-glasses			
	min	mean	max	σ	min	mean	max	σ
f [px]	1679	1714	1756	20.7	1353	1390	1432	18.4
c_x [px]	298	405	504	66.3	294	368	485	45.9
c_y [px]	244	368	605	82.4	159	247	394	65.6
s_x	0.955	0.976	0.996	0.01	1.021	1.036	1.045	0.007
s	-24.5	2.3	43.1	17.2	-16.6	-0.9	13.8	10.7
t_x [m]	-0.159	-0.149	-0.141	0.004	-0.049	-0.044	-0.040	0.003
t_y [m]	0.090	0.095	0.101	0.004	0.099	0.107	0.116	0.004
t_z [m]	-0.079	-0.059	-0.042	0.012	-0.086	-0.074	-0.062	0.007

Table 7.2. SPAAM Calibration parameters of different users, using all point correspondences

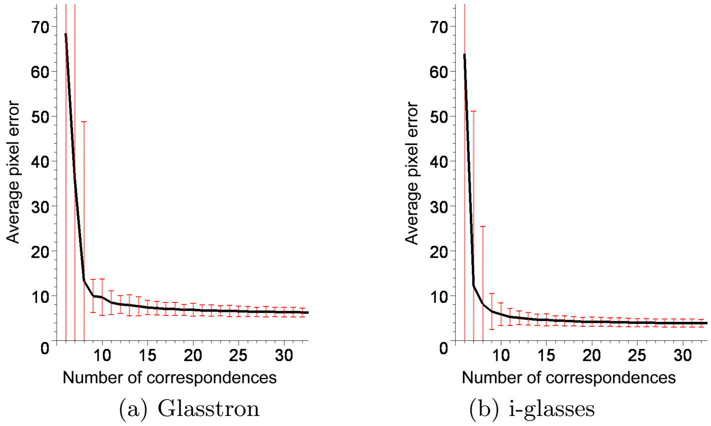
Comparison of SPAAM, EasySPAAM and DRC

To determine if a user-specific calibration is really necessary, a default SPAAM calibration was conducted by a single user by carefully collecting 50 point correspondences at varying depths.

As shown in [187], at least six point correspondences are required for a SPAAM calibration. In practice, more correspondences need to be collected in order to obtain more robust results. To evaluate how many actually are required and sufficient, different SPAAM calibrations for each user were computed out of randomly chosen subsets of the respective correspondences with varying size. The chosen correspondences were guaranteed to contain different depth information so that high-quality SPAAM calibrations were obtained. Furthermore, this procedure was repeated 30 times for each user and each subset size in order to compensate for noisy data and outliers. The results are depicted in figure 7.17 which shows both the average pixel error and the standard deviation with increasing number of correspondences.

Obviously, less than nine point correspondences are not sufficient to obtain an acceptable accuracy and robustness. Although the pixel error continuously decreases, no significant improvements can be expected with more than 20 correspondences. However, the decreasing variance shows that more correspondences contribute to more robust calibrations. Thus, the number of point correspondences has to be chosen according to the application's requirements in terms of accuracy and robustness as well as to the effort which is reasonable for the users.

To eliminate the influence of erroneous default calibrations and thus to obtain reliable facts concerning the accuracy and effectiveness of EasySPAAM and DRC, the computations described before were repeated with different default calibrations as starting point. For that, the default SPAAM and DRC calibrations were used as a starting point for both an EasySPAAM and a DRC recalibration. In summary, the following four calibrations were evaluated:



(a) Glasstron (b) i-glasses

Fig. 7.17. Average pixel error and standard deviation of SPAAM calibrations

- EasySPAAM starting with SPAAM calibration
- EasySPAAM with DRC off-line phase parameters
- DRC with a default SPAAM calibration
- DRC with DRC off-line phase parameters

These computations are possible since the DRC parameters can be combined to a projection matrix which can be used as a default calibration for EasySPAAM and since a SPAAM matrix can be decomposed to the intrinsic and extrinsic parameters which in turn can be applied during the DRC on-line phase. For those parameters which are not included in a SPAAM matrix such as the display plane depth and curvature, the respective default values of the

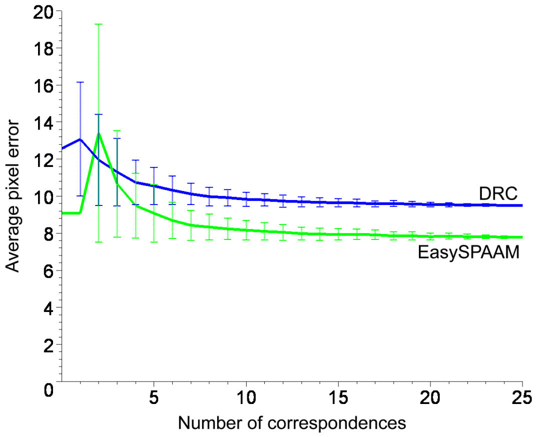
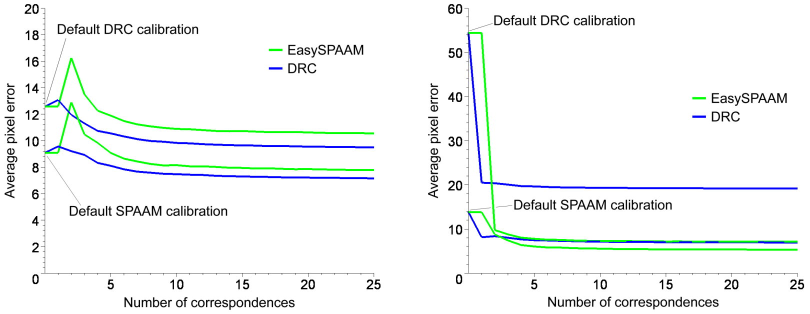


Fig. 7.18. Average pixel error and standard deviation of EasySPAAM and DRC

DRC calibration are used. The same procedure as described before then yields the results shown in figure 7.4.2.



(a) Glasstron (b) i-glasses

Fig. 7.19. Average pixel error of EasySPAAM and DRC for different default calibrations

Most striking in these diagrams are the contrasting results concerning the Glasstron and the i-glasses. While the DRC method consistently yields higher accuracy for the Glasstron, it is the opposite way round for the i-glasses. While this seems to be contradictory at a first glance, it can be explained by considering the results of a prior DRC off-line calibration of the Glasstron and an informal user study with less participants conducted earlier [79]. There, the same computations as described above yielded a consistently higher pixel error for DRC. The explanation for this varying behavior of the display-relative calibration is supposed to be found in the parameters determined in the off-line phase. While the conventional intrinsic and extrinsic parameters of both DRC calibrations more or less coincide, the values describing the virtual display plane are different, especially the display plane depth z_0 (-1.35 vs. -1.88). Although both calibrations were conducted as carefully as possible, the results show a significant variance which obviously influences the accuracy compared to the more robust EasySPAAM method.

7.4.3 Conclusions from the Experiments

In this section we present the conclusions that we drew from our experiments [79] to help other researchers make their choice in what kind of calibration to choose for their application.

Necessity of User-Specific Calibration

The most important result of our work was the observation that it is in fact necessary to calibrate optical see-through HMDs for a specific user at any usage. First of all due to different wearing positions the actual center of projection with respect to the tracking system varied in our experiments by up to 3.7cm in the z direction.

Additionally, a qualitative feedback during the user study suggested that the users were consistently more satisfied with their own calibration than with some carefully conducted default calibration.

	Glasstron		i-glasses	
	mean	σ	mean	σ
Default SPAAM	2.1	0.9	3.1	0.7
User-specific SPAAM	1.8	0.8	1.6	0.6
Default DRC	2.8	1.6	5.5	0.7
User-specific DRC	2.0	1.0	2.7	1.3

Table 7.3. Summary of the user feedback (marks 1–6) for different calibrations

However, in real world applications the static registration error might not be the biggest source of errors and may be shadowed completely by dynamic errors due to lag or by other kinds of calibration or registration errors. In such cases it still can be feasible to use one default calibration instead of user specific calibrations. The additional effort of user specific calibration would not lead to visible improvements.

How to Collect the Point Correspondences

One of the key factors for a good calibration is to collect corresponding points with different distances to the calibration point, in particular at the distances in which objects are displayed later on during the application. This should be enforced by the calibration software. Otherwise the calibration will only provide acceptable results at the depth where the point correspondences were collected.

Another limitation on the quality of calibration is the fact that head pointing might not be suited as input method [179] since it is not capable of providing the required level of precision due to physical limitations.

Theoretically the calibration under laboratory conditions with additional online user adjustment can be regarded as a good idea. In practice however we had the problem that the attached markers could be twisted easily by accident, which produced a significantly increased error. Therefore, due to the fact that the used tracking system is an integral part of the calibration procedure, it is important to have an easy way of performing the calibration if such influences can not be avoided.

SPAAM is More Robust than DRC

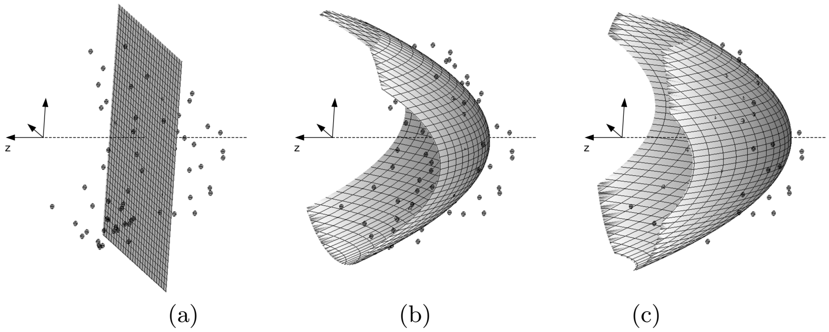
As shown before, the EasySPAAM method seems to be more robust to errors in the measurements than the DRC method. Furthermore the DRC calibration requires a more or less complex setup for the offline phase, while the EasySPAAM method uses only point correspondences which are needed for the second phase of DRC as well. Using one of the simpler methods to gather the data for the second phase as proposed in [133] does not seem to be a good idea [79].

Modifications of DRC

To justify the additional overhead of the DRC offline phase, we tried to analyze some modifications to the originally proposed method to gain additional accuracy. We have analyzed [79] several modifications of the DRC algorithm and their influences to the registration accuracy.

Different Image Plane Models

In particular we tried to fit different plane models to the data in the off-line phase: The radially symmetric parabolic as proposed by Owen, a flat plane, and a more general parabolic with different curvature in x and y direction, as seen on figure 7.20.



(a) (b) (c)

Fig. 7.20. Visualization of different display plane models (Glasstron): Radially symmetric parabolic (b), flat plane (a), and general parabolic (c)

We then evaluated the accuracy again with random subsets of the users correspondences. Surprisingly, as seen on figure 7.21, the error for the most simple model was smallest for few correspondences in the online phase and converged to the same error with more correspondences. The error for the general parabolic and for the radially symmetric one were quite similar due to almost identical parameters.

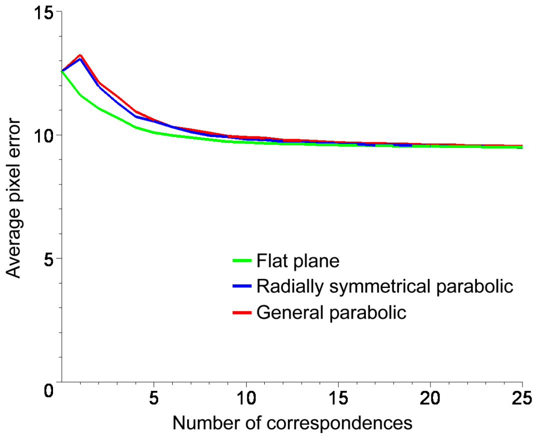


Fig. 7.21. Average pixel error of DRC with different display plane models

Nonlinear Mapping

Although the DRC calibration uses a nonlinear display plane, in the end for the point correspondences and for the final augmentation a projection matrix and therefore a linear mapping is used. We analyzed the influence of this obvious mismatch and developed in theory a model for nonlinear projection to compensate for the distortion as perceived by the user, neglecting the problem of an efficent implementation for this mapping at runtime. Anyways, although this would be theoretically justified, the experiments did not result in a significant increase in accuracy.

However, there might be room for improvements here. While the display-relative calibration is unique in characterizing the virtual display plane, this information is not taken into consideration consistently. Since the intrinsic and extrinsic parameters are based on the pinhole camera model and are determined with Tsai's camera calibration algorithm in the off-line phase, a flat display plane is assumed first. The computation of the display plane as a parabolic then leads to a mismatch which requires several compromises in the further calibration process. In order to exploit the advantages of estimating the shape and location of the virtual display plane, the implied nonlinearity should be considered throughout the whole calibration procedure. As a starting point, a modified pinhole camera model which incorporates the curvature of the display plane could be utilized. Finally, the effects of such modifications have to be analyzed empirically to be able to justify these theoretical considerations.

