Technische Universität München
Fakultät für Informatik

Diplomarbeit in Informatik

Predicting the Accuracy of Optical Tracking Systems

Michael Schlegel

Technische Universität München
Fakultät für Informatik

Diplomarbeit in Informatik

Predicting the Accuracy of Optical Tracking Systems

Michael Schlegel

Aufgabensteller:	Univ-Prof. Gudrun Klinker, Ph.D.
Betreuer:	Dipl.-Inform. Martin Bauer
Abgabedatum:	17. Juli 2006

Erklärung

Ich versichere, dass ich diese Ausarbeitung der Diplomarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 17. Juli 2006 Michael Schlegel

Zusammenfassung

Optisches Tracking ist eine Technologie, um Objekte mit Hilfe von optischen Kameras im Raum fortlaufend zu lokalisieren. Dabei kann sowohl die Position als auch die räumliche Lage bestimmt werden. Optisches Tracking ist eine Basistechnologie, welche in vielen Einsatzgebieten zur Anwendung kommt. So wird optisches Tracking in der *Augmented Reality* benutzt, um reale Objekte mit virtueller Information zu erweitern (engl.: *to augment*). Damit diese Erweiterungen an den richtigen Stellen platziert werden können, müssen Position und Blickrichtung des Beobachters und von den zu erweiternden Objekten bestimmt werden.

Um nun eine genaue Verbindung zwischen realen und virtuellen Objekten herzustellen, ist es nötig die Messungen so präzise wie möglich durchzuführen, aber leider tritt selbst bei grösster Sorgfalt und Präzision immer Rauschen in den Messungen auf. Dieses Rauschen in der Kamera kann zu beträchtlichen Fehlern in der Lokalisierung von Objekten führen. In bestimmten Anwendungen ist es nötig diese Fehler im Voraus und zur Laufzeit zu kennen und zu minimieren. Als Beispiel sei hier die Medizintechnik genannt.

Das Rauschen in den Messungen kann durch ein gaußsches Fehlermodell beschrieben werden. Zusammen mit dem Fehlerfortpflanzungsgesetz erlaubt es beispielsweise aus dem Rauschen auf der Bildebene der Kamera auf die Positionsungenauigkeit schliessen.

Mit Hilfe des gaußschen Fehlermodells und des Fehlerfortpflanzungsgesetzes wurden mehrere mathematische Modelle entwickelt, die es erlauben ausgehend von Rauschfehlern in einzelnen Messungen auf die zu erwartende Gesamtungenauigkeit zu schliessen. Das Fehlerfortpflanzungsgesetz wird in dieser Arbeit sehr häufig benutzt, unter anderem auch für die zwei entwickelten Anwendungen, welche die Genauigkeiten eines Trackingsystems mit mehr als zwei Kameras quantitativ schätzen können.

Abstract

Optical tracking is a technology to localize the position and rotation of objects continuously with the help of optical cameras. Optical tracking is a basic technology used in many applications. In *augmented reality* tracking is used to augment real objects with virtual informations. To place the informations at the right position, the position and viewing direction of the observer and that of the objects to be augmented have to be determined.

To determine an accurate relationship between real and virtual object it is necessary to take the measurements as precise as possible, but even with great diligence and precision noisy errors will be inevitable. The noise on the image plane of the camera can lead to considerable errors in the localization of the objects. In certain applications for instance in the medical domain it is necessary to know and minimize these errors a priori and at runtime

The noise in measurements can be described with a gaussian error model. Together with the error propagation rules it is for instance possible to predict the localization error.

With the help of the gaussian error model and the error propagation rules several mathematical models are developed which allow to predict the expected uncertainty based on noise errors. The error propagation is widely used in this thesis eg. for the two developed applications which can quantitatively predict the accuracy of an optical tracking system with more than two cameras.

Contents

1 Introduction

In this chapter a short overview over the context and the goals of this diploma thesis will be given.

1.1 Optical Tracking

Optical tracking is a technology to localize the position and rotation, also called pose, of objects continuously with the help of optical cameras. Therefore optical tracking systems are taking pictures and extracting features from the pictures which allow the determination of the pose. Without placing artificial markers on the object this is a challenging task, so most optical tracking system use distinctive markers to ease the feature extraction from the image.

As the extraction of the features and the measurements in the camera can only be done with errors, the estimated pose will also be erroneous. There are several causes for errors for instance when the camera produces distorted pictures. Some kinds of errors can be reduced or event eliminated but others are inevitable. The noise occurring on the sensor of the camera is an example for such an inevitable source of error.

The user of an optical tracking system desires accurate results and so the errors affecting the estimation of the pose have to be reduced. In some applications, for instance in the medical domain it is necessary to know these errors a priori and at runtime.

1.2 Augmented Reality

Augmented Reality is a technique which combines the real world and the virtual world. The user of an augmented reality system can see the real world with virtual objects superimposed upon or composited with the real world. This can be achieved by using head-mounted displays (HMDs) which allow the user to see the real world and to add additional computer generated visual informations. A more general definition is given by Azuma [5, 6] where a augmented reality system has to fulfill the following criteria:

1. Combines real and virtual
2. Interactive in real time
3. Registers (aligns) real and virtual objects

The first point emphasizes that the real and the virtual world have to be combined, which excludes pure virtual applications. The second criterion claims that the system has to be interactive for the user. The third criterion demands that the virtual objects have to be aligned with the real world. As the third criterion is not easy to understand for those people who never heard of augmented reality an example will be given. Caudell et al. are describing an example for augmented reality in fabrication of an airplane [11]. An user has to drill a hole in a workpiece and a HMD is used to visualize the location. So the user looks at the workpiece and sees the exact 3D location indicated by a green arrow along with the drill size and depth of the hole specified in a text window next to the arrow. As the user walks around the workpiece the arrow will be kept aligned with the desired drill location. This example fulfills Azuma's criteria. First the reality is augmented with virtual objects, the user can interact with the system and the virtual objects are aligned with real objects. The alignment of the virtual objects with real objects on the HMD requires several informations. First the pose of the HMD and second the pose of the real world object to which the virtual object should be aligned have to be known. This relationship is called the *registration* and the process of keeping this registration up to date is termed *tracking*. With help of the pose data determined from the tracking system the system can add the virtual object at the right position and orientation, even if the user moves his head or is walking around.

More about augmented reality can be found in a comprehensive survey by Azuma et al. [6].

1.3 Goals of this Thesis

The goal of this thesis is the development of a mathematical framework, which allows to predict the expected accuracy of an optical tracking system based on errors caused by noise.

The noise error can be described with a gaussian error model. With the help of error propagation rules it is possible to predict the expected error based on the noise in the measurements. For example the features of the tracked object are measured with noise on the sensor of the camera in two dimensions. When using multiple cameras, all affected by noise, the estimated spatial position of the object is also noisy. Knowledge of the noise in estimating the 2D position on the sensor can the be used to predict the noise of the spatial position.

The users of optical tracking systems want to know how accurate their specific tracking system is in the working volume which is crucial for eg. in medical applications. Manufacturers of optical tracking systems want to predict the accuracy of the tracking system without assessing real setups. Designers or users want to optimize the setup with respect to accuracy and costs. For example the accuracy of an optical tracking system can be increased merely by adding more cameras, but this would also increase the monetary costs of the system. With help the of developed framework different tracking setups can be assessed by changing the basic features of the system as there are the positions, the kind and the quantity of cameras used.

Therefore a framework was developed for providing mathematical procedures to predict the expected accuracy of different types of optical tracking systems, as there are n-ocular and monocular tracking systems. The framework was developed in a computer algebra system allowing to adjust the methods for new purposes easily.

Based upon this mathematical framework two small applications were developed to visualize the predicted accuracy. The first tool visualizes the positional accuracy of tracking a position with multiple cameras and the second tool visualizes the positional and rotational accuracy of optical marker targets.

There are other possible uses for this framework. For instance is it possible to predict the tracking accuracy at online during the tracking process runs. So users of safety-crucial application can be warned if the accuracy is getting beyond a lower bound.

1.4 Related Work

In this section an overview of the research done on the accuracy of optical tracking systems will be given. The first part is focused on the accuracy tracking single features with n-ocular systems. The second part addresses the accuracy of estimating the pose of optical marker targets with n-ocular and monocular systems.

1.4.1 Accuracy of n-ocular Tracking Systems

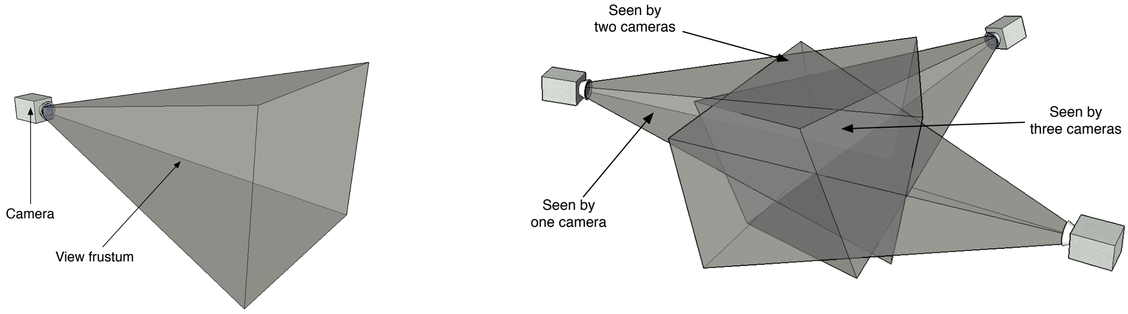
Chen et al. presented in [13] a quality metric for multi-camera setups. The uncertainty in estimating the 3D position of a feature is computed geometrically based on the uncertainty in estimating the 2D position on the image planes of the used cameras. The metric for positional uncertainty is combined with a metric considering occlusions to a overall metric. With the help of this overall metric the accuracy of the tracking setup can be determined based on the placement of the cameras and the 2D uncertainty of the cameras.

In [2] Allen et al. are presenting a technique for estimating the accuracy of measuring the feature positions with a tracking system. Therefore a steady-state model was used, which propagates the measurement uncertainty in the sensors back into the working volume. The approach of Allen et al. is also considering the errors caused by motion of the tracked object. An other aim of this work is to visualize the expected accuracies within the working volume graphically.

In [12] Cerfontaine et al. presents a method for automated optimal camera alignment for a n-ocular optical tracking system. The alignment of the cameras is optimized in respect to two goals, one is the widest possible working volume, the other is the maximum camera visibility. It is also possible to define constraints for the camera positions concerning restrictions of the local environment.

The volume of interest is defined by a set of positions p_k, which allows to define an arbitrary volume in a discrete way. Emphasizing certain regions is done by duplicating the corresponding points.

The cameras used for tracking are parameterized by the position, the orientation and a view frustum, which describes which volume can be seen by a camera. Such a view frustum can be seen in figure 1.1(a). The parameters describing the frustum are fixed for each camera, but the position and the orientation will be varied for optimizing the working volume and visibility.



(a) (b)

Figure 1.1: The view frustums of cameras are used for visibility determination; (a) shows the view frustum of a single camera; (b) shows the intersected frustums of three cameras

For a certain camera setup each position p_k will be tested for visibility of the different cameras. The results are stored in a $n \times m$ matrix, where n is the number of camera and m the number of positions. This matrix is then evaluated with a score, which indicates how close or far the result is from the final optimization goal. Cerfontaine et al. are using an approach minimizing the difference between the maximum score and the current score by using a discrete, gradient-based steepest descent method. The gradient vector is determined by finite differences for all variable parameters.

This technique is useful to place the cameras around the working volume while assuring the maximal visibility. Unfortunately the tracking accuracy is not only depending on the visibility, but also on the distance from the targets to the several cameras and the accuracy of locating features on the image plane, which are not covered by this approach.

1.4.2 Accuracy of Marker Targets

In [24] Hoff et al. presented a technique for predicting the 6 degree of freedom (DoF) pose accuracy of marker based targets. Based upon the accuracy of estimating the spatial position of markers, the pose accuracy is determined by using the covariance forward propagation with a first order approximation. Hoff is excluding errors in the target geometry, so it is assumed that the model of the target is known exactly. Another assumption is that only noise errors are occurring. Static errors are covered by this work.

As this method is described deeply in chapter 6, only a short overview will be given here.

The benefit of Hoff's approach is the use of covariance matrices for describing the positional and pose accuracy which allows to use un-isotropic and jointly distributed noise errors, but this technique does not consider static (biased) errors in contrast to the approach of Davis et al. [15]. The approach of Hoff et al. is also assuming that the target topology is known exactly, which is in general not the case, but the error can be kept small by diligent calibration of the target geometry.

Hoff is also presenting a method for transforming covariance matrices from one coordinate frame to another and for combining the covariance matrices of two pose estimations. Another achievement of this work is the presentation of methods for interpretation and visualization of positional and rotational errors represented by covariance matrices which will be used in section 3.3 and section 6.6. Examples for the visualization of errors represented by covariance matrices are given in section 7.3.