Part III

Applications

8

Applications

In this chapter we describe some applications of the theory presented in this thesis. We start with an example from an industrial scenario where virtual objects should be graphically overlaid in a video camera image of the real objects. Then we analyze in a medical context the accuracy of a tracked laparoscopic camera, which is a common setup in medical augmented reality. Finally we show a visualization tool that can be used to explore the errors in optical tracking in an interactive way.

8.1 Camera Augmentation in an Industrial Application

Augmentations of video images are common in augmented reality systems. In the following example, we analyze the tracking requirements for an industrial augmented reality application [129]. In a car manufacturing company, virtual CAD models of planned car parts should be augmented onto existing parts of the car (see figure 8.1). This can be used in an early design phase to evaluate prototypical designs. To perform the overlay, a camera with an attached tracking target is tracked in a four-camera setup. The camera views the real scene in a distance of one meter.

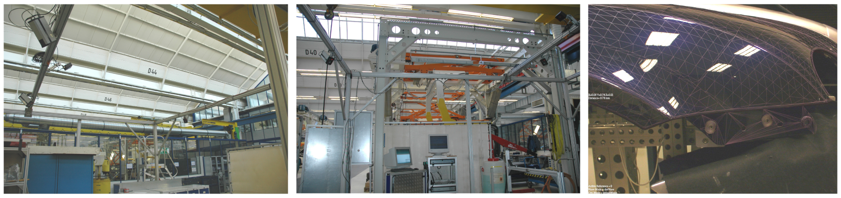


Fig. 8.1. Tracking setup with four cameras in an industrial environment (left, middle). Virtual overlay on a real car part (right).

The specifications from the manufacturer require a maximum error of 1 mm in space at the real car part. This corresponds to an angular error of about $\tan(10^{-3}) = 0.0572958°$. At a 75% confidence level (see table 3.1), this requires a standard deviation of 0.0139462° for the angle.

To reach the specified accuracy in a setup where the camera is tracked with an attached marker, in a typical setup with a fiducial location error of about 0.1 mm and a four-fiducial target in tetrahedral form, this would require a diameter of about 60 cm for the fiducial distribution. Figure 8.2 shows the expected accuracy for different diameters.

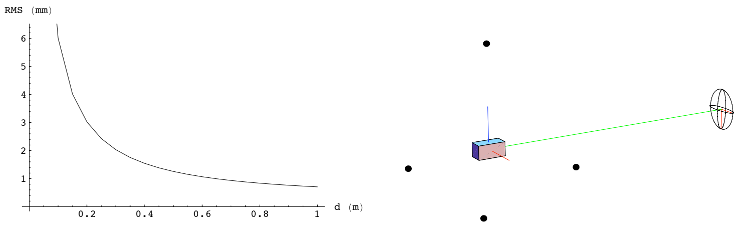


Fig. 8.2. Expected accuracy (75% confidence level) at 1 m distance in an outside-in configuration (ellipsoid magnified by a factor of 50 for visualization)

The position of the real object which needs to be overlaid with the virtual graphics also has to be known; if this is done using another tracking target or by point-based registration (cf. section 7.1), the error during registration adds to the final augmentation error. The same applies for the error in the hand-eye calibration (cf. section 7.2) for the tracked camera. This increases the resulting error even more. Both calibration errors however are fixed after calibration and therefore are systematic errors. They can be reduced by diligent re-calibration if necessary.

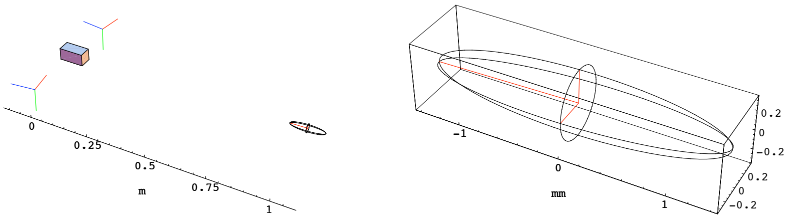


Fig. 8.3. Expected accuracy (75% confidence level) at 1 m distance in an inside-out configuration with a video camera mounted rigidly to a stereo camera system.

When we consider the results from section 6.3, an inside-out configuration might be a better choice. For this configuration, we mount the video camera rigidly as close as possible to the baseline of a stereo camera tracking system. This system is used to track the real object using a tracking target mounted onto the object close to the actual point of interest. Figure 8.3 shows that the expected accuracy is not very good in the depth direction, but on the camera projection we perceive only the smaller directions of the error. While this setup is not a good choice for general measurement applications due to the larger depth error, it is suitable for the camera augmentation example. The error propagation from the tracked target to the point of interest is sufficiently small when the target is close enough to the point of interest.

This example shows how a systematic analysis of both the requirements and the potential solutions can lead to surprising results. Using the methods presented in this thesis, we can in advance evaluate the expected tracking accuracy of different tracking setups and choose the optimal solution.

8.2 Laparoscope Tracking Accuracy

In medical applications, accuracy considerations are of special importance for safety reasons. One common medical application is the augmentation of endoscopic images [213, 212, 196, 197, 156, 159, 160]. We have evaluated the design of tracking targets for laparoscope augmentation [55]. A laparoscope is a telescopic rod lens connected to a video camera to perform minimally invasive surgeries. Figure 8.4 shows a schematic drawing and an image of an actual laparoscope. While the camera is outside the patient body during an operation, the optical center of the camera is at the tip of the rod and therefore inside the patient.

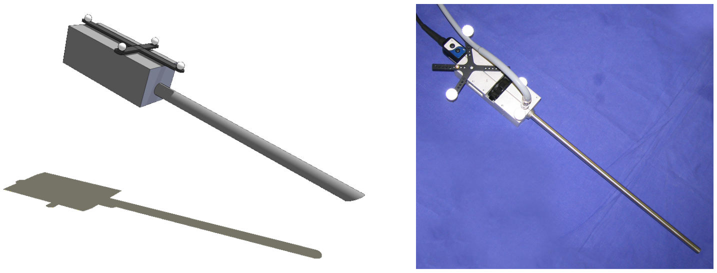


Fig. 8.4. Schematic drawing of an endoscope with tracked fiducials (left). The point of interest is about 30 centimeter away from the centroid of the fiducial distribution. Image of a real endoscope with attached prototypical target (right)

The instrument is tracked to enable augmentation of preoperative planning data onto the image plane of the laparoscope. This involves additionally a

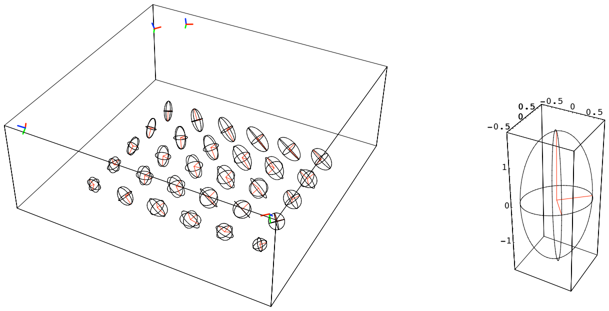


Fig. 8.5. Four-camera setup in the lab with covariance ellipsoids (left) at a height of one meter inside the tracking volume of all four cameras. In the center we have an expected RMS error of about 0.5 mm (right) with significant anisotropy.

registration of the planning data with the patient, the tracking of the patient, camera calibration of the laparoscope and the hand-eye calibration of the laparoscope camera with the tracked fiducials on the instrument. While all these steps might involve errors as well, and all theses errors contribute to the final application error, we focus only on the accuracy of the instrument tracking itself.

The laparoscope is tracked by a four-camera tracking system as shown in figure 8.5. The expected fiducial location error shows significant anisotropic behavior; we will analyze the effect of this anisotropy later as well. The expected fiducial location error in the center of the tracking volume is also shown in figure 8.5. The optical tracking fiducials need to stay visible and outside the body, therefore they are mounted about 30 centimeters away from the optical center of the camera, which is located approximately in the tip of the instrument. The exact location needs to be determined using the hand-eye calibration (section 7.2)

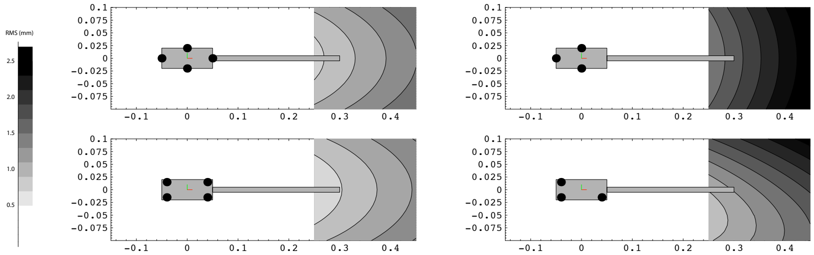


Fig. 8.6. Statistically expected tracking error (RMS) in an instrument's field of view (left) with occlusion of single fiducials (right)

We compute the statistically expected tracking error [218] in the instrument's field of view for a specific orientation of the instrument, in this case oriented horizontally. The spatial distribution of tracking error is shown in figure 8.6 for four different fiducial configurations.

On the right hand side of the figure, the effect of occluded fiducials can be seen. While in the top row the accuracy decreases a lot when the frontmost fiducial is occluded, in the bottom row the accuracy remains acceptable. In particular in crowded operating rooms, an occlusion of single fiducials is rather likely and has to be considered in advance when designing the setup.

Additionally it can be seen that the levels of constant RMS error actually form ellipsoids around the centroid of the fiducial configuration, with the main axes of the ellipsoid oriented proportional to the main axes of the fiducial distribution.

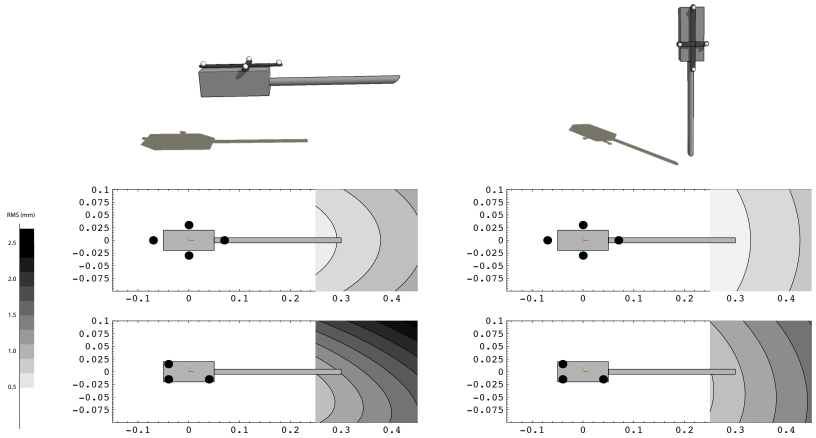


Fig. 8.7. The expected tracking error changes under rotation of the instrument due to the anisotropy in the fiducial location error. Left column shows the expected error for different fiducial distributions when the instrument is held horizontally, right column shows the error when the instrument is oriented vertically.

Figure 8.7 shows the change in the expected error due to the anisotropy of the fiducial location error. The fiducial location error is larger in the vertical direction (see figure 8.5). The fiducial configurations in this case are flat configurations (cf. section 6.1.6). This results in a higher rotational error when the instrument is oriented horizontally, and due to the error propagation to the tip of the instrument, the positional error at the tip is larger.

Figure 8.7 shows the error as a RMS value in the field of view of the camera. This is useful to evaluate the error from the instrument tracking alone. For an actual endoscope augmentation application [55] we would further need to propagate the error through the laparoscope camera projection to get a two-dimensional error on the image plane of the camera, since this overlay error should be minimized in the end.

8.3 Interactive Visualization of Tracking Errors

During the development of an augmented reality application it is often useful to get an impression of the tracking errors in an intuitive way. We have developed and presented [18] an application for the interactive exploration of the tracking errors in a running application.

The tool can be used as an add-on to any existing optical tracking application and provides online visualization of the current error characteristics. Figure 8.8 shows a screenshot of the running application.