Davis et al. presented in [15] methods for predicting the expected error of tracking a marker target based upon a first-order propagation of the errors associated with the markers of the target. By applying an error term to each marker location this error is propagated through the pose estimation within a first-order approximation. The theoretical prediction of pose error presented is valid for any target topology with at least three non-collinear markers, with given marker coordinations x_k in local target coordinate frame and measured positions $y_k$in global coordinate frame.

Both positions x_k and y_k can be affected with errors Δx_k and Δy_k which are assumed to be isotropic The errors in the targets marker model Δx_k is either obtained from the manufacturer of the target, when using a commercial marker target, or from the target design process. The position estimating error Δy_k of a single marker can be estimated by measuring the root mean square error (see section 4.2.2) of detecting the position of a single marker.

The relationship between the true positions x_k and y_k can be expressed as

$$y_k = \mathbf{R}x_k + \mathbf{t} \quad , \tag{1.1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ a translation vector. Davis et al. used a pose estimation algorithm that determines $\mathbf{R}$ and $\mathbf{t}$ by minimization of least squares error.

To propagate errors, the positions are tainted with errors Δx_k and Δy_k, which can be written as

$$\begin{aligned} x_k &\rightarrow x_k + \Delta x_k \\ y_k &\rightarrow y_k + \Delta y_k \quad . \end{aligned} \tag{1.2}$$

The error in pose estimation is then

$$\begin{aligned} \mathbf{R}_{\text{err}} &\approx \Delta\mathbf{R}\mathbf{R} \\ \mathbf{t}_{\text{err}} &\approx \mathbf{t} + \Delta\mathbf{t} \quad , \end{aligned} \tag{1.3}$$

where $\Delta\mathbf{t}$ and $\Delta\mathbf{R}$ are the differential translation and rotation. Davis et al. developed a method to determine these differential variables $\Delta\mathbf{t}$ and $\Delta\mathbf{R}$ by propagating the error through the pose estimation algorithm.

The developed method then was tested with a Monte Carlo simulation, by evaluating the average pose error

$$E = \frac{1}{N} \sum_{k=1}^{N} \| y_k - \mathbf{R}_{\mathrm{err}} x_k - \mathbf{t}_{\mathrm{err}} \| \quad , \tag{1.4}$$

where N is the amount of markers in the target and y_k and x_k are true marker positions.

This average pose error function fuses the rotational and positional error into a one-dimensional value which does not allow to distinguish between the rotational and positional error. For example Davis examined the relationship between the size of the target and the resulting avg. pose error. It is claimed that the avg. pose error is decreasing with increasing marker size, which is only true for the avg. total pose error. The rotational error is influenced by the size of the target, but the positional part of the pose error depends only on the amount of feature points. This correlation is discussed more deeply in chapter 6.

An advantage of this error predicting model is that it takes care of errors in the target topology, where other models like [24] do not consider this kind of errors.

Vogt et al. presented in [53] a system for monocular optical tracking with planar and non-planar arrangement of the features. The pose estimation accuracy of the tracking system was analyzed with a Monte Carlo simulation and afterwards the results of the Monte Carlo simulation were experimentally verified.

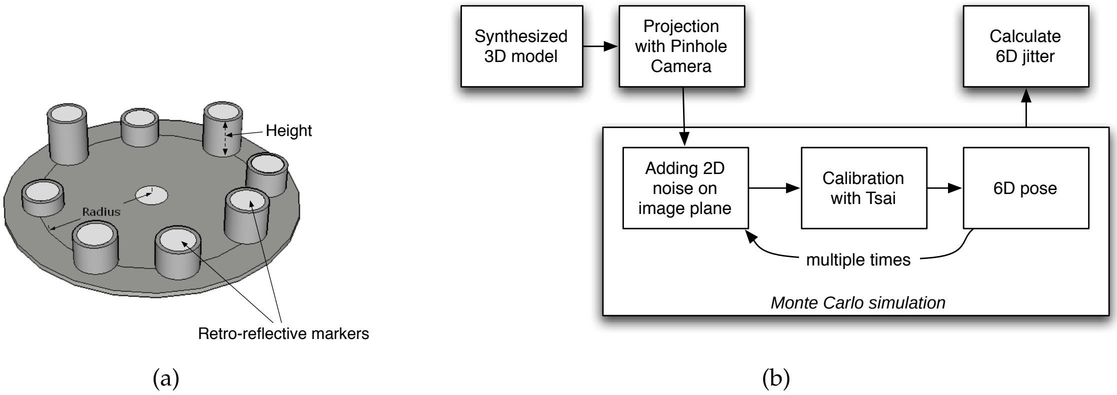
Vogt et al. are assuming an unbiased error (cf.4.1), because the parameters affecting the biased error can be pre-calibrated off-line with high degrees of accuracy. So they are assuming zero-mean and some standard deviation in estimating a 2D position on the image plane. The unbiased error is also called jitter.

As a first step the 2D jitter detecting a single marker on the image plane was experimentally estimated. Therefore a single marker was tracked with a ARTtrack1 camera which uses infrared light to detect retro-reflective markers. The 2D position of the marker was estimated by calculating the center of gravity of the grey pixels on the image plane. The standard deviation σ of the measured center of gravity was consistently varying from 0.013 $(\approx \frac{1}{77})$ to 0.010 $(= \frac{1}{100})$ pixel.

Then several target configurations were analyzed with a Monte Carlo simulation. In the tested configurations, the disc shaped retro-reflective markers are arranged on the perimeter of a circle in various heights relative to the plane of the circle. An example for such a target can be seen in figure 1.2(a). The simulation process starts with a synthesized target model consisting of 3D points p_i, which are then projected with a modeled camera to 2D positions q_i on the image plane. The q_i are then affected with the previous estimated noise σ of detecting a single marker, which leads to error affected measurements $\tilde{q}_i$. With Tsai's calibration technique [50] the pose p of the target can be calculated from the noise affected points $\tilde{q}_i$. The last two steps were repeated several times to determine the standard deviation of the pose estimation. A diagram showing the Monte Carlo simulation in principle can be seen in figure 1.2(b).

Vogt et al. are summarizing that accuracy is increasing, as the size of the target, the number of markers, and the variation in markers heights increase. The comparison between coplanar and non-coplanar target configurations showed that a non-coplanar target delivers a much more consistent error distribution that is comparable with the best case scenario of a coplanar target.

The presented method for analyzing the pose accuracy with Monte Carlo simulation can also be used for n-ocular tracking system. In section 6.4 the accuracy of a marker target was analyzed in the same way.



(a) (b)

Figure 1.2: Monte Carlo analysis of a target using monocular tracking; (a) shows the analyzed target; (b) shows how the Monte Carlo simulation was done

Abawi et al. developed in [1] an accuracy function for monocular optical tracking with the ARToolKit tracking system [31]. The accuracy function assesses the accuracy of ARToolKit measurements depending from distance to target and from the orientation of the target.

To determine the accuracy function a specific ARToolKit setup was studied with respect to accuracy. Therefore Abawi et al. used a standard commercial web cam and the ARToolKit pattern *sample1*. The camera was fixed mounted on a mechanical jig and the pose of the marker was varied in distance and orientation, where the center of the marker is kept on the principal axis of the camera. The resulting accuracy function is regarding both systematic (biased) error and the noise (jitter) error.

A drawback of the presented method is the lack of portability, because it uses the errors from one specific setup, which consists of camera, kind of target, light conditions and others, to determine the accuracy of other setups. However, the user of the ARToolKit tracking system can gain an idea of the expected errors.

2 Tracking Systems

This chapter gives an overview which criteria can be used to classify the performance of tracking systems and which common types of tracking technology are existing.

2.1 Criteria for Tracking Performance

The following criteria [8] can be used to classify a tracking system, so the right system can be chosen for a specific application.

Resolution The smallest change of position and orientation of the tracked object which can be measured by the tracking system is termed *resolution*.

Accuracy The error in position and orientation of the tracked object. This error is in general smaller than the resolution [42]. Because of the non-linear nature of optical projections the errors are not homogeneous in the tracking volume

Update Rate This is the rate at which the system delivers tracking information.

Lag If t_1 is the time at which the pose of the tracked object is changing and t_2 the time at which the new pose is reported by the system, then $\Delta t = t_2 - t_1$ is called *lag* or *latency*.

Working Volume The *working volume* is the volume within the tracking can be used with the specified resolution and accuracy

Costs Of course the costs of a system play an big role in the choice of which system should be used.

There is no perfect tracking system for an application, so balance between the different criteria has to be found.

2.2 Optical Trackers

Optical Trackers usually use vision based systems to estimate the position and orientation. This can be done by taking pictures with optical cameras and identifying features on the image plane. In general the projected features will result in points on the image plane.

Optical tracking systems can be categorized into two common groups [35]. *Marker-based* tracking systems use artificial visual features to help recognition in the taken image. The *Marker-less natural features-based* approach relies on features naturally present in the images without any supplementary features; natural features can be edges of buildings, a set of planar parts or the outline of the pupil against the iris for eye-tracking [28]. Marker-less tracking is still a challenging task which many researchers try to improve eg. [40]. The details of marker-less tracking are not covered by this thesis.

Marker-based tracking systems can be again classified in two common main groups. One uses special lighting conditions to improve the recognizability of the features, the other group tries to use the existing light conditions.

Tracking systems of the first group use infrared light to ease finding the features in the taken image. This can be achieved by illuminating the scene with infrared light and using retro-reflective markers, which are called *passive markers*. Or alternatively by using infrared emitting LEDs, which are termed *active markers*. The cameras used for tracking have an infrared filter mounted in front of the lens, so only the infrared spectrum can pass the filter and can get on the image plane. As the infrared intensity of the natural or artificial light of the environment is low against the emitted infrared light of the tracking system, the markers can be obtained more easily on the image plane because the contrast from the features to the rest of the scene is high. This can be seen in figure 2.1. When the image is taken, the contrast will be increased by thresholding the image, so unnecessary details of the image are widely eliminated, then the markers have to be found and identified which allows to calculate the 2D position of the markers on the image plane.

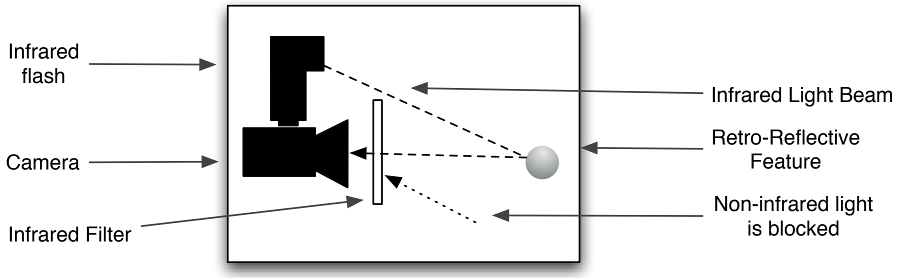
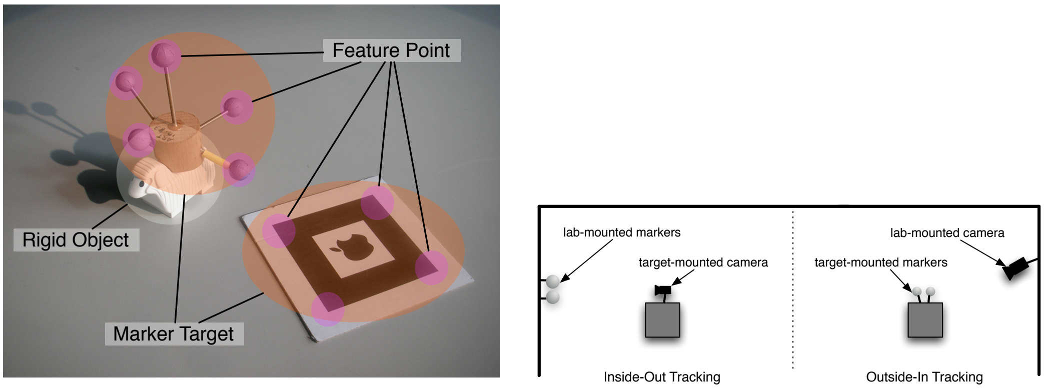


Figure 2.1: Principle design of an infrared tracking system

The second group of tracking systems is using natural lightning conditions to find the feature points of the target on the image plane. A target for such a system could be a black square on a white background providing a high contrast, so it can be more easily distinguished from the background. With methods from computer vision the edges of the square can be found and as the edges are intersecting in four points, so four feature points have been found. Images for such a target can be seen in the figures 1.2(a) and 5.9. The marker targets of monocular systems are designed in an almost flat manner because this reduces the possibility that a feature is hidden by another. More details for tracking accuracy of monocular systems using marker targets can be read in section 5.2.

A tracking system using n cameras is called a *n-ocular tracking system*. Assuming the positions, orientations and internal parameters of all n cameras are known, then the n 2D positions of a feature on the image planes can be used to triangulate the spatial position of the feature. So it is possible to track the 3D positions of single markers in the working volume. To obtain the orientation of the tracked object at least three feature points are necessary. These features have to be rigidly mounted on the object, because the geometrical relation of the features has to be known by the tracking system. This set of rigidly mounted features is termed a *marker target* or shortly a *target* and can be seen in figure 2.2(a). Once the positions of the single features are estimated, the position and the orientation of the whole target can be calculated by a 3D-to-3D pose estimation algorithm such as Horn [27]. In section 5.1 this topic will be discussed more deeply.



(a) (b)

Figure 2.2: (a) shows marker targets consisting of several single feature points; (b) shows Inside-Out and Outside-In tracking configuration

If the tracking system uses only one camera it is called a *monocular tracking system*. With a monocular system it is not possible to estimate the spatial position of a single feature, there has to be a set of rigidly mounted features with well known geometry which then allows also to estimate the complete pose. In figure 1.2(a) such a target can be seen.

Two system configurations can be distinguished. When tracking *inside-out* the cameras are mounted on the rigid object and the markers are fixed mounted in the laboratory. In the corresponding *outside-in* tracking configuration the cameras are mounted fixed in the laboratory and the markers on the target object. The inside-out configuration has some drawbacks, as it is for example not possible to mount heavy weighted cameras on the rigid object.

2.3 Inertial Trackers

Inertial systems are using accelerometers for measuring positional acceleration and gyroscopes for rotational accelerations. The two devices have to be mounted on the target object and the accelerations have to be transmitted to a tracking system. Unfortunately this technique allows only to measure changes of the pose, so the target object should have a well known pose at the beginning of the tracking process. Measuring relative changes of the pose leads to cumulative errors, because errors in estimating pose changes over time are summarized. So the estimated pose is drifting away from the real pose over time. More details like technical basics of this tracking technique and its appliance in augmented reality can be found in [42, 9, 61, 34].

An example for inertial tracking is the InertiaCube2 system from InterSense[1].

Advantages Inertial sensors are independent from any infrastructure. The tracking volume has no constraint, because not fixed devices are needed; so inertial systems do allow an wide-area use. Modern inertial sensors are small, light of weight and inexpensive. No external influence is affecting the measurement except changes in gravity.

Disadvantages The measurement will drift over time because only relative changes are detected. So some kind of ground truth has to be used. This could be a combination with another tracking system [61] or a well defined mechanical determined pose.

2.4 Magnetic Trackers

Magnetic tracker technology uses magnetic fields to estimate the position and orientation, therefore an emitter and a detector is needed. The sender is at a fixed and well known position and the receiver is mounted on the tracked object. Two types of magnetic trackers can be distinguished. The *alternate current* or short *A.C.* emits three mutually perpendicular electromagnetic coils. When an A.C. signal is applied to the sender a rotating magnetic field will be emitted which induces currents in the receiver. Similar to the A.C. technology the *direct current* or short *D.C.* sender emits three perpendicular coils but short pulses are supplied to the sender. The advantage of D.C. is that the measurements are only distorted by ferromagnetic materials. In contrast the A.C. trackers are distorted by any kind of metal material [8, 9].

An example for a product which uses magnetic tracking technology is the Aurora System from Northern Digital (NDI)[2].

Advantages The sensors are light and small and therefore comfortably to wear or easily mounted on objects. Magnetic trackers are not affected by obstructed line-of-sights. So there can be objects between the sender and the receiver, except these object contain ferromagnetic or metal materials.

Disadvantages As mentioned before, the magnetic tracking technology can be distorted by metal or ferromagnetic things, even earth's magnetic field is affecting the measurement. The negative effect of static magnetic fields can be avoided by measuring the static field and subtracting it from the following measurements. The influence of non-static electromagnetic fields like CRT monitors can not be eliminated easily.

2.5 Acoustic Trackers

Acoustic trackers use ultrasonic waves to estimate the position of the target object. As the time, a sound signal travels from a sender to a receiver, is determined only by speed of

---

[1] Intersense: `http://www.isense.com/`

[2] Northern Digital (NDI): `http://www.ndigital.com`

sound in air, the distance of receiver and sender can be estimated by measuring the *time-of-flight*. For tracking a 3D position it is necessary to determine the distance from the target object to three different well known positions. Usually the sender of the ultrasonic waves is mounted on the target object and the three receivers are mounted at fixed known positions. Considering the relative low speed of sound in the air the update rate is limited. For example let the distance be $d = 10\text{m}$ and the sound speed $c = 343\frac{\text{m}}{\text{s}}$ then the update rate can not be greater than $f = \left(\frac{d}{c}\right)^{-1} \approx 34.3\text{Hz}$.

An example for acoustic tracking is the IS-600/900 system from Intersense.

Advantages The sender and receiver for acoustic tracking are small and light in weight, so it is comfortable for the users. The hardware is inexpensive and the tracking environment has not to be designed specially. The acoustic tracking is not influenced by external effects like magnetism or metal materials.

Disadvantages The acoustic energy diminishes with the square of the distance, so large working volumes are hardly to realize. The distance from target and sender is also limiting the update rate. As the speed of sound varies with the temperature of air and with some other physical conditions, the sound speed can not be treated constantly. This can be adjusted by measuring the speed of sound by a separate sender and receiver pair with fixed and well known distance. Another drawback is the constraint of unobstructed line-of-sight. If there is an object between the sender/receiver pair the signals are traveling not directly to the receiver but with reflection. The time-of-flight is then longer which leads to a wrong distance estimation. The acoustic tracking allows only the tracking of positions and does not provide rotational informations, therefore the IS-600/900 system from Intersense includes also an inertial tracking system, which measures the orientation of the target object.

2.6 Mechanical Trackers

Mechanical tracking is using a movable mechanical arm to measure the position and the orientation of the target object. The arm is fixed mounted to a well known position and it is sectioned in several parts. The joints are applied with potentiometers to register the movements, which allows to calculate the angles between the sections and therefore the position and orientation of the arm's tip.

The mechanical tracking technology is not common in augmented reality applications, because the disadvantages below outbalance the advantages. In some applications like finger tracking, mechanical tracking is still popular. An example for commercially produced mechanical tracker is the Shooting Star ADL-1, used in [60] for head tracking purposes. More details for mechanical tracking can be read in [8] or in [9].

Advantages Mechanical tracking delivers measurements with high accuracy, high resolution, low latency and high update rate.

Disadvantages The movement of the rigid object is constrained by the mechanical arm and has not full freedom. The mechanical arm is also limiting the working volume and a bigger volume would cause a new arm. In addition the arm suffers from mechanical wear.

2.7 Global Navigation Satellite Trackers

A *global navigation satellite system* uses satellites for tracking 3D positions. With at least four satellites the position can be estimated by measuring the time-of-flight of radio-magnetic signals from the satellites to a receiver. As the receiver needs an unobstructed line-of-sight to at least four satellites it can not be used indoors or in some urban areas with high buildings which are hiding the satellites. Further details can be found in [26].

Examples for global navigation satellite systems are the *Global Positioning System* (GPS) and the *Global'naya Navigatsionnaya Sputnikovaya Sistema* (GLONASS).

Advantages The system is available globally and its use is free of charge. No special hardware environment is needed. The receiver is small and inexpensive.

Disadvantages The resolution of the GPS system is low and the accuracy without technical improvements is about 4m up to 20m. The accuracy can be enhanced over short distances up to 1cm, but this improvement needs a technical infrastructure.

3 Mathematical Basics

In this chapter an overview of the mathematical instruments used in this thesis will be given.

3.1 Rotations

In many applications it is necessary to track not only the position but also the rotation of an object. The combination of position and orientation is called a pose. Unfortunately there are many representations of rotation which are all leading to a different representation of the pose. The most common representations of rotation are *rotation matrices*, *euler angles* and *quaternions* which will be now introduced.

3.1.1 Matrices

Describing rotations as matrices is common in computer graphics. When operating in 3 dimensional euclidian space rotation matrices are in the special orthogonal group $\mathrm{SO}(3)$ [18]. If a matrix R is element of $\mathrm{SO}(3)$ then it has the following properties:

1. There is a matrix $R^{-1} \in \mathrm{SO}(3)$ with $R^{-1}R = I$
2. $R^t = R^{-1}$
3. $\det(R) = 1$

So inverting a rotation matrix can be easily done by transposing the matrix, which saves much computation time.

By extending a rotation matrix R to a homogeneous matrix $\tilde{R} \in \mathrm{GL}(4)$ translation, rotation can be done with $\tilde{R}$. Rotating and then translating a point can be written as a affine map $x \mapsto Rx + b$ where $R \in \mathrm{SO}(3)$ and $b \in \mathbb{R}^3$. This affine map can also be written as a homogeneous matrix

$$\begin{pmatrix} R & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} Rx + b \\ 1 \end{pmatrix} \quad .$$

It can be shown that $\tilde{R}$ is invertible if and only if the rotational part R is invertible. The inversion of a homogeneous matrix can be done by the next formula

$$\begin{pmatrix} R & b \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} R^{-1} & -b \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R^t & -b \\ 0 & 1 \end{pmatrix} \quad .$$

A drawback of matrices is, when two matrices of floating points are multiplied in a computer system, the result will not necessarily be an orthogonal matrix due to numerical errors. So the result has to be tested for orthogonality and if necessary has to be made orthogonal.

3.1.2 Euler Angles

Another way to represent rotation are Euler Angles, which are describing the rotation around the coordinate axes. There are several conventions of Euler Angles, representing different axes which are used for the rotation. They can be devided into two commonly used groups. In the first group the rotation axes are not affected by the previous rotations so they are fixed, in the other the axes are also rotated. In this thesis only fixed Z-Y-X angle setting will be used, but many other ways of euler angles would be possible. A comprehensive overview is given by Craig [14].

The fixed Z-Y-X angle setting rotates first around the x axis by the angle α, then around the y axis by the at angle β and at last around the z axis by the angle γ.

The rotations around the three axes written as matrices are

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} R_y = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} R_z = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Combining R_x R_y and R_z leads to

$$R = R_x R_y R_z = \begin{pmatrix} \cos(\beta)\cos(\gamma) & -(\cos(\beta)\ \sin(\gamma)) & \sin(\beta) \\ \cos(\gamma)\sin(\alpha)\sin(\beta) + \cos(\alpha)\sin(\gamma) & \cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma) & -\cos(\beta)\sin(\alpha) \\ \cos(\alpha)\cos(\gamma)\sin(\beta) + \sin(\alpha)\sin(\gamma) & \cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin\sin\sin(\gamma) & \cos(\alpha)\cos(\beta) \end{pmatrix} \quad . \tag{3.1}$$

3.1.3 Quaternions

The *quaternions* were first described by Hamilton in the year 1847 [20]. A Hamilton quaternion q can be thought as a complex number with three different imaginary parts. Thus it can be written

$$q = iq_x + jq_y + kq_z + q_0 \quad , \tag{3.2}$$

where q_x, q_y and q_z are the imaginary parts and q_0 the real part. These four values can also be written as a vector $\tilde{q} = (q_0, q_x, q_y, q_z)^t \in \mathbb{R}^3$.

The norm of quaternions is defined like the euclidean norm

$$\|q\|^2 = q_x^2 + q_y^2 + q_z^2 + q_0^2 \quad . \tag{3.3}$$

A quaternion with norm $\|q\| = 1$ is called a *unit quaternion*.

To avoid scaling while rotation, quaternions have to be unit quaternions, which also allows to drop one value of the quaternion without loss of information. So a rotation using unit quaternion representation can be described with only three variables.

Further details for instance how to use quaternions for rotation can be read in for example in [27, 57] and many others.

3.2 Statistics

For describing errors in tracking some statistical methods are needed, which are introduced in the next sections. It is assumed that the reader is familiar with the basic statistical terms and definitions; if not, the reader can get more information about statistics in [17] and [32].

3.2.1 Errors

When tracking objects the measured poses will be always tainted with errors. In the following the basic statistical principles are introduced to describe the errors in a mathematical way.

A pose can be represented in many ways, which mainly differ in the representation of rotation. For example if taking fixed Z-Y-X Euler angles the pose p can be written as

$$p = (x, y, z, \alpha, \beta, \gamma)^t \in \mathbb{R}^6 \quad , \tag{3.4}$$

where x, y, z is the true position and α, β, γ the true rotation. Assuming a tracking system delivers a measured pose $\tilde{p} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^t \in \mathbb{R}^3$, then this measurement will be tainted with the error

$$\Delta p = (\Delta x, \Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)^t \quad . \tag{3.5}$$

This error vector can be treated as multi-dimensional random variable.

3.2.2 Multi-Dimensional Random Variable

It would be possible to model the error as a set of scalar random variables, but then all random variables would be independent from each other, which meets not the reality. So the errors have to be modeled as a multi-dimensional random variable.

3.2.3 Probability Density Functions

A probability density function is a function assigning probability to certain events.