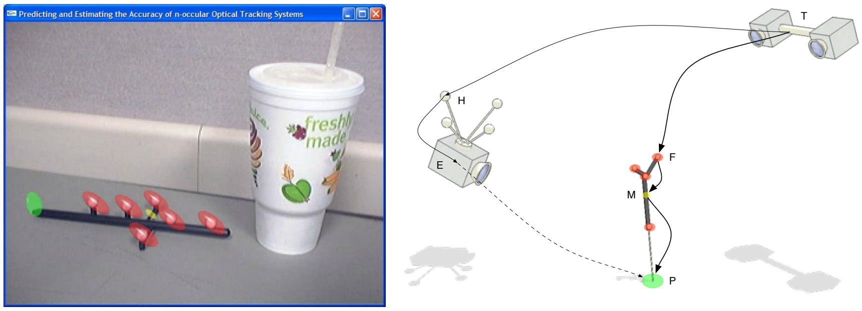
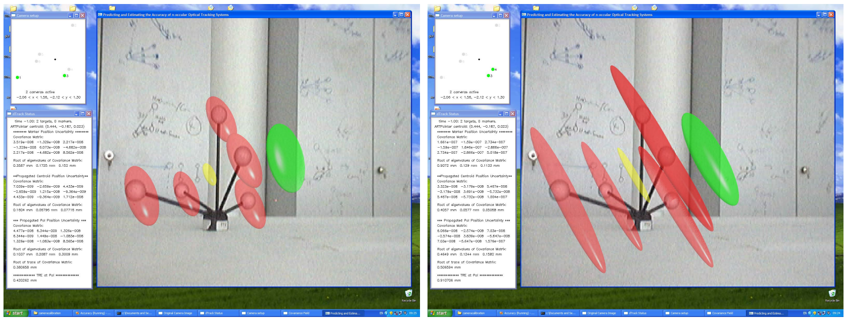

Predicting and Estimating the Accuracy of n-occular Optical Tracking Systems
T
H
E
F
M
P

Fig. 8.8. Augmented Reality visualization of the error propagation in a pointing device. Fiducial location errors (red), marker target error (yellow), and target registration error (green)

The system uses augmented reality overlays on the image of a tracked webcam to show the magnitude of the positional fiducial location errors (red) together with the propagated marker target error (yellow) in the centroid of the target and the target registration error (green) at the point of interest. The errors are displayed as covariance ellipsoids with a user-defined uniform scaling to make them visible even when they are small.

The extrinsic camera parameters, the target geometry and the location of the point of interest are imported from configuration files and can be adopted to any tracking application. The tool receives the pose of the pointer M as well as the pose of the target H mounted onto the webcam E from the tracking system. The hand-eye calibration for the tracked webcam is performed in advance using a second tool implementing the algorithms described in section 7.2 [185].

It is important to note that we display only the prediction for the errors from the tracking system to the point of interest. This is the error of



```
2 cameras active
-2.06 < x < 1.56, -2.12 < y < 1.30

time -1.00: 2 targets, 0 markers.
ARTPointer centroid: (0.444, -0.167, 0.023)
******** Marker Position Uncertainty ********
Covariance Matrix:
3.519e-008  -1.329e-008  2.217e-008
-1.329e-008  6.073e-008  -4.682e-008
2.217e-008  -4.682e-008  8.562e-008

Root of eigenvalues of Covariance Matrix:
0.3587 mm  0.1725 mm  0.152 mm

**Propagated Centroid Position Uncertainty**
Covariance Matrix:
7.039e-009  -2.659e-009  4.433e-009
-2.659e-009  1.215e-008  -9.364e-009
4.433e-009  -9.364e-009  1.712e-008

Root of eigenvalues of Covariance Matrix:
0.1604 mm  0.08796 mm  0.07716 mm

*** Propagated PoI Position Uncertainty ***
Covariance Matrix:
4.477e-008  6.344e-009  1.326e-008
6.344e-009  1.449e-008  -1.083e-008
1.326e-008  -1.083e-008  8.565e-008

Root of eigenvalues of Covariance Matrix:
0.1037 mm  0.2087 mm  0.3009 mm

Root of trace of Covariance Matrix:
0.380658 mm

************* TRE at PoI *************
0.420262 mm
```

```
2 cameras active
-2.06 < x < 1.56, -2.12 < y < 1.30

time -1.00: 2 targets, 0 markers.
ARTPointer centroid: (0.444, -0.167, 0.023)
******** Marker Position Uncertainty ********
Covariance Matrix:
1.661e-007  -1.59e-007  2.734e-007
-1.59e-007  1.846e-007  -2.866e-007
2.734e-007  -2.866e-007  5.018e-007

Root of eigenvalues of Covariance Matrix:
0.9072 mm  0.129 mm  0.1133 mm

**Propagated Centroid Position Uncertainty**
Covariance Matrix:
3.323e-008  -3.179e-008  5.467e-008
-3.179e-008  3.691e-008  -5.732e-008
5.467e-008  -5.732e-008  1.004e-007

Root of eigenvalues of Covariance Matrix:
0.4057 mm  0.0577 mm  0.05068 mm

*** Propagated PoI Position Uncertainty ***
Covariance Matrix:
6.066e-008  -2.574e-008  7.03e-008
-2.574e-008  3.839e-008  -5.647e-008
7.03e-008  -5.647e-008  1.576e-007

Root of eigenvalues of Covariance Matrix:
0.4649 mm  0.1244 mm  0.1582 mm

Root of trace of Covariance Matrix:
0.506594 mm

************* TRE at PoI *************
0.910706 mm
```

Fig. 8.9. Visualization of the error for different (hypothetic) camera setups. Difference between large baseline (left) and small baseline (right)

the transformation T_{TP} in figure 8.8. What we actually see on the camera image as augmentation error is the propagated combined error of the edges $T_{\mathsf{PE}}T_{\mathsf{EH}}T_{\mathsf{TH}}^{-1}T_{\mathsf{TP}}$ (cf. section 4.2.2), which is usually by orders of magnitude larger than the predicted error.

Besides this graphical output we have implemented also a display of the relevant covariance matrices, the Eigenvalues,, the RMS errors, and for comparison an estimate of the TRE following the point based registration error method by Fitzpatrick et al. [60].

Since the tracking camera setup is loaded from a configuration file, we can use the same tool also as an evaluation tool for hypothetic camera setups. Figure 8.9 shows two screenshots of the same target using two different tracking camera setups, one with large baseline between the cameras and another one with small baseline. In a future version we plan to receive and adopt the error estimation to additional runtime information from the tracking system

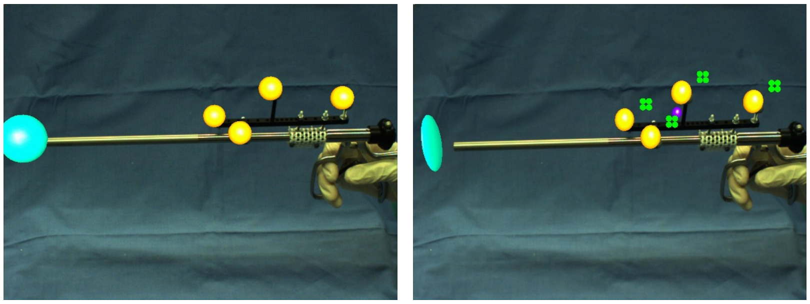


Fig. 8.10. Estimation of the accuracy at a point of interest using target RMS [60] (left) and estimation of covariance using fiducial covariance [19] (right) — orange: covariance of fiducial, cyan: covariance at point of interest, purple: covariance at center of fiducials

on the number of markers and cameras used for a particular measurement. This will enable the user to interactively hide single markers from the view of single cameras and observe the decrease in accuracy instantaneously in the visualization. So far this information is usually not available from outside the tracking system software.

8.4 Integration of visibility Information

In a prototypical implementation [168], we have integrated visibility information from an actual tracking system into our error estimation. The tracking software provides information about the geometric setup of the tracking cameras, and at runtime the full visibility information for every camera and every fiducial. Using this information, we can now compute individual error covariances for the fiducial location error (cf. section 5.3.2) and use these covariances in the estimation of the marker target error (cf. section 5.3.3) and the target registration error (cf. section 5.3.5).

Figure 8.10 shows a comparison of the estimated accuracy using the target RMS method [60] and our proposed method. The most obvious difference is the spherical error covariance in the target RMS error method, resulting from the one-dimensional RMS error estimate. Our method shows a flat disc resulting from the propagation of the angular error as described in section 4.2.2.

Figure 8.11 shows the effect of occluded cameras onto the estimated error. For every fiducial, we show the visibility by each of the four cameras in our setup by a red or green disk besides the fiducial. Occluding cameras increases the estimated error, and additionally changes the general shape of the error.

Note that the difference in the errors is not as significant as in figure 8.9, since in this example we chose a camera setup which provides a good baseline (cf. section 5.3.2) even when only two cameras are used for tracking.

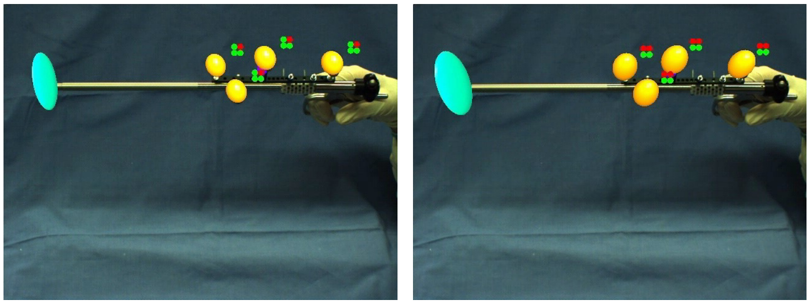


Fig. 8.11. Effect of occluded cameras on the estimation of covariance [168]. Pose estimation using three (left) or two (right) cameras — orange: covariance of fiducial, cyan: covariance at point of interest, purple: covariance at center, green/red: camera visibility of fiducials

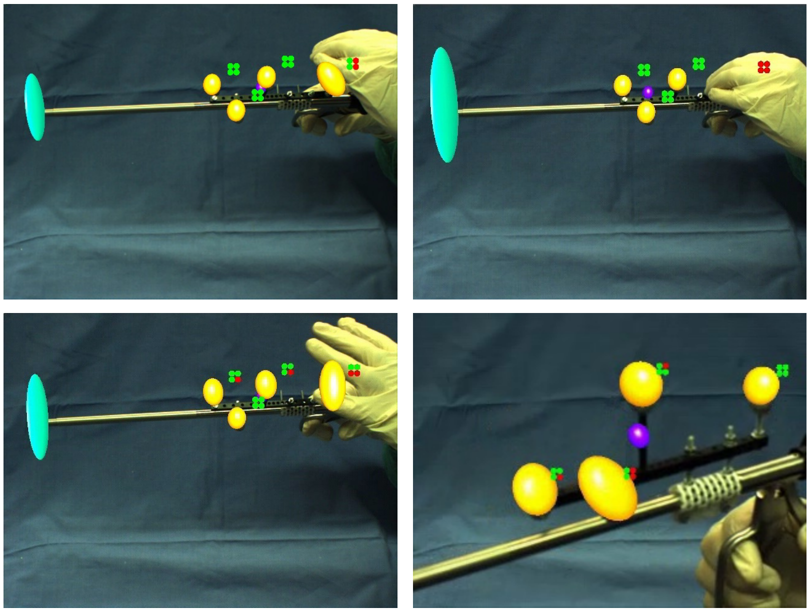


Fig. 8.12. Effect of occluded fiducials on the estimation of covariance [168] — orange: covariance of fiducial, cyan: covariance at point of interest, purple: covariance at center, green/red: camera visibility

Figure 8.12 shows the effect of the occlusion of fiducials, either by the hand of the user or by partial self occlusion (cf. section 5.3.2). It can be seen that partial occlusions can have a significant impact on the final accuracy. Our experiments also showed, that these partial occlusion are much more common that we expected during the normal use of the tracking system. It proved to be rather difficult to avoid partial occlusions for certain movements.

This shows, that the currently used offline error estimation using all possible fiducials can be used to analyze the general achievable accuracy, but is insufficient to provide actual online error estimates. Applications that rely on a guaranteed accuracy need to take the visibility of fiducials at runtime into account. This is even more important in setups with potentially small baselines.

9

Conclusion

In this thesis I have presented a comprehensive mathematical framework for the prediction of errors in complex tracking setups. This is similar to the work of Holloway [85], but includes full six-dimensional error distributions with covariant parts rather than maximum error statistics. I have presented simple formal rules for the propagation of these errors in general spatial relationship graphs. This makes the theory easily usable in a tracking middleware [200].

For the special case of optical tracking systems using one or more matrix cameras, I have presented [19] an estimation of the anisotropic fiducial location error. Similar to the estimation of the *dilution of precision* in *global positioning systems*, this estimation is only based on the geometrical configuration of the cameras and the fiducials. It can be seen roughly as an extension of the triangle intercept theorem to multiple dimensions.

Under some assumptions on the linearity I have shown the estimation of the marker target error for n-ocular (see section 5.3) and monocular (see section 5.4) cameras. With only slight modifications in the projection function the same estimation can be applied to tracking systems using line cameras as well [210]. It therefore is a useful tool for anyone using optical tracking. The predicted errors are consistent with previously measured results from the literature [214, 101], but provides full six-dimensional error covariances instead of one-dimensional root mean square error only. This is important for the further propagation of such errors.

I have shown in section 6 that there is a benefit of this new theory compared to the previous estimations that were based on the assumption of isotropic and independent errors when we consider error propagation. Under certain circumstances that are quite common in currently used tracking setups, the tracking errors can actually be larger than expected before. Additionally I have shown that it is really important to consider how many of the cameras and fiducials were actually used for the pose estimation. The occlusion of single cameras or single fiducials has a significant influence on the expected accuracy of an optical tracking system.