Let X be a n-dimensional random variable and A a set of events. Then $\mathrm{P}(X \in A)$ is the probability that the random variable X reaches values in A. A probability density function f describes that coherence between events and probability of occurrence of that events.

3.2.4 Expectation

The expectation is the mean of a random variable. Let $X = (X_1, X_2, \ldots, X_n)$ be a random variable and

$$\mu_i = \mathrm{E}(X_i) = \int_{-\infty}^{+\infty} x_i f(x_i) dx_i \tag{3.6}$$

the scalar mean of the several random variables X_i. Then the expectation of the n-dimensional random variable X is

$$\mu = \mathrm{E}(X) = (\mu_1, \ldots, \mu_n)^t \quad . \tag{3.7}$$

3.2.5 Variance and Covariance

The *variance* is a statistical measure for dispersion, indicating how far the values of the scalar random variable are away from the expected value. It is defined only for scalar random variables. If $\mu = \mathrm{E}(X)$ is the mean of the random variable X, then the variance is

$$\mathrm{Var}(X) := \mathrm{E}((X - \mu)^2) \quad . \tag{3.8}$$

Another formula for calculating the variance of a random variable is

$$\mathrm{Var}(X) = \mathrm{E}(X^2) - (\mathrm{E}(X))^2 \quad . \tag{3.9}$$

The root of the variance is called the *standard deviation σ*

$$\sigma := \sqrt{\mathrm{Var}(X)} = \sqrt{\mathrm{E}(X - \mu)} \quad . \tag{3.10}$$

The advantage of the standard deviation over the variance is that it is measured in the same units as the original data, which is more intuitive for the user.

The variance is invariant under addition and linear under multiplication with scalars

$$\mathrm{Var}(aX + b) = a \, \mathrm{Var}(X) \quad . \tag{3.11}$$

For two scalar random variables the covariance ($(X, Y) \mapsto \mathrm{Cov}(X, Y) \in \mathbb{R}$) is a statistical measure how much the X and Y are varying together. For example the $\mathrm{Cov}(X, Y)$ becomes more positive for each pair X and Y which differ from their mean in the same direction and more negative with each pair which differ from their mean in opposite directions. If the covariance of X and Y is zero then these two random variables are called uncorrelated which are not necessarily be independent.

Let X and Y be to scalar random variables and $\mu_X = \mathrm{E}(X)$ and $\mu_Y = \mathrm{E}(Y)$ their expected values. Then the covariance of X and Y is

$$\mathrm{Cov}(X, Y) = \mathrm{E}((X - \mu_X)(Y - \mu_Y)) \quad . \tag{3.12}$$

It is easy to see that the variance of X is the same as $\mathrm{Cov}(X,X)$

$$\begin{aligned}\mathrm{Cov}(X,X) &= \mathrm{E}((X-\mu_X)(X-\mu_X)) \\ &= \mathrm{E}((X-\mu_X)^2) = \mathrm{Var}(X) \quad .\end{aligned} \tag{3.13}$$

Also easy to see is that the covariance is symmetric

$$\begin{aligned}\mathrm{Cov}(X,Y) &= \mathrm{E}((X-\mu_X)(Y-\mu_Y)) \\ &= \mathrm{E}((Y-\mu_Y)(X-\mu_X)) = \mathrm{Cov}(Y,X) \quad .\end{aligned} \tag{3.14}$$

For two multi-dimensional random variables X and Y with dimensions n and m there are $n \times m$ single covariances. These can be combined to one single matrix where $\mathrm{Cov}(X,Y)_{i,j} = \mathrm{Cov}(X_i, Y_j)$. If $X = Y$ with dimension n then $\mathrm{Cov}(X)$ is called covariance matrix and can be written as

$$\mathrm{Cov}(X) = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1,X_1) & \cdots & \mathrm{Cov}(X_1,X_n) \\ \mathrm{Cov}(X_1,X_2) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_n,X_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_1,Xn) & \mathrm{Cov}(X,X) & \cdots & \mathrm{Var}(X_n) \end{pmatrix} \quad . \tag{3.15}$$

The diagonal entries of the covariance matrix are the variances of X_i. It is also easy to be seen that the covariance matrix is symmetric because of the symmetry of the covariance function:

$$\mathrm{Cov}(X) = \mathrm{Cov}(X)^t \tag{3.16}$$

It can be proved that a covariance matrix is also always positive semi-definite, which means that all eigenvalues of $\mathrm{Cov}(X)$ are positive or zero [18].

3.2.6 Probability Density Functions

There are several probability density functions as there are normal distribution, Student's t-distribution and many more. Because only the normal distribution is needed in the following chapters the other distributions are not explained.

Normal Distribution

The most important distribution is the *normal distribution*, also called *gaussian distribution*. A scalar random variable can be called normally distributed with parameters μ and σ^2 if it has the density function

$$\mathrm{f}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad , \tag{3.17}$$

where the standard deviation is σ and the expected value is μ. If X is normally distributed, it can be shortly written as $X \sim \mathrm{N}(\mu, \sigma^2)$.

The normal distribution is symmetric around μ, which can be written as

$$\mathrm{f}(\mu - x) = \mathrm{f}(\mu + x) \quad \forall x \in \mathbb{R} \quad . \tag{3.18}$$

Multivariate Normal Distribution

There is also a standard deviation for multi-dimensional random variables. Let X be a multi-dimensional random variable with dimension n, $\Sigma \in \mathbb{R}^{n \times n}$ a symmetric, positive semi-definite matrix and $\mu \in \mathbb{R}^n$ then the multivariate normal distribution is given by

$$\mathrm{f}(x) = \frac{1}{\sqrt{2\pi}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)} \quad . \tag{3.19}$$

It can be shown that μ is the expected value of X and Σ its covariance matrix.

If X is multivariate normally distributed it can be written $X \sim \mathrm{N_n}(\mu, \Sigma)$. Similar to the scalar case.

If the covariance matrix Σ has only one n-fold eigenvalue the distribution is called an *isotropic* distribution because the standard deviations in each of the n dimensions are equal.

3.3 Interpretation of Covariance

In this section a method will be shown, how a multi dimensional normally distribution and the corresponding covariance matrix can be interpreted in a geometric manner. It will be shown that the set of points ($X \sim \mathrm{N_n}(\mu, \Sigma)$) with the same probability will form an ellipsoid.

3.3.1 Quadratic Forms

A *quadratic form* or shortly *quadric* is a special polynomial function, which can describe curves, planes and second order hyperplanes.

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix then

$$\mathrm{q} : \mathbb{R}^n \to \mathbb{R}, x \mapsto \mathrm{q}(x) = x^t A x \tag{3.20}$$

is called a quadratic form.

In the next, without loss of generality the dimension is restricted to $n = 3$, because it is easier for the reader to follow. The principles can be easily extended to the n-dimensional case.

Thus $A \in \mathbb{R}^{3 \times 3}$ is a symmetric matrix $\mathrm{q}(x)$ it depends only on six parameters. So $\mathrm{q}(x)$ can be written as a polynom of degree two.

$$\mathrm{q}(x) = \mathrm{q}(x_1, x_2, x_3) = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 + 2a_{13}x_1x_3 + 2a_{23}x_2x_3 + a_{33}x_3^2 \tag{3.21}$$

An ellipsoid is defined by

$$\frac{x_1^2}{a} + \frac{x_2^2}{b} + \frac{x_3^2}{c} = 1 \quad \text{where} \quad a, b, c > 0 \quad . \tag{3.22}$$

With $A = \mathrm{diag}(\frac{1}{a}, \frac{1}{b}, \frac{1}{c})$ the equation (3.22) can also be written as

$$\frac{x_1^2}{a} + \frac{x_2^2}{b} + \frac{x_3^2}{c} = x^t \begin{pmatrix} \frac{1}{a} & & \\ & \frac{1}{b} & \\ & & \frac{1}{c} \end{pmatrix} x = x^t A x \quad . \tag{3.23}$$

So the ellipsoid fulfills the definition of quadrics. The principal axis of such an ellipsoid are the three coordinate axes x_1, x_2 and x_3 and the length of these axes are $\sqrt{a}$, $\sqrt{b}$ and $\sqrt{c}$. Usually the principal axes of the ellipsoid do not coincide with the coordinate axes but are somehow rotated. This can be achieved by rotation of the coordinate system for instance with a rotation matrix R. This leads to

$$y = Rx \Rightarrow y^t A y = (Rx)^t A (Rx) = x^t (R^t A R) x \quad . \tag{3.24}$$

Lanczos shows that the principal axes are now along the rows of R with the same length as before [33]. So with help of a rotation and a diagonal matrix it is possible to define an arbitrary ellipsoid in space.

If an arbitrary symmetric and positive definite matrix B, for example a covariance matrix, is given, B can be decomposed with eigenvalue decomposition into a rotation matrix V with the eigenvectors as columns and a diagonal matrix D consisting of the eigenvalues. This is called the principal axis theorem [47]

$$B = VDV^t \quad . \tag{3.25}$$

Thus B is a symmetric matrix eq. (3.25) can also be written as

$$B = B^t = (VDV^t)^t = V^t D V \quad . \tag{3.26}$$

Using eq. (3.26) in eq. (3.24) it is clear that B is defining an ellipsoid. So each positive symmetric matrix defines an ellipsoid with the eigenvectors as the principal axes. The length of the principal axes are $\frac{1}{\sqrt{\lambda_1}}$, $\frac{1}{\sqrt{\lambda_2}}$ and $\frac{1}{\sqrt{\lambda_3}}$, where λ_i are the eigenvalues.

In the one dimensional case $n = 1$ the quadric leads to two points, if $n = 2$ an ellipse and for an arbitrary n a n-dimensional ellipsoid is defined.

3.3.2 Surface of Constant Probability

In the following section it will be shown that normal distributed events occurring with the same probability are lying on a quadric, which is termed the *surface of constant probability*.

Let $X \sim \mathrm{N}_n(0, \Sigma)$ be a n-dimensional random variable with zero-mean, so the density function of X can be written with eq. (3.19)

$$\mathrm{f}(x) = \frac{1}{\sqrt{2\pi}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}x^t\Sigma^{-1}x} \quad . \tag{3.27}$$

To get all events which have the same probability p, the eq. (3.27) has to be solved for x

$$x^t\Sigma^{-1}x = -2\ln(p\sqrt{2\pi}|\Sigma|^{\frac{1}{2}}) = z^2 \quad . \tag{3.28}$$

Because the multi variate distribution $\mathrm{N}(\Sigma, \mu)$ is not defined for singular covariance matrix Σ, it can be assumed without loss of generality that Σ is not singular, so the covariance matrix is positive definite. It is not hard to see that eq. (3.28) describes an n-dimensional ellipsoid (cf. section 3.3.1), so all events with the same probability are located on an ellipsoid in n dimensions.

To calculate the principal axes of the ellipsoid eq. (3.28) has to be rearranged

$$\Delta x^t\Sigma^{-1}\Delta x = z^2 \Leftrightarrow \Delta x^t\frac{1}{z^2}\Sigma^{-1}\Delta x = 1 \quad . \tag{3.29}$$

Because Σ is positive definite and symmetric it can be decomposed to $\Sigma = RDR^t$, where R is the matrix of eigenvectors and D the diagonal matrix of the corresponding eigenvalues. This leads to

$$\begin{aligned}
& & \Delta x^t\tfrac{1}{z^2}\Sigma^{-1}\Delta x &= 1 & (3.30)\\
&\Leftrightarrow & \Delta x^t\tfrac{1}{z^2}(RDR^t)^{-1}\Delta x &= 1 & (3.31)\\
&\Leftrightarrow & \Delta x^t\tfrac{1}{z^2}(R^tD^{-1}R)\Delta x &= 1 & (3.32)\\
&\Leftrightarrow & \Delta x^t(R^t\begin{pmatrix} \frac{1}{\lambda_1 z^2} & & \\ & \frac{1}{\lambda_2 z^2} & \\ & & \frac{1}{\lambda_3 z^2}\end{pmatrix}R)\Delta x &= 1. & (3.33)
\end{aligned}$$

So the principal axes are along the columns of R and have the length $l_i = z\sqrt{\lambda_i}$ $(i = 1, 2, 3)$ (cf. section 3.3.1).

In most applications it makes more sense to deal with a *cumulative probability* P, which is the probability of an event being inside the ellipsoid given by the covariance matrix Σ and the magnification factor z.

Let n be the dimension and P the wanted cumulative probability then the formula for cumulative probability is given by

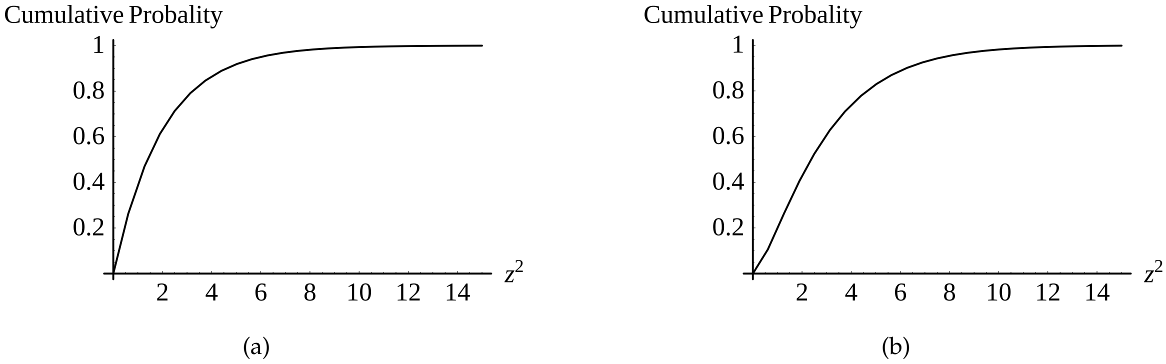
$$1 - P = \frac{n}{2^{n/2}\Gamma(\frac{n}{2}+1)} \int_{z}^{\infty} x^{n-1} e^{x^2/2} dx \quad [51]. \tag{3.34}$$

The cumulative probability can also be interpreted as a *confidence level*. For example it is 75% confident that events occur within the given ellipsoid.

For $n = 3$ dimensions, the ellipsoid defined by $z = 3$ corresponds to a confidence level of $\approx 97\%$. More confidence levels can be seen in table 3.1 and plot of the relationship of magnification and confidence level can be seen in figure 3.1.

	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: Magnification factor resulting from given dimension and confidence level



(a)

(b)

Figure 3.1: Relation between magnification factor z^2 and the confidence level; (a) shows the cumulative probability in two dimensions and (b) in three dimensions

3.4 Error Propagation

This section is about transforming random vectors. This kind of task is common in augmented reality. For example the error vector distribution of a pointing device is known in the centroid of the marker target, but it is desired at the tip of the pointing device. So the error vector has to be transformed with some kind of mapping, which is also affecting the distribution of the error vector. Therefore the covariance and the mean of the distribution have also to be transformed, which can be done with *covariance forward propagation* and *covariance backward propagation*.

3.4.1 Forward Propagation

The forward propagation of covariance is rule for affine function f as defined in [23]: Let $v \sim \mathrm{N}_M(\bar{v}, \Sigma)$ be a random vector with mean $\bar{v}$ and covariance matrix Σ. Let f be an affine mapping defined by $\mathrm{f} : \mathbb{R}^M \rightarrow \mathbb{R}^N, v \mapsto \mathrm{f}(v) = \mathrm{f}(\bar{v}) + A(v - \bar{v})$. Then $\mathrm{f}(v)$ is again a random variable with mean $\mathrm{f}(\bar{v})$ and a covariance matrix Σ_{f}

$$\Sigma_f = A\Sigma A^t \tag{3.35}$$

Often the mapping f is not linear, but the covariance forward propagation in eq. (3.35) is only defined for linear mappings. This Problem can be solved by a linear approximation at a certain point.

Let f be function differentiable in point v_0. Then the Taylor Series delivers a linearization

$$\mathrm{f}(v) = \mathrm{f}(v_0 + \Delta v) = \mathrm{f}(v_0) + \frac{\partial \mathrm{f}}{\partial x} |_{v=v_0} \Delta v + \cdots \tag{3.36}$$

where $\frac{\partial \mathrm{f}}{\partial x} |_{v=v_0}$ is the Jacobian matrix evaluated at the point v_0.

The Jacobian matrix is a matrix of all first-order partial derivatives of a function $\mathrm{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and it is of the following shape

$$\mathrm{J}_{\mathrm{f}}(v_1, \ldots, v_n) = \begin{pmatrix} \frac{\partial \mathrm{f}_1}{\partial v_1} & \cdots & \frac{\partial \mathrm{f}_1}{\partial v_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{f}_m}{\partial v_1} & \cdots & \frac{\partial \mathrm{f}_m}{\partial v_n} \end{pmatrix} \quad . \tag{3.37}$$

With the help of the Jacobian matrix $\mathrm{f}(v)$ can linearized around point v_0

$$\mathrm{f}(v) = \mathrm{f}(v_0 + \Delta v) \approx \mathrm{f}(v_0) + J\Delta v \tag{3.38}$$

The covariance matrix for the random variable f(v) in v_0 can be determined with the linearization for f and the linear forward propagation eq. (3.35)

$$\mathrm{Cov}(\mathrm{f}(v)) \approx \mathrm{Cov}(\mathrm{f}(v_0) + J\Delta v) = \mathrm{Cov}(J\Delta v) = J\,\mathrm{Cov}(v)J^t \quad . \tag{3.39}$$

3.4.2 Backward Propagation

Sometimes the distribution of the random variable $\mathrm{f}(V)$ is known as $\mathrm{f}(V) \sim \mathrm{N}_n(\mu, \Sigma)$, but the distribution of V is of interest. This problem could be solved by doing the forward propagation (3.39) of the inverse f^{-1}, but unfortunately it is often hard or even not possible to calculate such an inverse.

So instead of calculating the forward propagation $J_{f^{-1}}\Sigma J_{f^{-1}}^t$ with the inverse, it can be shown that is equal to compute [23]:

$$\Sigma_{\mathrm{f}^{-1}} = (J_{\mathrm{f}}^t \Sigma^{-1} J_{\mathrm{f}})^{-1} \tag{3.40}$$

Beside the advantage to calculate the new covariance matrix straightforward without the inverse of f, it provides also a solution of overparametrized cases [23].

Using the pseudoinverse $M^{\ddagger} := \left(M^t M\right)^{-1} M^t$ [57] leads to

$$\Sigma_{\mathrm{f}^{-1}} = (J_{\mathrm{f}}^t \Sigma^{-1} J_{\mathrm{f}})^{\ddagger} \quad . \tag{3.41}$$

To determine the complete distribution of $\mathrm{f}^{-1}(V)$, also the mean of f has to be calculated. Unfortunately there is no such easy way as for the covariance matrix, but in this thesis only distributions with zero-mean are used.

3.5 Camera Model

A camera model is a simplification of a real world camera which maps from 3D world to the 2D image plane. The mapping can be divided into two parts:

1. Transforming the points from the world coordinate frame into the coordinate frame of the camera.
2. Projecting the transformed points onto the image plane.

In this section the pinhole camera model, which is used widely in this theses, and some photogrammetric enhancements will be described.

3.5.1 Pinhole Camera Model

The *basic pinhole camera model* [23] is a projection of a point $\vec{x} = (x, y, z)^t$ in space onto a point $\vec{u} = (u, v)^t$ on the image plane, where the center of projection is at the origin of a Euclidean coordinate system and the image plane is assumed as the plane $z = f$. The mapping of a point is then done by intersecting the line connecting the origin $\vec{0}$ and $\vec{x}$ with the image plane. The line perpendicular to the image plane is called the *principal axis* and the point where the principal axis intersects the image plane is called *principal point* (cf. figure 3.2).

$$\mathrm{p} : \mathbb{R}^3 \to \mathbb{R}^2, \vec{x} \mapsto u = \mathrm{p}_f(\vec{x}) = \frac{1}{z} \begin{pmatrix} fx \\ fy \end{pmatrix} \tag{3.42}$$

This mapping can also be written with homogeneous coordinates and a matrix $\mathbf{P}$

$$p_f(\vec{x}) = \frac{1}{z}\mathbf{P}\mathbf{x} = \frac{1}{z} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad , \tag{3.43}$$

where $\mathbf{P}$ is called the *camera projection matrix*. In the later the matrix $\mathbf{P}$ is extended by more parameters than just the focal length to get a more capable camera model. The parameters can be divided into two groups where one is describing the spatial position and orientation of the camera and the other the camera itself. This kind of separation is also used for the projection matrix $\mathbf{P} = \mathbf{KT}$, where the matrix $\mathbf{T}$ represents the *extrinsic* position and orientation of the camera and the matrix $\mathbf{K}$ the *intrinsic* parameters of the camera like focal length and others.

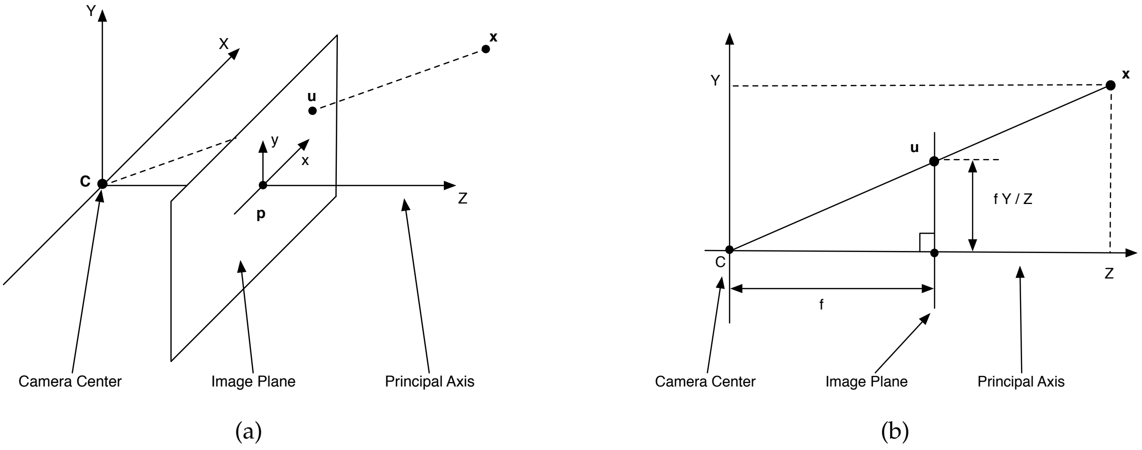


Figure 3.2: Pinhole camera geometry. C is the center of the camera and p the principal point. The camera is headed towards the positive z-axis. [23]; (a) shows the projection of x to the point u on the image plane; (b) shows the relationship between focal length f and the projected point u

Extrinsic Camera Matrix

In general the cameras are not heading in the same direction as defined in the basic pinhole camera model. So before the projection can be done, the general 3D points have to be transformed from the world coordinate frame to the camera coordinate frame.

A coordinate frame consists of translation $\mathbf{t}$ and rotation $\mathbf{R}$, which are in general combined in a single homogeneous matrix $\mathbf{T}$

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad . \tag{3.44}$$

This matrix is called the *extrinsic camera matrix* or *exterior camera parameters*. The mapping from the world to the camera coordinate frame can then be done by

$$\mathbf{x}_{\text{cam}} = \mathbf{T}\mathbf{x}_{\text{world}} \quad , \tag{3.45}$$

where $\mathbf{x}_{\text{cam}}$ and $\mathbf{x}_{\text{world}}$ are homogeneous vectors.

The extrinsic camera parameter matrix is completely independent from the camera specifications, as it only describes the position and orientation of the camera in space.

Intrinsic Camera Matrix

The *intrinsic camera parameters* are describing the internal parameters of a camera as for example the focal length f, the *principal point offset*, the *skew* and the *pixel aspect ratio* for electronic sensors in cameras. The intrinsic camera matrix is filled with these parameters. The full calibration matrix can be seen in photogrammetric and computer vision books like [36] and [23].

To show how the calibration matrix is built with the parameters, the principal point offset will be introduced as an example. The principal point offset describes the offset between the principal point and the center of the image plane [4, 23] (cf. figure 3.3) which is occurring due to imperfections during the production of the camera. Equation eq. 3.46 shows the intrinsic camera matrix $\mathbf{K}$ with focal length f and the principle point offset.

$$p_f(\vec{x}) = \frac{1}{z}\mathbf{Px} = \frac{1}{z}\mathbf{KTx} = \frac{1}{z}\begin{pmatrix} f & 0 & x_{\text{off}} & 0 \\ 0 & f & y_{\text{off}} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\mathbf{T}\vec{x} \tag{3.46}$$

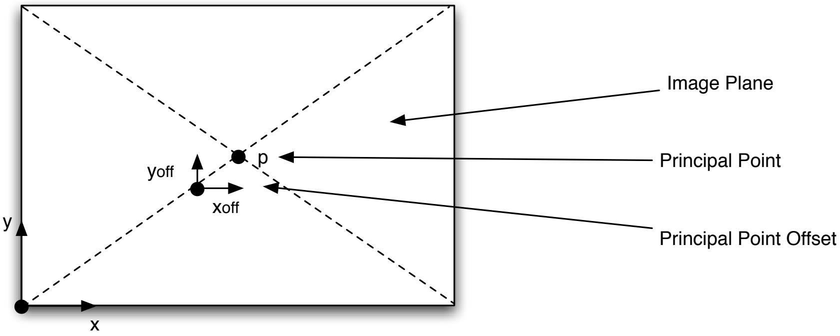


Figure 3.3: Principal point offset $(x_{\text{off}}, y_{\text{off}})$ on the image plane

Lens Distortion Model

Lens distortion models are supposed to eliminate imperfections of the lenses in a photogrammetric systems. The models which are presented here are based on the polynomial description of the distortion developed by Brown [10].