Finally I have given an overview of different calibration methods in the context of this general framework. This includes pointing device calibration, hand-eye calibration and the calibration of optical see-through head-mounted displays. A set of applications shows the use of the theory in common real-world scenarios and also emphasizes the need for an online estimation of the error inside the tracking system software.

Part IV

Appendix

A

Mathematica Implementations

This chapter explains the implementation of the core concepts in Mathematica. Additional implementations for visualization only are not explained but can be found in the according Mathematica notebooks.

Before we start to implement the algorithms, we need some helper functions to deal with homogeneous coordinates. The functions can be used to convert vectors and matrices to homogeneous form and back.

```
MakeHom[mat_?MatrixQ ] := Transpose[Append[Transpose[Append[mat,
   Table[0, {Dimensions[mat][[2]]}]]], Append[Table[0, {Dimensions[mat][[1]]}], 1]]]
MakeHom[v_?ListQ] := Append[v, 1]
MakeDeHom[mat_?MatrixQ] := Transpose[Drop[Transpose[Drop[mat, −1]], −1]]
MakeDeHom[v_?ListQ] := Drop[v, −1]
```

A block-diagonal matrix is can be constructed from a list of matrices using the following function:

```
BlockDiagonalMatrix[ mat : {_?MatrixQ ..}] :=
   Normal[SparseArray[Flatten[MapThread[Array[List, #1, #2 + 1] &,
   {#, Most[FoldList[Plus, {0,0},#]]}&[Dimensions /@ mat]], 2] → Flatten[mat]]]
```

A.1 Estimation of the FLE

This is the implementation of equation 5.2 that is used to generate C code for the accuracy prediction. This code makes use of the `Format` package which is not part of the original distribution but can be downloaded from the Website[1].

First we have to define the perspective projection function. For the sake of simplicity we use a very basic camera model with equal focal length in horizontal and vertical direction and the principle point exactly in the center of the image. However, a full pinhole camera model could be used likewise here if the parameters are available.

[1] `Format.m`, http://library.wolfram.com/infocenter/MathSource/60/

```
proj[{x_, y_, z_, h_}, f_] := 1/z {{f, 0, 0}, {0, f, 0}}.{x, y, z}
```

The last parameter h in the projection function is simply required for homogeneous coordinates and ignored in the projection.

We then need to define some abstract named parameters for the image plane covariances

```
S1 = {{S111, S112}, {S121, S122}};
S2 = {{S211, S212}, {S221, S222}};
S3 = {{S311, S312}, {S321, S322}};
...
```

for the maximum number of cameras we want to use.

Also we need the entries of the extrinsic camera matrices

```
C1 = {{C111, C112, C113, C114}, {C121, C122, C123, C124},
      {C131, C132, C133, C134}, {0, 0, 0, 1}};
C2 = {{C211, C212, C213, C214}, {C221, C222, C223, C224},
      {C231, C232, C233, C234}, {0, 0, 0, 1}};
C3 = {{C311, C312, C313, C314}, {C321, C322, C323, C324},
      {C331, C332, C333, C334}, {0, 0, 0, 1}};
...
```

and the output matrix for the resulting covariance

```
R = {{R11, R12, R13}, {R31, R32, R33}, {R31, R32, R33}};
```

As a helper function we define a function that builds a blockdiagonal matrix from a list of matrices

```
BlockDiagonalMatrix[ mat : {_?MatrixQ ..}] :=
   Normal[SparseArray[Flatten[MapThread[Array[List, #1, #2 + 1] &,
           {#, Most[FoldList[Plus, {0,
          0}, #]]} &[Dimensions /@ mat]], 2] -> Flatten[mat]]]
```

Now we can get the C code by applying equation 5.2 from section 5.3.2 to get the covariance in the two-camera case

```
fs = {f1, f2};
cams = {C1, C2};
covs = {S1, S2};
jacobi = BlockMatrix[MapThread[
                  {Outer[D, proj[#1.{x, y, z, 1}, #2], { x, y, z}]} &, {cams, fs}]];
back = Inverse[Transpose[jacobi].Inverse[BlockDiagonalMatrix[covs]].jacobi];
CAssign[R, back] >> twocamera_covariance.c
```

and likewise in the three-camera case

```
fs = {f1, f2, f3};
cams = {C1, C2, C3};
```

```
covs = {S1, S2, S3};
jacobi = BlockMatrix[MapThread[
                    {Outer[D, proj[#1.{x, y, z, 1}, #2], { x, y, z}]} &, {cams, fs}]];
back = Inverse[Transpose[jacobi].Inverse[BlockDiagonalMatrix[covs]].jacobi];
CAssign[R, back] >> threecamera_covariance.c
```

and so on for as many cameras as needed.

In Mathematica code, the same computation can be performed with

```
CalcCovariance[cams_, fs_, covars_, position_] := Module[{jacobi},
   jacobi = BlockMatrix[MapThread[
      {Outer[D,proj[#1.{x,y,z,1},#2], {x,y,z}]}&, {cams, fs}]];
   Inverse[Transpose[jacobi].Inverse[BlockDiagonalMatrix[covars]].jacobi]
      /. { x→position[[1]], y→position[[2]], z→position[[3]]}
]
```

The result is a 6 × 6 covariance matrix that can be used for further error propagation. Note that the extrinsic camera parameters in `cams` need to be defined such that they convert world coordinates into camera coordinates according to equation 5.1.

A.2 Estimation of the MTE from the FLE

To estimate the six-dimensional marker target error according to section 5.3.3, we implement the algorithm proposed by Hoff et.al [83]. We need a list `balls` containing the locations of the fiducials and a list `covs` containing the according covariances, both already in the marker target coordinate system. We can convert the covariances from room coordinates to marker target coordinates using the error propagation from equation 4.9:

```
rcov = Transpose[R].cov.R
```

where `R` is the rotation matrix from the pose of the marker target, which is given to us by the tracking system. Now we can implement equation 5.5:

```
PropagateFLEtoMTE[balls_, covs_] :=
   Module[{Mf},
   Mf = Join @@ (Drop[Jf[MakeDeHom[#1]], −1] & ) /@
       MakeHom /@ Function[x, x − Plus @@ balls/Length[balls]] /@ balls;
   PseudoInverse[Mf].BlockDiagonalMatrix[covs].Transpose[PseudoInverse[Mf]]
]
```

Note that using the pseudo-inverse here after expanding the outer matrix inversion is equivalent to equation 5.5.

If we want to simplify the estimation a bit, we can under certain circumstances use the same covariance `cov` for all fiducials in the target [154].

```
covs = Table[cov, {Length[balls]}]
```

A.3 Estimation of the TRE from the MTE

The target registration error can be estimated from the marker target error by forward propagation to the point of interest (see section 5.3.5 and 4.2.2).

First we need to define a general rotation matrix using Euler angles (cf. section 4.1.2).

```
R[ax_, ay_, az_] :=
   {{1, 0, 0}, {0, Cos[ax], −Sin[ax]}, {0, Sin[ax], Cos[ax]}}.
   {{Cos[ay], 0, Sin[ay]}, {0, 1, 0}, {−Sin[ay], 0, Cos[ay]}}.
   {{Cos[az], −Sin[az],0}, {Sin[az], Cos[az], 0}, {0, 0, 1}};
```

Using this matrix, we can build the function for the composition of the two poses.

```
f = Join[ R[ax, ay, az].{xq, yq, zq} + {tx, ty, tz} , {ax, ay, az} + {aq, bq, cq}];
```

This is the Jacobian of the composed function, evaluated at zero (cf. equation 4.11):

```
Jg[{xq_, yq_, zq_}] := Outer[D, f, {tx, ty, tz, ax, ay, az}] /.
   {ax → 0, ay → 0, az → 0, tx → 0, ty → 0, tz → 0}
```

Now we can apply the forward propagation (cf. equation 4.12):

```
PropagateMTEtoTRE[poi_?VectorQ, cov_?MatrixQ] :=
   Jg[MakeDeHom[poi]].cov.Transpose[Jg[MakeDeHom[poi]]]
```

Note that for a three-dimensional point of interest, the upper left 3 × 3 matrix is sufficient.

A.4 Hand-Eye Calibration

For the hand-eye calibration we assume that the tracking target (hand) and according camera (eye) poses are given as lists of homogeneous matrices `Ai` for the eye (pose of the calibration pattern in camera coordinates) and `Bi` for the hand (pose of the target in tracking system cordinates).

From this we compute all possible pairs of relative motions

```
Hc = Flatten[Table[Table[Ai[[j]].Inverse[Ai[[i]]],
           {j, i + 1, Length[Ai]}], {i, Length[Ai]}], 1];
Hg = Flatten[Table[Table[Inverse[Bi[[j]]].Bi[[i]],
           {j, i + 1, Length[Bi]}], {i, Length[Bi]}], 1];
```

The hand eye calibration itself is implemented according to Tsai-Lenz [185]

```
HandEyeTsaiLenz[Hg_, Hc_] := Module[{Pg, Pc, lefthand, righthand, Rcg, Tcg},
    Pg = Drop[DCM2Quat[#1], 1] & /@ Hg;
    Pc = Drop[DCM2Quat[#1], 1] & /@ Hc;
    (∗ Solve for the rotational part ∗)
    lefthand = Flatten[Skew /@ (Pc + Pg), 1];
    righthand = Flatten[Pc − Pg, 1];
    Pcg = PseudoInverse[lefthand].righthand;
    Pcg = 2/Sqrt[1 + Pcg.Pcg] Pcg;
    Rcg = (1 − 0.5 ( Pcg.Pcg))IdentityMatrix[3]
         + 0.5 ∗( (Transpose[{Pcg}].{Pcg}) + Sqrt[4 − Pcg.Pcg] Skew[Pcg]);

    (∗ Solve for the translational part ∗)
    lefthand = Flatten[ (#[[Range[3], Range[3]]]
              − IdentityMatrix[3]) & /@ Hg, 1];
    righthand = Flatten[(Rcg.(#[[Range[3], 4]]) & /@ Hc), 1]
              − Flatten[(#[[Range[3], 4]]) & /@ Hg, 1];

    Tcg = PseudoInverse[lefthand].righthand;
    Append[Transpose[Append[Transpose[Rcg], Tcg]], {0, 0, 0, 1}]
]
```

with

```
Skew[{x_, y_, z_}] := {{0, −z, y}, {z, 0, −x}, {−y, x, 0}}
```

A.5 Pointing Device Calibration

Pointing device calibration is described in section 7.1. We assume that we have the poses of the pivoting process already stored in two lists, `pos` for the positions and `rots` for the rotations. First we load an additional package for matrix manipulations,

```
<< LinearAlgebra‘MatrixManipulation‘
```

and then we can build the matrices for the equation system in equation 7.2.

```
A = BlockMatrix[Table[{rots[[i]], −IdentityMatrix[3]}, {i, Length[rots]}]];
b = −Flatten[Table[pos[[i]], {i, Length[pos]}]];
```

This system is solved using singular value decomposition with

```
svd = SingularValueDecomposition[A];
phpw = Plus @@ Table[
   ( (Transpose[svd[[1]]][[i]].b)/svd[[2, i, i]] ) ∗ Transpose[svd[[3]]][[i]],
 {i, Length[svd[[3]]]}]
```

The resulting vector **phpw** contains the position p_h of the tip in marker coordinates in the first three entries and the position p_w of the tip in world coordinates in the last three.

B

Authored and Co-Authored Publications

i. Bauer, M., Schlegel, M., Pustka, D., Navab, N., Klinker, G., *Predicting and Estimating the Accuracy of Vision-Based Optical Tracking Systems*, in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)* (Santa Barbara (CA), USA, 2006)

ii. Pustka, D., Huber, M., Bauer, M., Klinker, G., *Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems*, in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)* (2006)

iii. Wagner, M., MacWilliams, A., Bauer, M., Klinker, G., Newman, J., Pintaric, T., Schmalstieg, D., *Fundamentals of Ubiquitous Tracking*, in *Second International Conference on Pervasive Computing, Hot Spots section* (Vienna, Austria, 2004)

iv. Traub, J., Feuerstein, M., Bauer, M., Schirmbeck, E. U., Najafi, H., Bauernschmitt, R., Klinker, G., *Augmented Reality for Port Placement and Navigation in Robotically Assisted Minimally Invasive Cardiovascular Surgery*, in *Proceedings of Computer Assisted Radiology and Surgery (CARS 2004)* (Chicago, USA, 2004) 735–740

v. Newman, J., Wagner, M., Bauer, M., *Ubiquitous Tracking for Augmented Reality*, in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)* (Arlington, VA, USA, 2004)

vi. Bauernschmitt, R., Schirmbeck, E., Groher, M., Keitler, P., Bauer, M., Najafi, H., Klinker, G., Lange, R., *Navigierte Platzierung endovaskulaerer Aortenstents*, in Zeitschrift fuer Kardiology, S3 (2004) 116

vii. Bauer, M., Wagner, M., Toennis, M., Klinker, G., Broy, V., *Lehrkonzept für ein Augmented Reality Praktikum*, in *1. Workshop "Virtuelle und Erweiterte Realität"* (Chemnitz, Germany, 2004)

viii. Newman, J., Wagner, M., Pintaric, T., MacWilliams, A., Bauer, M., Klinker, G., Schmalstieg, D., *Fundamentals of Ubiquitous Tracking for Augmented Reality*, Tech. Rep. TR-188-2-2003-34, Vienna University of Technology 2003

ix. MacWilliams, A., Sandor, C., Wagner, M., Bauer, M., Klinker, G., Brügge, B., *Herding Sheep: Live System Development for Distributed Augmented Reality*, in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2003)

x. Bauer, M., Hilliges, O., MacWilliams, A., Sandor, C., Wagner, M., Newman, J., Reitmayr, G., Fahmy, T., Klinker, G., Pintaric, T., Schmalstieg, D., *Integrating Studierstube and DWARF*, in *International Workshop on Software Technology for Augmented Reality Systems (STARS)* (2003)

xi. Sandor, C., MacWilliams, A., Wagner, M., Bauer, M., Klinker, G., *SHEEP: The Shared Environment Entertainment Pasture*, in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2002)

xii. Kruppa, H., Bauer, M., Schiele, B., *Skin Patch Detection in Real-World Images*, in *Annual Symposium for Pattern Recognition of the DAGM 2002*, Lecture Notes in Computer Science (Springer, 2002)

xiii. Klinker, G., Dutoit, A., Bauer, M., Bayer, J., Novak, V., Matzke, D., *Fata Morgana – A Presentation System for Product Design*, in *International Symposium on Aumgented and Mixed Reality ISMAR 2002* (2002)

xiv. Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Sandor, C., Wagner, M., *An Architecture Concept for Ubiqutous Computing Aware Wearable Computers*, in *International Workshop on Smart Appliances and Wearable Computing (IWSAWC)* (2002)

xv. Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., Wagner, M., *Design of a Component-Based Augmented Reality Framework*, in *Proceedings of the International Symposium on Augmented Reality (ISAR)* (2001)

References

1. D. F. Abawi, J. Bienwald, R. Dörner. Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 260–261 (2004). (5.2.3, 5.4.2, 5.4.2, 5.4.2, 5.4.2)
2. S. Ahn, H. Warnecke, R. Kotowski. Systematic Geometric Image Measurement Errors of Circular Object Targets. *Photogrammetric Record*, 16(93):485–502 (1999). (5.3.2)
3. B. D. Allen, G. Welch. A general method for comparing the expected performance of tracking and motion capture systems. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, 201–210. ACM Press, New York, NY, USA (2005). (1.2, 5.1.1, 5.3.1)
4. M. Aron, G. Simon, M. O. Berger. Handling uncertain sensor data in vision-based camera tracking. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, 58–67 (2004). (7.2.5)
5. K. O. Arras. An Introduction To Error Propagation: Derivation, Meaning and Examples of Equation $C_y = F_x C_x F_x^T$. Tech. Rep. EPFL-ASL-TR-98-01 R3, École Polytechnique Fédérale de Lausanne (1998). (3.4)
6. ARTESAS. German Federal Ministry for Education and Research sponsored research project (2004-2006). (1.3.3)
7. ARVIKA. German Federal Ministry for Education and Research sponsored research project (1999-2003). (1.3.3)
8. R. Azuma. Tracking requirements for augmented reality. *Communications of the ACM*, 36(7):50–51 (1993). (1.2, 7.4)
9. R. Azuma. *Predictive Tracking for Augmented Reality.* Ph.D. thesis, UNC Chapel Hill Dept. of Computer Science (1995). (3.4.3, 5.1.1)
10. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, 21(6):34–47 (2001). (1.1, 1.3)
11. R. Azuma, G. Bishop. A Frequency-Domain Analysis of Head-Motion Prediction. In *Proceedings of SIGGRAPH '95*, 401–408. Los Angeles, CA (1995). (1.2.3)
12. R. T. Azuma. A Survey of Augmented Reality. *Presence, Special Issue on Augmented Reality*, 6(4):355–385 (1997). (1, 1.1, 1.3, 5.1)

13. R. T. Azuma, G. Bishop. Improving Static and Dynamic Registration in an Optical See-through HMD. In *Proceedings of SIGGRAPH '94*, 194–204. Orlando (1994). (5.1.1, 7.4.2)
14. Y. Baillot, S. Julier, D. Brown, M. A. Livingston. A Tracker Alignment Framework for Augmented Reality. In *Proceedings of 2nd International Symposium on Mixed and Augmented Reality (ISMAR)*, 142–150 (2003). (7.3)
15. M. Bajura, H. Fuchs, R. Ohbuchi. Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery within the Patient. Tech. rep., University of North Carolina at Chapel Hill, Chapel Hill, NC, USA (1992). (1.1)
16. O. A. Bauchau, L. Trainelli. The Vectorial Parameterization of Rotation. *Nonlinear Dynamics*, V32(1):71–92 (2003). (4.1.4, 4.1.6)
17. M. Bauer, B. Brügge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, M. Wagner. Design of a Component-Based Augmented Reality Framework. In *Proceedings of the International Symposium on Augmented Reality (ISAR)* (2001). (2.2.2)
18. M. Bauer, M. Schlegel, D. Pustka, N. Navab, G. Klinker. Interactive Visualization of the Predicted Accuracy of Marker-Based Optical Tracking Systems. Demonstration at IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06) (2006). (8.3)
19. M. Bauer, M. Schlegel, D. Pustka, N. Navab, G. Klinker. Predicting and Estimating the Accuracy of Vision-Based Optical Tracking Systems. Proc. International Sympoium on Mixed and Augmented Reality (ISMAR 2006) (2006). (5.3.2, 5.3.2, 6.4.2, 8.10, 9)
20. C. Berenstein, L. Kanal, D. Lavine, E. Olson. A geometric approach to subpixel registration accuracy. *Computer Vision, Graphics, and Image Processing*, 40(3):334–360 (1987). (5.3.2)
21. D. K. Bhatanagar. Position Trackers for Head Mounted Dispaly Systems: A Survey. Tech. Rep. TR93-010, University of North Carolina at Chapel Hill (1993). (1.2, 6.3)
22. G. Bianchi, C. Wengert, M. Harders, P. Cattin, G. Székely. Camera-Marker Alignment Framework and Comparison with Hand-Eye Calibration for Augmented Reality Applications. In *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'05)*, 188–189 (2005). (7.2.1)
23. M. Billinghurst, H. Kato. Real world teleconferencing. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, 194–195. ACM Press, New York, NY, USA (1999). (5.3.2, 5.4.1, 5.16, 5.4.2)
24. M. Billinghurst, I. Poupyrev. Shared Space: Mixed Reality Interface for Collaborative Computing. In *Imagina'2000 Official Guide: Innovation Village Exhibition*, 52 (2000). (5.3.2)
25. O. Bimber, R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Ltd., Natick, MA, USA (2005). ISBN 1568812302. (7.4)
26. W. Birkfellner, M. Figl, K. Huber, F. Watzinger, F. Wanschitz, R. Hanel, A. Wagner, D. Rafolt, R. Ewers, H. Bergmann. The Varioscope AR - A Head-Monted Operating Microscope for Augmented Reality. In *MICCAI '00: Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, 869–877. Springer-Verlag, London, UK (2000). (1.2.1)
27. G. Bishop, H. Fuchs, W. Bricken, F. B. Jr., M. Brown, C. Burbeck, N. Durlach, S. Ellis, M. Green, J. Lackner, M. McNeill, M. Moshell,

R. Pausch, W. Robinett, M. Srinivasan, I. Sutherland, D. Urban, E. Wenzel. Research directions in virtual environments: Report of an NSF Invitational Workshop, March 23-24, 1992, University of North Carolina at Chapel Hill. *SIGGRAPH Comput. Graph.*, 26(3):153–177 (1992). (1.3)

28. G. Bishop, G. Welch, B. D. Allen. Tracking: Beyond 15 Minutes of Thought. SIGGRAPH Course Notes (2001). (1.2, 3, 3.4.3, 4.1.2, 5.1.1, 6.3)
29. O. Cakmakci, J. Rolland. Head-Worn Displays: A Review. *IEEE Journal of Display Technologies*, 2(3):199–216 (2006). (1.1)
30. T. P. Caudell, D. W. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proceedings of the 25th Hawaii International Conference on System Sciences*, vol. 2, 659–669 (1992). (1.1, 1.2.2)
31. P. A. Cerfontaine, M. Schirski, D. Bündgens, T. Kuhlen. Automatic Multi-Camera Setup Optimization for Optical Tracking. In *Proceedings of IEEE Virtual Reality* (2006). (5.1.2, 6.2.1)
32. M. Chasles. Note sur les proprietes generales du systeme de deux corps semblables entr'eux et places d'une maniere quelconque dans l'espace; et sur le deplacement fini ou infiniment petit d'un corps solide libre. *Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques*, 14:321–326 (1830). (4.1.5)
33. F. Chassat, S. Lavallee. Experimental Protocol for Accuracy Evaluation of 6-d Localizers for Computer-Integrated Surgery: Application to Four Optical Localizers. In *MICCAI '98: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, 277–284. Springer-Verlag, London, UK (1998). (5.3.2)
34. H. H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 145–151 (1991). (4.1.5, 7.2.3, 7.2.3)
35. H. H. Chen, H. Zhuang, Z. S. Roth. Comments on 'Comments on 'Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB" (with reply). *Robotics and Automation, IEEE Transactions on*, 8(4):493–494 (1992). (7.2.3)
36. X. Chen. *Design of many-camera tracking systems for scalability and efficient resource allocation*. Ph.D. thesis, Stanford University (2002). (5.1.2, 6.2.1)
37. X. Chen, J. Davis. Camera Placement Considering Occlusion for Robust Motion Capture. Tech. Rep. CS-TR-2000-07, Stanford University Computer Science (2000). (5.1.2, 6.2.1)
38. T. Clarke. An analysis of the properties of targets used in digital close range photogrammetric measurement. *Proceedings of SPIE Videometrics III*, 2350:251–262 (1994). (5.3.2)
39. E. M. Coelho, B. MacIntyre, S. Julier. OSGAR: A Scenegraph with Uncertain Transformations. In *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)*, 6–15. IEEE, Washington, DC (2004). (2.2.3, 4, 4.2.2)
40. A. Criminisi. Modelling and Using Uncertainties in Video Metrology. Tech. rep., Information Engineering Research Group, Department of Engineering Science, University of Oxford (1997). (5.4.3)
41. A. Cumani, A. Guiducci, P. Grattoni, G. Pettiti, F. Pollastri. High accuracy localization of calibration points for dimensional measurements by

image processing techniques. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1761–1765 vol.2 (1991). (5.3.2)

42. J. S. Dai. An historical review of the theoretical development of rigid body displacements from Rodrigues parameters to the finite twist. *Mechanism and Machine Theory*, 41(1):41–52 (2006). (4.1.4, 4.1.5)
43. E. B. Dam, M. Koch, M. Lillholm. Quaternions, Interpolation and Animation. Tech. Rep. IT-C758, Department of Computer Science, University of Copenhagen (1998). (4.1.4)
44. K. Daniilidis. Hand-Eye Calibration Using Dual Quaternions. *The International Journal of Robotics Research*, 18(3):286–298 (1999). (7.2.3)
45. K. Daniilidis, E. Bayro-Corrochano. The dual quaternion approach to hand-eye calibration. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, 318–322 (1996). (4.1.5, 7.2.3)
46. P. David, D. Dementhon, R. Duraiswami, H. Samet. SoftPOSIT: Simultaneous Pose and Correspondence Determination. *Int. J. Comput. Vision*, 59(3):259–284 (2004). (5.1.2)
47. L. Davis, E. Clarkson, J. P. Rolland. Predicting Accuracy in Pose Estimation for Marker-based Tracking. In *Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, 28 (2003). (4.2.1, 5.3, 5.3.3)
48. L. Davis, F. G. Hamza-Lup, J. P. Rolland. A Method for Designing Marker-Based Tracking Probes. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, 120–129 (2004). (5.1.2, 5.3.3, 6, 6.4.4)
49. L. Davis, J. Rolland, R. Parsons, E. Clarkson. Methods for designing head-tracking probes. *Proceedings of the Joint Conference on Information Sciences (JCIS)*, 498–502 (2002). (5.1.2, 5.3.3, 6)
50. T. Drummond, R. Cipolla. Real-Time Visual Tracking of Complex Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946 (2002). (4.1.6)
51. P. J. Edwards, D. L. G. Hill, D. J. Hawkes, R. Spink, A. C. F. Colchester, A. Strong, M. Gleeson. Neurosurgical guidance using the stereo microscope. In *Computer Vision, Virtual Reality and Robotics in Medicine, LNCS*, vol. 904, 555–564. Springer Verlag (1995). (1.2.1)
52. L. Euler. Découverte d'un nouveau principe de mécanique. *Mémoires de l'académie des sciences de Berlin*, 6:185–217 (1752). (4.1.3, 4.1.3)
53. L. Euler. Nova methodus motum corporum rigidorum degerminandi. *Novi Commentarii academiae scientiarum Petropolitanae*, 20:208–238 (1776). (4.1.2)
54. S. Feiner, B. Macintyre, D. Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62 (1993). (1.2.2)
55. M. Feuerstein, T. Mussack, S. M. Heining, N. Navab. Registration-Free Laparoscope Augmentation for Intra-Operative Liver Resection Planning. In *SPIE Medical Imaging (to appear)*. San Diego, California, USA (2007). (8.2, 8.2)
56. M. Fiala. ARTag, An Improved Marker System Based on ARToolkit. Tech. Rep. NRC/ERB-1111, national Research Council Canada (2004). (5.3.2)