Radial-Symmetric Distortion If the image of a target is either too close to the principal point or too far away from the principal point it has been *radially distorted*. This error is also called *barrel* or *pin cushion*.

This aberration is generally the biggest error of the optical mapping. The *balanced radial distortion* is a correction of the distortion. It is commonly written as a power series (Seidel series) [10]:

$$\Delta r_{\text{rad}} = A_1 r^3 + A_2 r^5 + A_3 r^7 + \dots \tag{3.47}$$

Where A_i are the correction parameters. In general only the first two or three terms are needed, only for complex lenses like fish-eyes some more terms are needed [4].

An alternative the Seidel series (6.5) can be written polynomial:

$$\Delta r_{\text{rad}} = A_1 r(r^2 + r_0^2) + A_2 r(r^4 + r_0^4) + A_3 r(r^6 + r_0^6) \tag{3.48}$$

The value of r_0 depends on the parameters A_i. Normally r_0 is about $\frac{2}{3}$ of the maximal radius of the image plane [37].

Radial-Asymetric and Tangential Distortion The radial-asymetric and tangential distortions occur if the center of the lenses do not coincide with the principal axes. This distortion can be described by

$$\begin{aligned} \Delta x_{\text{tan}} &= B_1(r^2 + 2x^2) + 2B_2 xy \\ \Delta y_{\text{tan}} &= B_2(r^2 + 2y^2) + 2B_1 xy \end{aligned} \quad . \tag{3.49}$$

The radial-asymetric and tangential distortion of the most high quality lenses is small compared to the radial-symetric distortion and is needed only for high quality demands. According to Luhmann [37] this corrections must be estimated for off-the-shelf lenses like in video cameras and comparable.

Affinity und Skew Affinity and *skew* describing the deviation of the image coordinate system from orthogonality and the uniformity of the coordinate axes. This distortions can be caused by non-uniform placing of the sensor-elements. To correct this aberrations the following equations can be used [37]:

$$\begin{aligned} \Delta x_{\text{aff}} &= C_1 x + C_2 y \\ \Delta y_{\text{aff}} &= 0 \end{aligned} \tag{3.50}$$

Entire Distortion The combined common distortions in photogrammetric systems can be written as:

$$\begin{aligned} \Delta x &= \Delta x_{\text{rad}} + \Delta x_{\text{tan}} + \Delta x_{\text{aff}} \\ \Delta y &= \Delta y_{\text{rad}} + \Delta y_{\text{tan}} + \Delta y_{\text{aff}} \end{aligned} \tag{3.51}$$

Experiments with lens distortion parameters provided by a manufacturer showed that the lens distortion can be neglected when using high quality tracking hardware. Further research has to be done on this topic, especially as some tracking systems are using low quality off-the-shelf cameras.

4 Errors And Accuracy

In this chapter the errors occurring while tracking rigid objects will be classified and described. Also an error model will be introduced which allows to describe the non systematic errors.

4.1 Kinds Of Errors

The sources of error in tracking can generally be divided into two primary classes: *static* and *dynamic* errors. The first class covers all errors related to static measurement, which includes errors during calibrating the system and errors occurring while using the tracking system in a static way. So all parts involved in the tracking process are not moving. In contrast to static erros, dynamic errors are caused by end-to-end system delays when parts of the tracking system are moving eg. the cameras or the tracked object [7, 56]. Dynamic errors are not covered by this thesis although these errors can be more significant than the static errors for some setups. However, there are many applications where the dynamic errors can be neglected. In video see-through systems, the lag can be compensated by delaying the video stream for a certain amount of time [45].

The class of static errors can again be split into two different groups of errors: *systematic* and *noise*. The two classes will be introduced in the next sections.

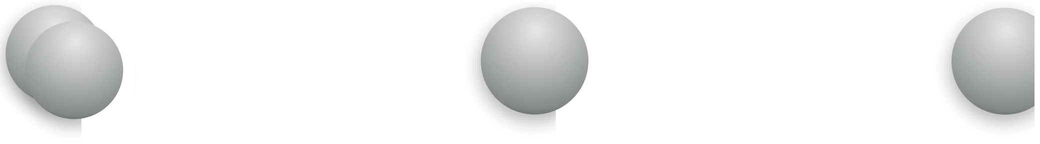
4.1.1 Systematic Errors

Systematic errors, also called *biased* errors, are repeatable which means when doing the same measurement several times the error will remain the same. In the next the most common error sources are described in detail.

Occlusions

When projecting feature points onto the image plane it can happen that one or more features are hidden by other objects, see figure 4.1 for some examples. In general this is no problem for detecting the whole target if there are still enough other features visible or other cameras can see the hidden features. If there are less than three features visible by at least two cameras the pose cannot be estimated anymore, which can easily be detected by the software and so the user of the tracking system can solve the problem by installing more cameras [12] or by building better targets [16].

If using a tracking system with retro-reflective marker balls as feature points it can happen that these markers are hidden partly by another marker ball or by another object. If the



(a) Marker partly hidden by other marker

(b) Marker fully hidden behind another marker

(c) One marker partly hidden by other object

Figure 4.1: Different Types of Occlusions

marker ball is still detected the estimation of the position will be erroneous, caused by the sub-pixel algorithms (cf. section 4.1.1). An example of this effect can be seen in the left illustration of figure 4.1(a). The position of the marker in front should be estimated, but the partly hidden marker will not be identified as a separate marker and so the position of the marker in front will be estimated slightly left of the real position.

Hidden or partly hidden markers change the geometry of the whole target, which leads to wrong estimated positions. So only visible markers and undisturbed markers should be used to determine the position of the target. Presumably the manufacturers of commercial tracking systems are using some quality metrics of the tracked markers to increase the tracking accuracy.

As the probability of occlusions is strongly depending on the placement of the cameras, a setup has to be found which increases the visibility of the markers and therefore minimizes the appearance of occluded markers and increases the accuracy. Chen et al. presented in [13] a quality metric for multi-camera configurations that includes a probabilistic occlusion model to address this kind of problem.

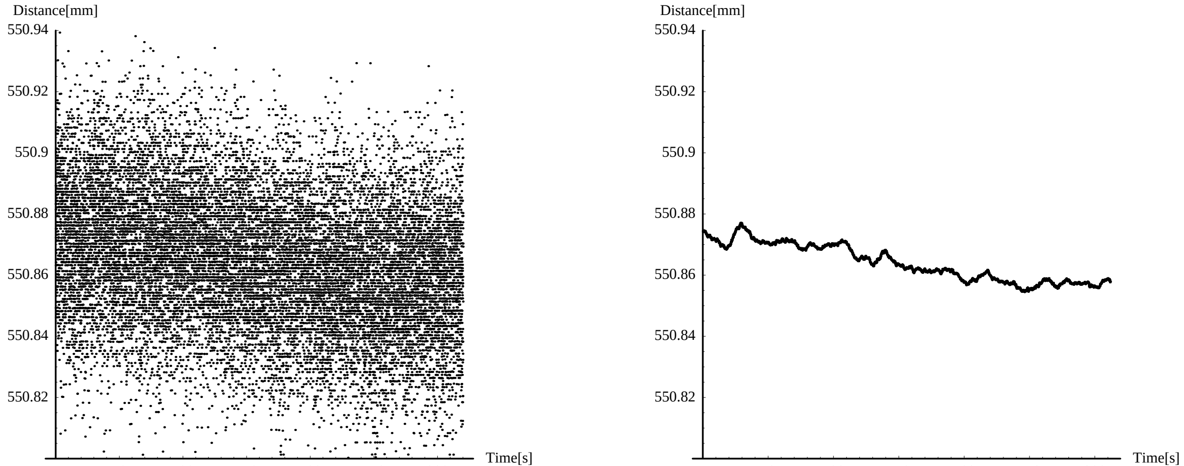
In this thesis errors caused by occlusion will not be analyzed.

Thermal Drift

The thermal drift arises if the camera is getting warmer than at the calibration time, due to some cameras that are using infrared flashes to illuminate the tracking scene. The electronic device needed for flashing is producing heat and also the power supply unit is getting warmer over the time.

The warming-up of the camera case changes the intrinsic parameters of the camera and is therefore affecting the accuracy. If the tracking system does not take care of this effect, the user should calibrate the cameras when the cameras are warmed up, otherwise the accuracy of the tracking results will get worse over the time. In [49] Trübswetter showed that in the first half an hour the drift is high. So the cameras should warm-up for at least half an hour. Trübswetter measured that drift for tracking a single feature with a stereo system is up to 0.7mm at a distance of 3m compared to the tracking accuracy of less than 0.5mm.

Even after half an hour warm-up phase a drift can be measured. Figure 4.2 shows the distance of two feature points over a period of 320 seconds measured with three ARTtrack1 cameras.



(a)

(b)

Figure 4.2: The thermal drift measured over 320 seconds after half an hour warm- up (ARTtrack1); (a) shows the measured distance of two rigidly connected features; (b) shows the mean of the measured distances

In the mathematical model developed later in this chapter the thermal drift will not be considered. First because the error can be avoided by warming-up the system, second because the system could be constantly recalibrated on-line while using the tracking system. Some researcher have done work on this real-time recalibration topic. Wang presented in [54] and [55] a technique to recalibrating the extrinsic parameters of a n-ocular tracking system, but as the thermal problem matters mainly in the intrinsic parameters a real-time recalibration for the intrinsic parameters should be found.

Wrong Poses of the Cameras

When using more than one camera a single feature point can be tracked by triangulating, but for the triangulation the position and orientation of all involved cameras have to be known. This estimation of the poses of cameras, also termed *room calibration* or *extrinsic parameter calibration*, has to be done every time the setup is changing.

For the room calibration 2D point correspondences on the image plane are needed which are erroneous due to noise, so the poses of the cameras will be estimated with errors which result in additional errors in the triangulation. Furthermore the cameras will never stay still, because of vibrations caused by users walking in the laboratory, by opening or closing doors or by other mechanical influences.

So three main error sources for bad extrinsic parameter calibration can be distinguished. The first error is caused if the initial calibration of the system has not be done carefully, the next error is produced by moved or rotated cameras after the initial calibration and the third source of error comes from short or periodical changes in the position or orientation of the cameras, caused by external physical influences. The three sources could also be eliminated by using a real-time recalibration algorithms, like presented by [54, 55]. Is has to be analyzed if these algorithms are also capable to eliminate the negative effects of vibrations.

Scaling of Room

When measuring the distance of two rigidly connected features with a multi-camera setup, the distance of these two features will be estimated with errors. The error in distance estimation is caused mainly by systematic errors. When moving the two features through the working volume the error in distance will change, as the static error in estimating the position of a single feature is varying within the working volume.

In section 5.1.3 an experiment is introduced where a two-feature target is rotated around its centroid and the distance between the two features is measured (cf. fig. 5.7(a)). Figure 5.7(b) shows the measured length estimation; the horizontal axis shows the angle of the two-feature target and the vertical axis the respective measured distance.

The plot shows an oscillation with period π in the measured distances. This systematic error presumably comes from a wrong scaling of the three room axes during the calibration of the used A.R.T.(cf. sec.5.1.1) tracking system.

Subpixel Accuracy

When tracking feature points the features have to be segmented and identified in the taken picture to estimate the 2D position of the feature on the image plane. In figure 4.3(b) a picture of a 12mm retro-reflective marker ball taken by tracking camera can be seen. The problem in estimating the position is that features projected on the chip are covering only a few pixels on the image plane. Table 4.1.1 shows the geometrical relationship between marker ball diameter, distance to the camera and the resulting pixel-sized diameter on the image plane when using a ARTrack1 camera. To increase the number of covered pixels the focus of most cameras are slightly defocused. If the tracking system uses passive retro-reflective or active LED markers the blooming on the image plane is also increasing the amount of covered pixels.

	Distance from Camera to Marker									
Size	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
12mm	5.67	2.84	1.89	1.42	1.14	0.95	0.81	0.71	0.63	0.57
14mm	6.62	3.31	2.20	1.66	1.32	1.10	0.95	0.83	0.74	0.66
20mm	9.46	4.73	3.15	2.36	1.89	1.58	1.35	1.18	1.05	0.95
30mm	14.19	7.09	4.73	3.55	2.83	2.36	2.02	1.77	1.58	1.42

Table 4.1: 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 $\times$ 3.67mm)

The estimation of 2D position is not restricted to the discrete pixel coordinates on the image plane (cf. 4.3(a)). In experiments presented later in this thesis, a subpixel accuracy of about $\frac{1}{115}$ was measured.