57. M. Fiala. ARTag Revision 1, A Fiducial Marker System Using Digital Techniques. Tech. Rep. NRC/ERB-1117, national Research Council Canada (2004). (5.3.2, 5.4.1)
58. M. Fiala. ARTag, a Fiducial Marker System Using Digital Techniques. *cvpr*, 2:590–596 (2005). (5.3.2)
59. M. Fitzgibbon, A. W.and Pilu, R. B. Fisher. Direct least-squares fitting of ellipses. *IEEE Tran. Patt. An. Mach. Int.*, 21(5):476–480 (1999). (5.3.2)
60. J. M. Fitzpatrick, J. B. West, J. C. R. Maurer. Predicting error in rigid-body, point-based registration. *IEEE Transactions on Medical Imaging*, 17(5):694–702 (1998). (5.2.3, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.3.5, 6.1.1, 6.1.2, 6.4, 6.4.2, 8.3, 8.10, 8.4)
61. G. Florou, R. Mohr. What Accuracy for 3D Measurements with Cameras? In *International Conference on Pattern Recognition*, I: 354–358 (1996). (5.1.3, 5.2.2, 5.3.2)
62. D. D. Frantz, S. R. Kirsch, A. D. Wiles. Specifying 3D Tracking System Accuracy - One Manufacturer's View. In *Bildverarbeitung für die Medizin*, 234–238. Berlin (2004). (5.2.1, 5.3.2)
63. D. D. Frantz, A. D. Wiles, S. E. Leis, S. R. Kirsch. Accuracy assessment protocols for electromagnetic tracking systems. *Phys Med Biol*, 48(14):2241–2251 (2003). (5.3.2)
64. W. Friedrich. ARVIKA - Augmented Reality for Development, Production and Service. In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, 3. IEEE Computer Society, Washington, DC, USA (2002). (1.2.2, 1.3.3)
65. W. Friedrich, ed. *ARVIKA - Augmented Reality für Entwicklung, Produktion und Service*. Publicis Corporate Publishing, Erlangen (2004). (1.2.2, 1.3.3)
66. A. Fuhrmann, R. Splechtna, J. Přikryl. Comprehensive Calibration and Registration Procedures for Augmented Reality. In *Proceedings of the joint 5th Immersive Projection Technology and 7th Eurographics Virtual Environments Workshop (EGVE)*, 219–228. Stuttgart (2001). (7.1.1, 7.2)
67. T. Furness. Helmet-mounted displays and their aerospace applications. *National Aerospace Electronics Conference* (1969). (1.1)
68. T. Furness, J. Kollin. Virtual retinal display. US Patent No. 5,467,104 (1995). (7.4)
69. A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts; London, 15th ed. (1974). (3.2, 3.2.1, 3.2.2, 3.4.1)
70. Y. Genc, F. Sauer, F. Wenzel, M. Tuceryan, N. Navab. Optical See-Through HMD Calibration: A Stereo Method Validated with a Video See-Through System. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR)*, 165–174. Munich (2000). (7.4.1, 7.4.2)
71. Y. Genc, M. Tuceryan, N. Navab. Practical Solutions for Calibration of Optical See-Through Devices. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 169–175. Darmstadt, Germany (2002). (7.4.1)
72. J.-F. V. Gomez, G. Simon, M.-O. Berger. Calibration Errors in Augmented Reality: A Practical Study. In *Fourth IEEE / ACM International Symposium on Mixed and Augmented Reality (ISMAR 2005)*, 154–163 (2005). (5.1.3)
73. F. S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48 (1998). (4.1.6)

74. W. Grimson, G. Ettinger, S. White, T. Lozano-Perez, W. Wells III, R. Kikinis. An automatic registration method for frameless stereotaxy, imageguided surgery, and enhanced reality visualization. *Medical Imaging, IEEE Transactions on*, 15(2):129–140 (1996). (1)
75. W. R. Hamilton. On a new Species of Imaginary Quantities connected with a Theory of Quaternions. *Proceedings of the Royal Irish Academy*, 2:424–234 (1844). (4.1.4)
76. W. R. Hamilton. On Quaternions, or on a New System of Imaginaries in Algebra. *Philosophical Magazine*, 25(3):489–495 (1844). (4.1.4)
77. R. I. Hartley, P. Sturm. Triangulation. *Lecture Notes in Computer Science*, 970:190–197 (1995). (5.1.2)
78. R. I. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press (2000). (3.4.1, 3.4.2, 5.3.2)
79. S. Hennauer. *Calibration of Optical See-Through Head-Mounted Displays*. Master's thesis, Technische Universität München (2005). (7.4.2, 7.4.2, 7.4.2, 7.4.2, 7.4.2, 7.4.3, 7.4.3, 7.4.3)
80. G. Hillebrand, M. Bauer, K. Achatz, G. Klinker. Inverse Kinematic Infrared Optical Finger Tracking. In *9th International Conference on Humans and Computers (HC 2006), Aizu, Japan* (2006). (5.1.2)
81. R. L. Hirsh, G. N. DeSouza, A. C. Kak. An Iterative Approach to the Hand-Eye and Base-World Calibration Problem. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Seoul, Korea (2001). (7.2.4)
82. D. Hoag. Apollo Guidance and Navigation Considerations of Apollo IMU Gimbal Lock. Tech. Rep. E-1344, MIT Instrumentation Laboratory (1963). (4.1.2)
83. W. Hoff, T. Vincent. Analysis of Head Pose Accuracy in Augmented Reality. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 6(4), 319–334. IEEE Computer Society (2000). (5.3, 5.3.3, A.2)
84. W. A. Hoff. Fusion of Data from Head-Mounted and Fixed Sensors. In *First IEEE International Workshop on Augmented Reality (IWAR '98)* (1998). (1.2.1, 2.1, 3.4.3, 5.3.3, 6.3)
85. R. L. Holloway. *Registration errors in augmented reality systems*. Ph.D. thesis, Univ. of North Carolina Chapel Hill (1995). (3.4.3, 3.4.3, 4.2.2, 4.2.2, 4.2.2, 4.2.2, 5.1.1, 9)
86. B. K. Horn. Some Notes on Unit Quaternions and Rotation (2001). Unpublished. (4.1.4)
87. B. K. P. Horn. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America*, 4(4) (1987). (5.3, 5.3.3)
88. Guide to the expression of uncertainty in measurement (GUM). International Organisation for Standardization (ISO) (1993). (1.2.2, 5.2)
89. Expression of the Uncertainty of Measurement in Calibration. European cooperation for Accreditation EA-4/02 (1999). (1.2.2, 5.2, 5.2.1)
90. A. L. Janin, D. W. Mizell, T. P. Caudell. Calibration of Head-Mounted Displays for Augmented Reality Applications. In *Proceedings of the Virtual Reality Annual International Symposium (VRAIS)*, 246–255. Seattle (1993). (7.4.1)

91. M. P. Johnson. *Exploiting Quaternions to Support Expressive Interactive Character Motion.* Ph.D. thesis, Massachusetts Institute of Technology (2002). (4.1.2, 4.1.3, 4.1.3, 4.1.4, 4.1.4)
92. A. T. Johnston. Non-Isotropic Behavior of Active Infrared Markers (Tested With the Polaris System). Tech. Rep. NDI-TR-0133, Northern Digital, Inc., Waterloo, ON, Canada (1999). (5.1.2, 5.3.2)
93. S. Julier. The scaled unscented transformation. *American Control Conference, 2002. Proceedings of the 2002*, 6 (2002). (3.4.5)
94. S. Julier, J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls* (1997). (3.4.5)
95. S. Julier, J. Uhlmann, H. Durrant-Whyte. A new method for the nonlinear transformation of means andcovariances in filters and estimators. *Automatic Control, IEEE Transactions on*, 45(3):477–482 (2000). (3.4.5, 3.4.5)
96. S. Julier, J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422 (2004). (3.4.5)
97. S. J. Julier, J. K. Uhlmann, H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, vol. 3, 1628–1632 vol.3 (1995). (3.4.5)
98. K. Kanatani. Unbiased estimation and statistical analysis of 3-D rigid motion from two views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(1):37–50 (1993). (3.3)
99. H. Kato, M. Billinghurst. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR)*, 85. San Francisco (1999). (7.4.1)
100. L. Kavan, S. Collins, C. O'Sullivan, J. Zara. Dual Quaternions for Rigid Transformation Blending. Tech. Rep. TCD-CS-2006-46, Trinity College Dublin (2006). (4.1.4)
101. R. Khadem, C. C. Yeh, M. Sadeghi-Tehrani, M. R. Bax, J. A. Johnson, J. N. Welch, E. P. Wilkinson, R. Shahidi. Comparative tracking error analysis of five different optical tracking systems. *Computer Aided Surgery*, 5(2):98–107 (2000). (5.3, 5.3.2, 9)
102. G. Klein. *Visual Tracking for Augmented Reality.* Optical tracking, accuracy, markerless tracking, University of Cambridge, Department of Engineering (2006). (5.1.2, 6.3)
103. G. Klinker. Augmented Reality: A Problem in Need of Many Computer Vision-based Solutions. In *Proc. NATO Advanced Research Workshop, Confluence of Computer Vision and Computer Graphics (CAIP'99).* Ljubljana, Slowenia (1999). (1.3, 7.4)
104. G. Klinker, A. Dutoit, M. Bauer, J. Bayer, V. Novak, D. Matzke. Fata Morgana – A Presentation System for Product Design. In *International Symposium on Aumgented and Mixed Reality ISMAR 2002* (2002). (1.2.2)
105. R. Langley. Dilution of Precision. *GPS World*, 10(5):52–59 (1999). (5.2.3)
106. S. Langron, A. Collins. Perturbation Theory for Generalized Procrustes Analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(2):277–284 (1985). (5.3.3)
107. V. Lepetit, P. Fua. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89 (2005). (4.1.1, 4.2.1, 4.2.1, 5.4.1, 5.4.1)

108. Q. Li, L. Zamorano, Z. Jiang, J. X. Gong, A. Pandya, R. Perez, F. Diaz. Effect of optical digitizer selection on the application accuracy of a surgical localization system – a quantitative comparison between the OPTOTRAK and flashpoint tracking systems. *Computer Aided Surgery*, 4(6):314–321 (1999). (6.1.2)
109. T. Luhmann. Photogrammetrische Verfahren in der industriellen Messtechnik. In *Publikationen der Deutsche Gesellschaft für Photogrammetrie und Fernerkundung eV (DGPF)*, vol. 9 (2001). (5.1.3)
110. T. Luhmann. *Nahbereichsphotogrammetrie. Grundlagen, Methoden und Anwendungen*. Wichmann, Heidelberg, Germany (2003). (5.1.3)
111. G. Lukács, R. Martin, D. Marshall. Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, 671–686. Springer-Verlag, London, UK (1998). (7.1.3)
112. B. Ma, R. E. Ellis. A Spatial-Stiffness Analysis of Fiducial Registration Accuracy. In *Proc. 6th International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI 2003*, 359–366 (2003). (5.3, 5.3.3)
113. B. MacIntyre, E. M. Coelho. Adapting to Dynamic Registration Errors Using Level of Error (LOE) Filtering. In *International Symposium on Augmented Reality (ISAR 2000)* (2000). (1.2, 2.2.3, 5.2.3)
114. A. MacWilliams. *A Decentralized Adaptive Architecture for Ubiquitous Augmented Reality Systems*. Ph.D. thesis, Technische Universität München, Institut für Informatik (2004). (2.2.2, 2.2.4)
115. P. Malbezin, W. Piekarski, B. Thomas. Measuring ARToolkit Accuracy in Long Distance Tracking Experiments. In *1st International ARToolkit Workshop*. Darmstadt, Germany (2002). (5.4.2, 5.4.2, 5.4.2)
116. P. Massatt, K. Rudnick. Geometric formulas for dilution of precision calculations. *Navigation*, 37:379–391 (1991). (5.2.3)
117. E. McGarrity, Y. Genc, M. Tuceryan, C. B. Owen, N. Navab. A New System for Online Quantitative Evaluation of Optical See-through Augmentation. In *Proceedings of the 4th International Symposium on Augmented Reality (ISAR)*, 157–166. New York (2001). (7.4.2)
118. J. P. Mellor. Realtime Camera Calibration for Enhanced Reality Visualization. In N. Ayache, ed., *Computer Vision, Virtual Reality and Robotics in Medicine*, no. 905 in Lecture Notes in Computer Science. Springer, Nice, France (1995). (1)
119. K. Meyer, H. L. Applewhite, F. A. Biocca. A Survey of Position Trackers. *Presence: Teleoperators and Virtual Environments*, 1(2):173–200 (1992). (1.2)
120. P. Milgram, F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information and Systems*, E77-D(12):1321–1329 (1994). (1)
121. P. Milgram, H. Takemura, A. Utsumi, F. Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. *SPIE: Telemanipulator and Telepresence Technologies*, 2351:282 (1994). (1)
122. M. Mitschke, N. Navab. Recovering Projection Geometry: How a Cheap Camera Can Outperform an Expensive Stereo System. In *Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, 1193–1200. IEEE Computer Society, Hilton Head, SC, USA (2000). (5.3.1, 6.3)

123. T. Morris, M. Donath. Using a maximum error statistic to evaluate measurement errors in 3d position and orientation tracking systems. *Presence: Teleoperators and Virtual Environments*, 2(4):314–343 (1993). (5.3.3)
124. R. M. Murray, S. S. Sastry, L. Zexiang. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., Boca Raton, FL, USA (1994). ISBN 0849379814. (4.1.5)
125. N. Navab, A. Bani-Kashemi, M. Mitschke. Merging Visible and Invisible: Two Camera-Augmented Mobile C-arm (CAMC) Applications. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, 134–141 (1999). (1.2.1)
126. N. Navab, S. Zokai, Y. Genc, E. M. Coelho. An On-line Evaluation System for Optical See-through Augmented Reality. In *Proceedings of the IEEE Virtual Reality Conference (VR)*, 245–246. Chicago (2004). (7.4.2)
127. J. Newman, M. Wagner, M. Bauer, A. MacWilliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, G. Klinker. Ubiquitous Tracking for Augmented Reality. In *International Symposium on Mixed and Augmented Reality (ISMAR).* Arlington, VA, USA (2004). (2, 2.1.1, 2.2.4)
128. J. Newman, M. Wagner, T. Pintaric, A. MacWilliams, M. Bauer, G. Klinker, D. Schmalstieg. Fundamentals of Ubiquitous Tracking for Augmented Reality. Tech. Rep. TUM-I0323, Technische Universität München (2003). (2.1)
129. S. Nölle. *Augmented Reality als Vergleichswerkzeug am Beispiel der Automobilindustrie.* Ph.D. thesis, Technische Universität München (2006). (1.2.2, 8.1)
130. Northern Digital, Inc. *Polaris Vicra User Guide* (2005). (7.1.1, 7.2)
131. Northern Digital, Inc. *Polaris Tool Design Guide* (2006). (5.3.2)
132. C. B. Owen, F. Xiao, P. Middlin. What is the best fiducial? In *1st International ARToolkit Workshop* (2002). (5.4.1, 5.16)
133. C. B. Owen, J. Zhou, A. Tang, F. Xiao. Display-Relative Calibration for Optical See-Through Head-Mounted Displays. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 70–78. Washington DC (2004). (7.4.1, 7.4.3)
134. K. Pentenrieder, P. Meier, G. Klinker. Analysis of Tracking Accuracy for Single-Camera Square-Marker-Based Tracking. In *3rd Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR* (2006). (5.2.2, 5.2.3, 5.4.2, 5.4.2)
135. K. Pentenrieder, J. Platonov, P. Meier. The need for accuracy statements in industrial augmented reality. In *2nd Workshop on Industrial Augmented Reality at ISMAR 2006* (2006). (1.2.2)
136. E. Pervin, J. A. Webb. Quaternions for Computer Vision and Robotics. *Computer Vision and Pattern Recognition*, 382–383 (1983). (4.1.4)
137. V. Pratt. Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph.*, 21(4):145–152 (1987). (7.1.3, 7.1.3)
138. W. Press, S. Teukolsky, W. Vetterling, B. Flannery. *Numerical recipes in C: The art of scientific computing.* Cambridge University Press New York, NY, USA (1992). (7.1.2)
139. D. Pustka. *Handling Error in Ubiquitous Tracking Setups.* Master's thesis, Technische Universität München (2004). (4.1.4, 4.2.1, 4.2.1, 4.2.2, 4.2.2, 4.2.3, 5.1.1)

140. D. Pustka. Construction of Data Flow Networks for Tracking in Augmented Reality Applications. In *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*. Koblenz, Germany (2006). (2.2.4)

141. D. Pustka, M. Huber, M. Bauer, G. Klinker. Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems. In *Proceedings of the 5th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2006). (2, 2.1, 2.1.1, 2.1.3, 7, 7.2.2)

142. G. Reitmayr, D. Schmalstieg. OpenTracker: An Open Software Architecture for Reconfigurable Tracking based on XML. In *Proceedings of IEEE Virtual Reality*, 285–286. Yokohama, Japan (2001). (2.2.1, 2.2.4)

143. J. Rekimoto. Matrix: a realtime object identification and registration method for augmented reality. In *3rd Asia Pacific Computer Human Interaction*, 63–68 (1998). (5.4.1, 5.16)

144. A. C. Rice. *Dependable Systems for Sentient Computing.* Master's thesis, University of Cambridge (2006). (5.3.2)

145. W. Robinett. Synthetic experience: a proposed taxonomy. *Presence: Teleoper. Virtual Environ.*, 1(2):229–247 (1992). (1.1)

146. R. Rohling, P. Munger, J. M. Hollerbach, T. Peter. Comparison of relative accuracy between a mechanical and an optical position tracker for image-guided neurosurgery. *J Image Guid Surg*, 1(1):30–34 (1995). (5.2.1)

147. J. Rolland, O. Cakmakci. The past, present, and future of head-mounted display designs. *Proc. SPIE*, 5638:368–377 (2004). (1.1, 7.4)

148. J. P. Rolland, L. Davis, Y. Baillot. A Survey of Tracking Technology for Virtual Environments. In W. Barfield, T. Caudell, eds., *Fundamentals of Wearable Computers and Augmented Reality*, 67–112. Lawrence Erlbaum, Mahwah, NJ, USA (2001). (1.2, 6.3)

149. M. Rothacher. Satellitengeodäsie 1 (Einführung in GPS). Vorlesungsskriptum (2003). (5.2.3)

150. J. N. Sanders-Reed. Error propagation in two-sensor three-dimensional position estimation. *Optical Engineering*, 40(4):627–636 (2001). (5.3.2)

151. P. Santos, A. Stork, A. Buaes, J. Jorge. Innovative geometric pose reconstruction for marker-based single camera tracking. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, 237–244. ACM Press, New York, NY, USA (2006). (5.4.2, 5.4.2)

152. F. Sauer, A. Khamene, S. Vogt. An Augmented Reality Navigation System with a Single-Camera Tracker: System Design and Needle Biopsy Phantom Trial. In *MICCAI '02: Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part II*, 116–124. Springer-Verlag, London, UK (2002). (1.2.1, 5.1.2, 5.4.1)

153. S. Schemerber, F. Chassat. Accuracy evaluation of a CAS system: Laboratory protocol and results with 6D localizers, and clinical experiences in otorhinolaryngology. *Computer Aided Surgery*, 13(1):1–6 (2001). (5.3.2)

154. M. Schlegel. *Predicting the Accuracy of Optical Tracking Systems.* Master's thesis, Technische Universität München (2006). (5.1.3, 5.3.2, A.2)

155. J. Schmidt, H. Niemann. Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, 399–406. Aka GmbH (2001). (4.1.3)

156. J. Schmidt, F. Vogt, H. Niemann. Robust Hand-Eye Calibration of an Endoscopic Surgery Robot Using Dual Quaternions. In B. Michaelis, G. Krell, eds., *25. Symposium für Mustererkennung, DAGM 2003, Magdeburg, LNCS*, vol. 2781. Springer-Verlag, Berlin (2003). (7.2.3, 8.2)

157. M. Schneberger. *Implementierung und Test eines videobasierten 3D-Messsystems zur Patientenpositionierung in der Praezisionsstrahlentherapie*. Master's thesis, Ruprecht-Karls-Universität Heidelberg (2000). (5.3.2)

158. M. Schneberger. *Spezifikation und Einsatz eines Stereokamerasystems zur videobasierten Patientenpositionierung in der Präzisionsstrahlentherapie*. Ph.D. thesis, Faculty of Physics and Astronomy, University of Heidelberg, Heidelberg, Germany (2003). (5.3.2)

159. R. Shahidi, M. R. Bax, C. R. M. Jr., J. A. Johnson, E. P. Wilkinson, B. Wang, J. B. West, M. J. Citardi, K. H. Manwaring, R. Khadem. Implementation, Calibration and Accuracy Testing of an Image-Enhanced Endoscopy System. *IEEE Trans. Med. Imaging*, 21(11):1524–1535 (2002). (1, 1.2.1, 8.2)

160. V. M. Shami, I. Waxman. Technology Insight: Current Status of Endoscopic Ultrasonography. *Nature Clinical Practice Gastroenterology & Hepatology*, 1(2):38–45 (2005). (8.2)

161. F. Shi, J. Wang, Y. Liu. An Approach to Improve Online Hand-Eye Calibration. *Pattern Recognition and Image Analysis*, 647–655 (2005). (7.2.3)

162. S.-W. Shih, Y.-P. Hung, W.-S. Lin. Accuracy Analysis on the Estimation of Camera Parameters for Active Vision Systems. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, 930. IEEE Computer Society, Washington, DC, USA (1996). (5.1.3, 5.3.2, 5.4.2)

163. Y. Shiu, S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneoustransform equations of the form AX=XB. *IEEE Transactions on Robotics and Automation*, 5(1):16–29 (1989). (7.2.3)

164. K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 245–254. ACM Press, New York, NY, USA (1985). (4.1.4)

165. M. R. Shortis, T. A. Clarke, T. Short. A comparison of some techniques for the subpixel location of discrete target images. *SPIE Videometrics III*, 2350:239–250 (1994). (5.3.2)

166. R. Sibson. Studies in the Robustness of Multidimensional Scaling: Procrustes Statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(2):234–238 (1978). (6.1.2)

167. T. Sielhorst. *High Accuracy Tracking for Medical Augmented Reality*. Master's thesis, TU München (2003). (5.1.2, 5.3.2, 5.4.1)

168. T. Sielhorst, M. Bauer, O. Wenisch, N. Navab. Online Estimation of the Target Registration Error for n-ocular Optical Tracking Systems. submitted (2007). (8.4, 8.11, 8.12)

169. T. Sielhorst, T. Blum, N. Navab. Synchronizing 3D Movements for Quantitative Comparison and Simultaneous Visualization of Actions. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 38– 47. Vienna, Austria (2005). (2.1, 7.3)

170. T. Sielhorst, M. Feuerstein, J. Traub, O. Kutter, N. Navab. CAMPAR: A software framework guaranteeing quality for medical augmented reality. *In-*

ternational Journal of Computer Assisted Radiology and Surgery, 1(S1):29–30 (2006). (2.2.5, 5.1.1)