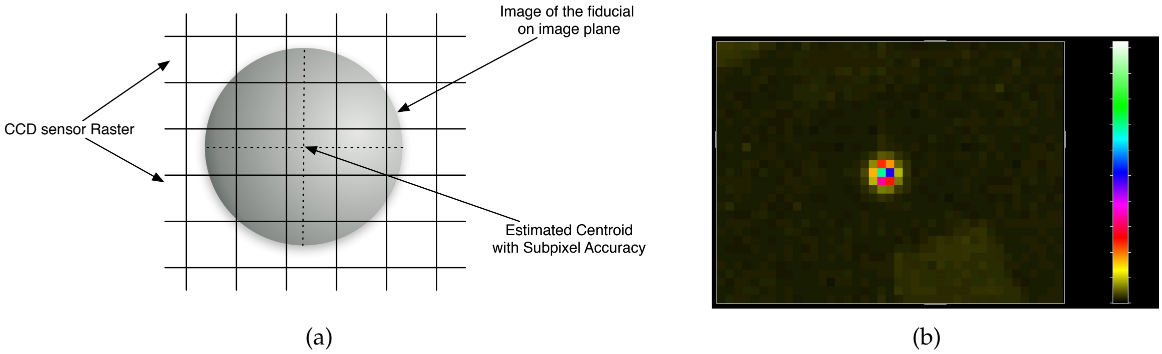
There are several subpixel algorithms. Shortis et al. [48] gives a survey over common techniques and their expected accuracy. As an example a simple algorithm is shown to gain an idea how location estimation with subpixel accuracy works.

Wong et al. [59] introduced the "Weighted-Centroid Algorithm". It is assumed that the features are already segmented. Let t_{up} the upper and t_{low} the lower threshold bound. The segmented image dimension are n and m and the weight w_{ij} is representing the grey value at the image position (i, j). Then the position of the feature can be calculated by the next formulae:

$$g_{ij} = \begin{cases} 1 & \text{if} \quad t_{\text{low}} \leq w_{ij} \leq t_{\text{up}} \\ 0 & \text{if} \quad \text{other cases} \end{cases} \tag{4.1}$$

$$\begin{aligned} M &= \sum_{i=1}^{n} \sum_{j=1}^{m} g_{ij} w_{ij} \\ x &= \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} j g_{ij} w_{ij} \\ y &= \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} i g_{ij} w_{ij} \end{aligned} \tag{4.2}$$

With this technique only these pixel are considered which are within the threshold bounds and the brighter pixels, which are usually at the center of the feature, are counting more then darker pixels.



(a)

(b)

Figure 4.3: Subpixel Accuracy of optical markers; (a) shows the estimation of a feature with subpixel accuracy; (b) shows the picture of a 12mm retro-reflective marker taken with a ARTrack1 Camera

The algorithm used for locating the markers 2D-position is crucial for error produced by occluded markers. The problem of occluded markers is discussed in section 4.1.1.

Which algorithms are implemented in commercial tracking systems is the decision of the manufacturer and can not be influenced by the user. Some tracking systems offer different options for location the markers, eg. in A.R.T.'s DTrack tracking software (cf. section 5.1.1) there is an *accurate* and a *fast* tracking mode. When tracking in accurate mode less markers can be tracked than in fast mode. Presumably two different segmentation and subpixel algorithms are used for the two different modes.

The subpixel accuracy will be considered in the error model derived later in this chapter.

Quantization

After digitalizing the image on the camera sensor, the continuous gray values have to be mapped to discrete gray values. Assuming there are N different gray values possible the mapping is done by

$$Q : [0, \infty[\mapsto \{g_0, g_1, \ldots, g_{N-1}\} \tag{4.3}$$

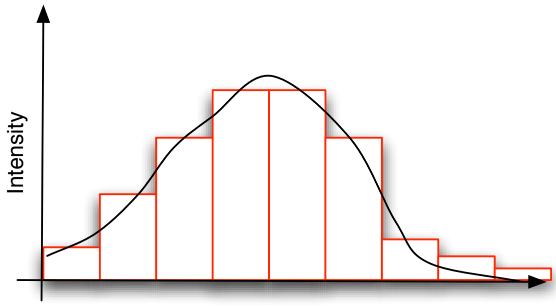


Figure 4.4: Quantization of Grey Values

This leads to an error because the real value q will be replaced by an discrete value g_q. If the steps of quantization are all equal and all gray values occurring with the same the probability then the variance due to quantization is [29]

$$\sigma_q^2 = \frac{1}{\Delta g} \int_{g_q - \Delta q/2}^{g_q + \Delta q/2} (g - g_q)^2 \mathrm{d}g = \frac{1}{12} (\Delta g)^2 \tag{4.4}$$

So the standard deviation of quantization σ_q for a single measurement is about 0.3 of the quantization steps Δg

By taking more than one measurement the mean of σ_q can be calculated. According to the statistical error propagation the mean σ_{mean} can be calculated with

$$\sigma_{\text{mean}} \approx \frac{1}{\sqrt{N}} \sigma \tag{4.5}$$

Where N is the amount of measurements and σ is the standard deviation of a single measurement. In a perfect system multiple measurements would result in the same value, so σ would be zero. Unfortunately there is noises so make the same measurements multiple time will not deliver equal results. If assuming the noise is the difference of two quantization steps Δg then the standard deviation of a single measurement would be three times bigger than the standard deviation resulting from the quantization; but when taking 100 measurements the resulting σ_{mean} is only a tenth part of σ and only $1/3$ of the standard deviation resulting from the quantization.

However, the accuracy of quantization is also limited by some other systematic errors. The most significant is caused by unequal quantization steps. Because analog digital converters which do the quantization in cameras, produce in general such unequal quantization steps, a carefully analysis has to be done what the limiting factor is for the accuracy of quantization.

In the tracking domain, this error plays only a small role, because it is influencing only the gray values but not the shape of the projected feature points on the image plane; and the shape is the most important factor for calculation the 2D position on the image plane.

Uncertainty in Target Geometry

Tracking a pose of a marker target needs corresponding points of the target model for solving the 3D-to-3D pose estimation problem in the n-ocular case and the 2D-to-3D pose estimation problem in the monocular case. For a correct pose estimation the positions of the model points should be exact to avoid additional errors in pose estimation.

Getting an exact model of the target is not possible. When designing the target virtually and then building the target physically according to the model even with high accuracy, errors are inevitable. Another way to get the geometric model is to build the target and then measure the 3D points with a tracking system or other measurement tools. It is easy to consider that the model obtained with this method is also tainted with errors, concerning the errors described in this section and that every measurement is tainted with noise. By taking multiple measurements this error can be reduced.

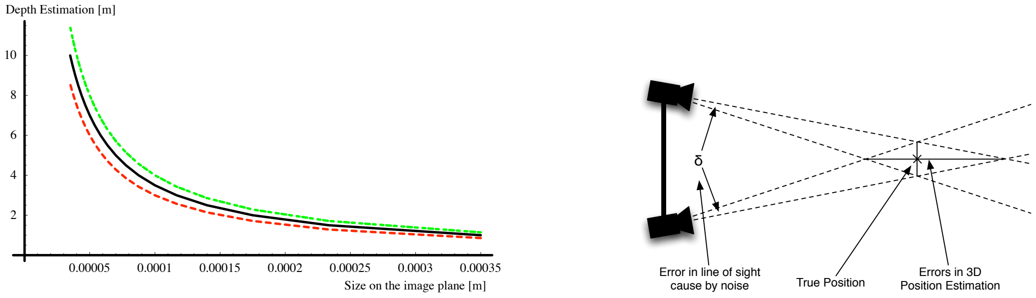
It is also possible that the target geometry is changing over longer periods of time, for example by rough treatment of the users.

So the model of the target will always be tainted with errors. The accuracy prediction technique presented by Davis et al. [15] takes care of the uncertainty in the model, whereas the accuracy prediction methods shown in this thesis are assuming that the model is known exactly.

Intrinsic Camera Parameters

The intrinsic camera parameters referred in section 3.5.1 are estimated with photogrammetric calibrations methods. In general the manufacturers of high precision cameras do a calibration in their factory, but the parameters are varying during usage [4]. And especially when using off-the-shelf cameras the calibration becomes a major issue, as the lenses of such cameras are usually of poor quality.

If the calibration of the intrinsic camera parameters is not done exactly, the projection of the feature points will be tainted with errors. When for example the focal length is calibrated falsely, positions on the image plane will be treated with a false scaling and so the distance of a rigid object tracked with a monocular system will be estimated falsely. This can be seen in figure 4.5(a), which shows the distance estimation with a false focal length.



(a)

(b)

Figure 4.5: In 4.5(a) the wrong depth estimation caused by false focal length can be seen. The abscissae address the object's size on the image plane and the ordinate corresponds to the estimated distance. The middle black curve shows the depth estimation with the correct focal length of $f = 3.5$mm, the upper green curve shows the estimation with $f = 4.0$mm and the lower red curve corresponds to a focal length of $f = 3.0$mm; In 4.5(b) shows the negative effect of the noise on the image plane which causes errors in triangulation [46]

As the calibration of the intrinsic parameters has not to be done often, it is recommended to do the calibration with high precision.

4.1.2 Random Noise

In contrast to systematic errors the errors caused by random noise, also called jitter, are not repeatable, which means that when taking multiple measurements the results will differ in each measurement. Noise in sensor input is inevitable with any measurement system [9]. In sense of tracking this will cause randomly changing gray values on the image plane, which then causes errors in estimation of the 2D position of feature which then again causes errors when solving the 3D-to-3D or the 2D-to-3D pose estimation problem. Figure 4.5(b) shows this effect of wrong 2D positions when triangulating a feature point, it can also be seen that the triangulation error gets greater with growing distance between the feature and the image plane. The intersecting noisy lines-of-sight are forming a deltoid and restricting the estimated 3D position within the deltoid boundaries. With growing distance also the deltoid grows, which results in a greater error of the triangulated position [46]. This model can also explain the negative effect of small baselines in depth estimation. If the baseline gets smaller the deltoid grows in length which results in a greater uncertainty.

Noise can not be corrected for example with calibration or other techniques. It would be possible to filter out noise but this would lead to delays in the tracking process and therefore to greater dynamic errors when objects are moving.

4.2 Representation of Accuracy

As stated earlier, pose accuracy describes how accurate a object can be tracked with position and orientation. To quantify the accuracy a mathematical description has to be found, which has to consider the biased and jitter errors. The following approach is based on the error description of Davis et al. in [15]. Let T_{err} be the translational part of the pose $p \in \mathbb{R}^6$ including all kind of errors and R_{err} the rotational part represented by a matrix. T and R are describing the "true" position and rotation and ΔT and ΔR are describing the errors. So it can be written

$$\begin{aligned} T_{\text{err}} &= \Delta T + T \\ R_{\text{err}} &= \Delta R R \quad , \end{aligned} \tag{4.6}$$

The errors of the pose ΔT and ΔR, which can be divided into the two classes, biased and jitter errors, so the complete pose including both errors can be written as

$$\begin{aligned} T_{\text{err}} &= \Delta T_{\text{jitter}} + \overbrace{\Delta T_{\text{biased}} + T}^{T_{bias}} \\ R_{\text{err}} &= \Delta R_{\text{jitter}} \underbrace{\Delta R_{\text{biased}} R}_{R_{bias}} \quad . \end{aligned} \tag{4.7}$$

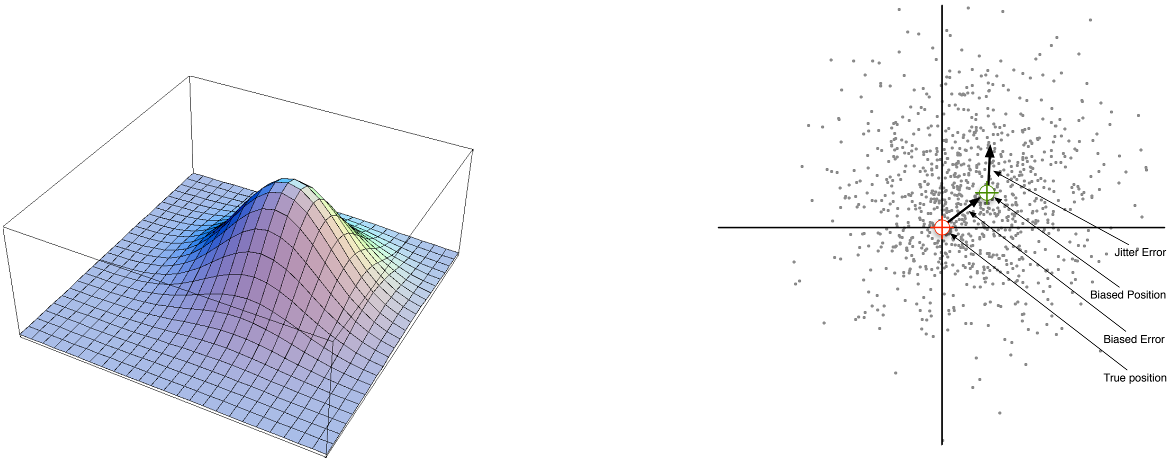
Now the errors are formally separately described, but this allows only to describe an error of a single measurement.

4.2.1 Multi Dimensional Normal Distribution

When repeating a measurement, the biased errors will remain the same, just as defined, but the jitter errors will differ in each measurement. By taking many measurements the results would be spreading around the true pose including the biased error. The true pose tainted with the biased error is also also called the *biased pose*.