171. G. Stetten, V. Chib, R. Tamburo. Tomographic Reflection to Merge Ultrasound Images with Direct Vision. In *IEEE Proceedings of the 2000 Applied Imagery Pattern Recognition (AIPR) Annual Workshop*. IEEE Computer Society, Los Alamitos, CA, USA (2000). (1.2.1)
172. I. Stock, M. Weber. Authoring technical documentation using a generic document model. In *SIGDOC '06: Proceedings of the 24th annual conference on Design of communication*, 172–179. ACM Press, New York, NY, USA (2006). (1.3.2)
173. D. Stricker, G. Klinker, D. Reiners. A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications. In *1. International Workshop on Augmented Reality (IWAR'98)*, 129–145. AK Peters, San Francisco (1998). (5.3.2)
174. K. H. Strobl, G. Hirzinger. Optimal Hand-Eye Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4647–4653. Beijing, China (2006). (7.2.3)
175. J. Stuelpnagel. On the parameterization of the three-dimensional rotation group. Tech. Rep. NASA-CR-53568, NASA (1948). (4.1.3, 4.1.4)
176. J. Sun, M. Smith, L. Smith, L. P. Nolte. Simulation of an optical-sensing technique for tracking surgical tools employed in computer-assisted interventions. *Sensors Journal, IEEE*, 5(5):1127–1131 (2005). (5.3)
177. I. E. Sutherland. A Head-Mounted Three Dimensional Display. In *Fall Joint Computer Conference*, vol. 33, 757–768 (1968). (1.1)
178. Z. Szalavári, D. Schmalstieg, A. Fuhrmann, M. Gervautz. Studierstube – An Environment for Collaboration in Augmented Reality. *Journal of the Virtual Reality Society* (1997). (2.2.1)
179. A. Tang, J. Zhou, C. B. Owen. Evaluation of Calibration Procedures for Optical See-Through Head-Mounted Displays. In *Proceedings of the 2nd IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 161–168. Tokyo (2003). (7.4, 7.4.3)
180. C. Teutsch, D. Berndt, E. Trostmann, M. Weber. Real-time detection of elliptic shapes for automated object recognition and object tracking. In *Proceedings of Machine Vision Applications in Industrial Inspection XIV (Electronic Imaging 2006, January 15-19, 2006, San Jose, California, USA)*, 171–179. SPIE (2006). (5.3.2)
181. M. Tidwell, R. Johnston, D. Melville, T. Furness. The Virtual Retinal Display—A Retinal Scanning Imaging System. *Proc. Virtual Reality World*, 95:325–333 (1995). (7.4)
182. J. C. Trinder, J. Jansa, Y. Huang. An assessment of the precision and accuracy of methods of digital target location. *ISPRS Journal of Photogrammetry and Remote Sensing*, 50(2):12–20 (1995). (5.3.2)
183. C. Trübswetter. Analyzing and Monitoring Tracking Accuracy of an A.R.T. System (2003). (5.1.3, 5.3.2)
184. R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344 (1987). (5.1.3)
185. R. Tsai, R. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358 (1989). (2.1, 7.2.3, 7.2.3, 7.2.3, 7.2.5, 8.3, A.4)

186. R. Y. Tsai, R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. In *Proceedings of the 4th International Symposium on robotics research*. Santa Cruz (CA) (1987). (7.2.3, 7.2.3)

187. M. Tuceryan, Y. Genc, N. Navab. Single-Point Active Alignment Method (SPAAM) for Optical See-Through HMD Calibration for Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 11(3):259–276 (2002). (7.4.1, 7.4.2, 7.4.2)

188. M. Tuceryan, D. S. Greer, R. T. Whitaker, D. E. Breen, E. Rose, K. H. Ahlers, C. Crampton. Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255–273 (1995). (2.1, 7.1.2)

189. M. Tuceryan, N. Navab, Y. Genc. System and Method for Calibrating a Stereo Optical See-Through Head-Mounted Display System for Augmented Reality. US Patent No. 6,753,828 B2 (2004). (7.4.1)

190. J. Van Zandt. A more robust unscented transform. *Proceedings of SPIE*, 4473:371 (2001). (3.4.5)

191. Optical 3D Measuring Systems - Imaging Systems with Point-by-Point Probing. VDI/VDE guideline 2634/1 (2002). (1.2.2, 5.2.1, 5.3.2, 5.3.2)

192. Optische 3D-Messsysteme - Bildgebende Systeme mit punktförmiger Antastung. Richtlinie VDI/VDE 2634 Blatt 1 (2002). (1.2.2, 5.3.2)

193. A. Verdun. Leonhard Eulers Einführung und Anwendung von Bezugssystemen in Mechanik und Astronomie. *Elemente der Mathematik*, 58(4):169–176 (2003). (4.1.2)

194. L. Vicci. Quaternions and Rotations in 3-Space: The Algebra and its Geometric Interpretation. Tech. Rep. TR01-014, Department of Computer Science, UNC Chapel Hill (2001). (4.1.4)

195. E. Viirre, H. Pryor, S. Nagata, T. Furness 3rd. The virtual retinal display: a new technology for virtual reality and augmented vision in medicine. *Stud Health Technol Inform*, 50:252–7 (1998). (7.4)

196. F. Vogt. *Augmented Light Field Visualization and Real-Time Image Enhancement for Computer Assisted Endoscopic Surgery*. Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2005). (8.2)

197. F. Vogt, S. Krüger, M. Winter, H. Niemann, W. Hohenberger, G. Greiner, C. Schick. Erweiterte Realität und 3-D Visualisierung für minimal-invasive Operationen durch Einsatz eines optischen Trackingsystems. In H. Meinzer, H. Handels, A. Horsch, T. Tolxdorff, eds., *Proceedings Bildverarbeitung für die Medizin*, 217–221. Springer, Berlin (2005). (8.2)

198. S. Vogt, A. Khamene, F. Sauer, H. Niemann. Single Camera Tracking of Marker Clusters: Multiparameter Cluster Optimization and Experimental Verification. In *Proceedings of the International Symposium on MIxed and Augmented Reality (ISMAR)*, 127–136 (2002). (5.1.4, 5.3.2, 5.3.3, 5.4.1, 5.4.2)

199. A. Vorozcovs, A. Hogue, W. Stuerzlinger. The Hedgehog: a novel optical tracking method for spatially immersive displays. In *Proceedings of IEEE Virtual Reality*, 83–89 (2005). (6.3)

200. M. Wagner. *Tracking With Multiple Sensors*. Ph.D. thesis, TU München (2005). (1.2, 2, 2.1.1, 2.1.3, 5.4.3, 9)

201. M. Wagner, A. MacWilliams, M. Bauer, G. Klinker, J. Newman, T. Pintaric, D. Schmalstieg. Fundamentals of Ubiquitous Tracking. In *Second International Conference on Pervasive Computing, Hot Spots section*. Vienna, Austria (2004). (2)

202. E. Wan, R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 153–158 (2000). (3.4.5)

203. J. Wang, R. Azuma, G. Bishop, V. Chi, J. Eyles, H. Fuchs. Tracking a head-mounted display in a room-sized environment with head-mounted cameras. *Proc. SPIE*, 1290:47–57 (1990). (6.3)

204. J.-F. Wang. *A real-time optical 6d tracker for head-mounted display systems.* Ph.D. thesis, Univ. of North Carolina Chapel Hill (1990). (5.4.1, 6.3)

205. C. Ware, R. Balakrishnan. Reaching for Objects in VR Displays: Lag and Frame Rate. *ACM Transactions on Computer-Human Interactions*, 1(4):331–356 (1994). (5.1.1)

206. G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, D. Colucci. High-Performance Wide-Area Optical Tracking: The HiBall Tracking System. *Presence: Teleoperators & Virtual Environments*, 10(1):1–21 (2001). (6.3)

207. G. Welch, E. Foxlin. Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *IEEE Computer Graphics and Applications*, 6(22) (2002). (1.2, 2)

208. P. Wellner, W. Mackay, R. Gold. Back to the real world. *Commun. ACM*, 36(7):24–27 (1993). (1, 1.1)

209. J. Weng, P. Cohen, M. Herniou. Camera Calibration with Distortion Models and Accuracy Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980 (1992). (5.1.3)

210. C. Wengert. Estimated fiducial location error for an Atracsys Easytrack 500 Optical Tracking System (2006). Personal communications. (9)

211. C. Wengert. Error Propagation for Hand-Eye Calibration (2007). Personal communications. (7.2.5)

212. C. Wengert, P. C. Cattin, J. M. Duff, G. Székely. Markerless Endoscopic Registration and Referencing. In R. Larsen, M. Nielsen, J. Sporring, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, *LNCS*, vol. 4190, 816–823. Springer (2006). (8.2)

213. C. Wengert, M. Reeff, P. C. Cattin, G. Székely. Fully Automatic Endoscope Calibration for Intraoperative Use. In *Bildverarbeitung für die Medizin*, 419–23. Springer-Verlag (2006). (8.2)

214. J. West, J. Fitzpatrick. Point-based rigid registration: clinical validation of theory. *Proceedings of SPIE*, 3979:353 (2003). (6.4, 9)

215. J. B. West, J. M. Fitzpatrick. The Distribution of Target Registration Error in Rigid-Body, Point-Based Registration. In *IPMI '99: Proceedings of the 16th International Conference on Information Processing in Medical Imaging*, 460–465. Springer-Verlag, London, UK (1999). (5.3.3)

216. J. B. West, J. M. Fitzpatrick, S. A. Toms, C. R. Maurer, R. J. Maciunas. Fiducial point placement and the accuracy of point-based, rigid-body registration. *Neurosurgery*, 48:810–817 (2001). (5.3.3, 5.4.3, 6.1)

217. J. B. West, C. R. Maurer. Extension of target registration error theory to the composition of transforms. In M. Sonka, J. M. Fitzpatrick, eds., *Proc. SPIE Vol. 4684, p. 574-580, Medical Imaging 2002: Image Processing, Milan Sonka; J. Michael Fitzpatrick; Eds.*, 574–580 (2002). (4.2.2, 5.3.3)

218. J. B. West, C. R. Maurer. Designing optically tracked instruments for image-guided surgery. *IEEE Transactions on Medical Imaging*, 23(5):533–545 (2004). (5.3, 5.3.3, 6, 6.1, 6.1.5, 6.1.6, 8.2)

219. T. Weyrich. *Entwicklung eines Kopfverfolgungssystem auf der Basis einer Panoramakamera und künstlichen Landmarken.* Master's thesis, Universität Karlsruhe (2001). (3.4.4, 5.2.2, 6.3)

220. T. Weyrich, T. Salb, R. Dillmann. Using a Panoramic Camera for 3D Head Tracking in an AR Environment. In *Proceedings of 9th IEEE Int. Conference on Mechatronics and Machine Vision in Practice*. Chiang Mai, Thailand (2002). (5.2.2, 6.3)

221. R. T. Whitaker, C. Crampton, D. E. Breen, M. Tuceryan, E. Rose. Object Calibration for Augmented Reality. *Computer Graphics Forum*, 14(3):15–28 (1995). (7.1.1)

222. A. Wiles, D. Frantz, D. Swart, *et al.* NDI Accuracy Assessment Kit Guidelines. Tech. rep., Northern Digital (2003). (5.2.1)

223. A. Wiles, D. Thompson, D. Frantz. Accuracy assessment and interpretation for optical tracking systems. *SPIE Medical Imaging*, 5367 (2004). (5.2.1, 5.3.2)

224. H. J. Woltring, R. Huiskes, A. de Lange, F. E. Veldpaus. Finite centroid and helical axis estimation from noisy landmark measurements in the study of human joint kinematics. *Journal of Biomechanics*, 18(5):379–389 (1985). (5.3.3, 6.1.2, 6.1.3)

225. R. Yarlagadda, I. Ali, N. Al-Dhahir, J. Hershey. Geometric dilution of precision (GDOP) bounds and properties. Tech. Rep. 97CRD119, General Electric (1997). (5.2.3)

226. R. Yarlagadda, I. Ali, N. Al-Dhahir, J. Hershey. GPS GDOP metric. *IEEE Proceedings on Radar, Sonar and Navigation*, 147(5):259–264 (2000). (5.2.3)

227. J. S. C. Yuan. A general photogrammetric method for determining object position and orientation. *Robotics and Automation, IEEE Transactions on*, 5(2):129–142 (1989). (5.4.1)

228. X. Zhang, S. Fronz, N. Navab. Visual Marker Detection and Decoding in AR Systems: A Comparative Study. In *First IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*. Darmstadt, Germany (2002). (5.4.1, 5.16, 5.4.2, 5.4.2)

229. G. Zhou. Accurate determination of ellipse centers in digital imagery. *American Congress on Surveying and Mapping and American Society for Photogrammetry and Remote Sensing, Annual Convention, Washington, DC*, 256–264 (1986). (5.3.2)

230. H. Zhuang, Z. Qu. A new identification Jacobian for robotic hand/eye calibration. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(8):1284–1287 (1994). (7.2.4, 7.2.5)

231. H. Zhuang, Z. S. Roth, Y. C. Shiu, S. Ahmad. Comments on 'Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB' [with reply]. *Robotics and Automation, IEEE Transactions on*, 7(6):877–878 (1991). (7.2.3)

232. H. Zhuang, Y. C. Shiu. A Noise Tolerant Algorithm For Wrist-mounted Robotic Sensor Calibration With Or Without Sensor Orientation Measurement. In *Proceedings of the 1992 lEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 1095–1100 (1992). (7.2.4)

233. H. Zhuang, K. Wang, Z. S. Roth. Simultaneous calibration of a robot and a hand-mounted camera. *Robotics and Automation, IEEE Transactions on*, 11(5):649–660 (1995). (7.2.4)

234. W. Zylka, J. Sabczynski. Effect of localization devices and registration methods on the accuracy of stereotactic frame systems predicted by the Gaussian approach. *Computer Aided Surgery*, 4(2):77–86 (1999). (5.3.2)

Index