By interpreting the error in pose measurement as a six-dimensional normally distributed random variable, the mean $\mu_p \in \mathbb{R}^6$ would represent the biased error and a covariance matrix $\Sigma_p \in \mathbb{R}^{6\times 6}$ would describe the jitter errors. Because it not easy to illustrate this in all six dimensions, the problem has been restricted to two dimensions for illustration purposes, so let X a two dimensional random variable normally distributed with $X \sim \mathrm{N}(\mu, \Sigma)$. In Figure 4.6(a) a probability density of X can be seen. The peak in the plot represents the biased pose, which is equivalently to the mean. In figure 4.6(b) a set of 1000 sample events of X is plotted. The origin, representing the true pose, is plotted as a red cross, the green cross is representing the biased pose, the difference vector represents the biased error and the difference vectors from the plotted points to the biased pose are standing for the jitter errors.

Back to the real six-dimensional case, it can be stated that error variable P representing the error in pose measurement is normally distributed as follows



(a) (b)

Figure 4.6: Two dimensional normal distribution with mean μ and covariance matrix Σ; In figure 4.6(a) the plot of density function, which describes the probability for appearance of a specific point can be seen. The peak represents the biased position; Figure 4.6(b) shows the plot of normal distributed points, which are arranged around the biased position.

$$p \sim \mathrm{N}(\mu_p, \Sigma_p) \quad . \tag{4.8}$$

This model is not restricted for only pose accuracy, but it can also be used for position and rotation separately or for any kind of measurements.

The advantage of this accuracy representation is mainly the possibility of describing unisotropic jointly distributed errors, which is generally the case, and the possibility to use the covariance propagation rules. So this kind of accuracy representation is used in this thesis.

4.2.2 Root Mean Square

This section is about *root-mean-square*, or shortly *rms*. It will be shown how the rms is defined and how it can be used to represent the accuracy in the sense of tracking. The relationship between the rms and the representation of errors with a multidimensional normal distribution will be derived.

Definition

The rms is a statistical measure of the magnitude of a varying quantity. In (4.9) the definition of the RMS can be seen.

$$x_{\text{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_N^2}{N}} \quad . \tag{4.9}$$

If the standard deviation σ_x and the mean $\bar{x}$ of the measured values x_i are known, the rms can be calculated with

$$x_{\text{rms}}^2 = \bar{x}^2 + \sigma_x^2 \quad . \tag{4.10}$$

The following proof shows the correctness of (4.10):

$$\begin{aligned} \bar{x}^2 + \sigma_x^2 &= \bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \\ &= \bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N}(x_i^2 + \bar{x}^2 - 2x_i\bar{x}) \\ &= 2\bar{x}^2 + \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \frac{2}{N}\bar{x}\sum_{i=1}^{N} x_i \\ &= 2\bar{x}^2 + x_{\text{rms}}^2 - 2\bar{x}^2 \\ &= x_{\text{rms}}^2 \end{aligned} \tag{4.11}$$

Representing the error with the root mean square is a simple and intuitive and therefore a common way.

Relation between Rms and Normal Distribution

The rms of a multi dimensional random variable with known distribution $X \sim \mathrm{N}(\mu, \Sigma)$ can be calculated from the distribution parameters.

As known, the variances $\mathrm{Var}(X_i)$ along the principal axes of a covariance matrix are identical with the eigenvalues of the covariance matrix Σ. The formula for calculating the rms with mean μ and standard deviation σ can also be written with variance

$$rms_i^2 = \mu_i^2 + \sigma_i^2 = \mu_i^2 + \mathrm{Var}(X_i) \quad . \tag{4.12}$$

As mentioned before, the sum of eigenvalues of a matrix Σ is the same as the trace $\mathrm{tr}(\Sigma)$, so it can be written

$$\begin{aligned} rms^2 &= \sum_{i=1}^{n} rms_i^2 = \sum_{i=1}^{n}(\sigma_i^2 + \mu_i^2) = \sum_{i=1}^{n}\sigma_i^2 + \|\mu\|^2 \\ &= \sum_{i=1}^{n} \operatorname{Var}(X_i) + \|\mu\|^2 \\ &= \operatorname{tr}(\Sigma) + \|\mu^2\| \quad [30]. \end{aligned} \tag{4.13}$$

Sometimes it is also possible interpreting a given rms as a multidimensional random variable, but in general the rms error includes both mean and standard deviation in a single scalar value and thus it is not possible to separate these two statistical moments. By assuming that the rms error is not biased, $\mu = 0$, the sum of the standard deviations along the principal axes is known. By knowing only the sum of σ_i it is again impossible to distinguish the single deviations, and so it is assumed that all standard deviations σ_i are of the same size

$$\begin{aligned} rms_{\mu=0}^2 &= \sum_{i=1}^{n}\sigma_i^2 = \sum_{i=1}^{n}\sigma^2 = n\sigma^2 \qquad . \\ \Leftrightarrow \quad \sigma^2 &= \frac{1}{n} rms_{\mu=0}^2 \quad . \end{aligned} \tag{4.14}$$

This leads to a isotropic independent normal distributed random variable $X \sim \mathrm{N}(\mu, \Sigma)$ with $\Sigma = \operatorname{diag}(\underbrace{\sigma^2, \sigma^2, \ldots, \sigma^2}_{\text{n times}})$:

$$X \sim \mathrm{N}(0, \frac{1}{n}\begin{pmatrix} rms_{\mu=0}^2 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & rms_{\mu=0}^2 \end{pmatrix}) \tag{4.15}$$

Calculation of rms in n dimensions

There is no clear definition how to determine the rms of a given data set. A simple approach would be to calculate the arithmetic mean $\mu \in \mathbb{R}^n$ and the standard deviations $\sigma_1, \ldots, \sigma_n$ along the axes of the coordinate frame, which corresponds to an independent distribution $X \sim \mathrm{N_n}(\mu, \Sigma)$ with $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$. The total rms can then be calculated with

$$\begin{aligned} rms^2 &= \sum_{i=1}^{n} rms_i^2 = \sum_{i=1}^{n}(\sigma_i^2 + \mu_i^2) = \sum_{i=1}^{n}\sigma_i^2 + \|\mu\|^2 \\ \Leftrightarrow \quad rms &= \sqrt{\sum_{i=1}^{n}\sigma_i^2 + \|\mu\|^2} \quad . \end{aligned} \tag{4.16}$$

The two methods of rms determination lead to different rms values. The simple approach estimates the rms too big because it does not consider the joint distribution of the data. An

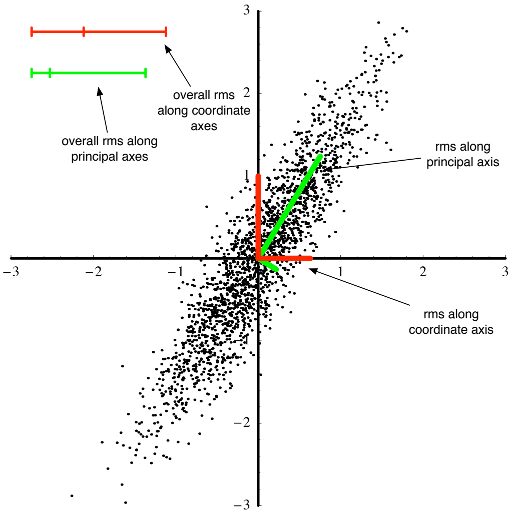


Figure 4.7: Difference of rms calculated along the coordinate axes and along the principal axes

example for the difference between the simple method and the one considering the principal components can be seen in figure 4.7, where the red lines showing the simple approach and the green line the advanced. So the second approach, which uses covariance matrix, is more precise but the calculation of the covariance matrix needs more complexity than just calculating the standard deviations along the coordinate axes.

Using the rms for representing errors is much more intuitive for the users, because it is measured in the same unit as the measured data and there is only one value to take care of. Reducing the error to one single number implies that errors are represented as an independent isotropic distribution and that the mean and the standard deviation can not be separated. Representing errors with a normal distribution in n dimension is describing the errors more precisely, but a covariance matrix is harder to be interpreted by the user.

Another way to calculate the rms error considers the principal components of the dataset in n dimensions. This can be achieved by computing the covariance matrix and the mean of the data set, then a normal distribution in n dimensions is given, which allows to determine the rms with eq. (4.13).

5 Accuracy of Optical Tracking Systems

In this chapter the theoretical accuracy of optical tracking systems will be discovered.

The chapter is divided into two sections which address the two main classes of tracking systems. The first part is about a mathematical framework for predicting the accuracy of n-ocular tracking systems which use more than one camera to track spatial positions. This kind of system is most common in eg. augmented reality and photogrammetric measurement systems as it is providing higher accuracy and a greater working volume than a monocular system. The accuracy framework for n-ocular tracking systems considers errors caused by noise while other types of errors such as biased or dynamic errors are not regarded. Some systematic errors like subpixel accuracy (cf. 4.1.1) are highly influenced by random noise, so these kind of errors can also be considered.

In the second part a framework for predicting the pose estimation accuracy of monocular systems is presented which can predict the pose accuracy of a target consisting of multiple features. As this model uses the same mathematical methods as the framework for n-ocular systems, also only noise errors are considered.

5.1 N-Ocular System Detecting Single Feature Point

N-ocular tracking systems use multiple cameras for tracking. They locate the projections of the features on the image plane and then determine the 3D position by triangulation.

As mentioned in section 2.2 optical trackers use usually artificial spherical markers or LEDs to ease the process of finding and estimating the position of the features in the taken picture. But other kinds of features and even natural features can also provide feature points; for instance two edges are intersecting in one single point; then this feature point could be triangulated to get a spatial position of the point.

When the picture is taken by the camera, the features have to be segmented, identified and their positions on the image plane have to be determined. As the measurements will always be tainted with noise the rays used for triangulation will in general not meet each other (cf. fig. 5.1), this happens of course also if using more then two cameras for the triangulation where the triangulation problem is over determined. The triangulation problem of 3D spatial positions with noisy 2D positions can be formulated as a least-square minimization problem. Solutions for this problem can be found in [22, 23]. It is easy to consider that the triangulation is getting the more precisely the more cameras are used. The distances of two cameras is termed *baseline*. The baseline is also strongly influencing the accuracy triangulating a feature point; with increasing baseline the accuracy is also increasing [25].

By rigidly combining multiple features to a marker target the full *six-degree-of-freedom*

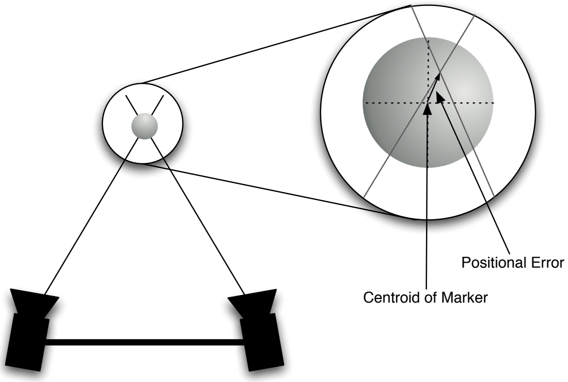


Figure 5.1: Triangulation and Resulting Errors

(DoF) pose of the target can be determined, when the geometric relation of the single features is known. This 3D-to-3D pose estimation – also called *absolute orientation problem* – can be done if there are at least three point correspondences. Let $\{p_i\} \in \mathbb{R}^3$ be the known positions of the marker target and $\{q_i\} \in \mathbb{R}^3$ the measure positions. The goal is to find the pose of the target with respect to the sensor or another fixed coordinate system. Because the measurement and the topology of the marker target are always tainted with noise there will be in general no exact mapping found that brings the two sets q_i and p_i in a exact relationship, so some kind of least-square solution has to be found. There are many algorithms to solve that problem. Horn presented in [27] a solution which delivers the positional offset, a scaling factor and an orientation in quaternion form. Other solutions are developed for instance by Haralick et al. in [21] and by Zhengyan et al. in [64]. Figure 5.2 is showing the function of a n-ocular tracking in principle.

Beside the higher accuracy [24] of n-ocular tracking systems there are more advantages, for example the possibility of occlusions is decreasing with increasing amount of cameras [13]. Occlusions are happening if the user or other objects are obstructing the line-of-sight of one ore more cameras. Another advantage is that the working volume can be increased by combining more cameras [12].

The drawbacks of an n-ocular system for instance are that the poses of the cameras have to be carefully determined via room calibration, because the poses are needed for a precise triangulation. And as most commercially available n-ocular tracking systems using infrared light conditions to locate the features on the image plane, the technical effort is greater in contrast to a monocular system. Each camera needs to illuminate the working volume with its own flash. As only one flash should illuminate the scene at the same time the cameras and flashes have to be synchronized; if more than one flash is active, the accuracy detecting the markers on the image plane is decreasing as subpixel algorithms assuming the brightest spot in the middle of the retro-reflective feature.

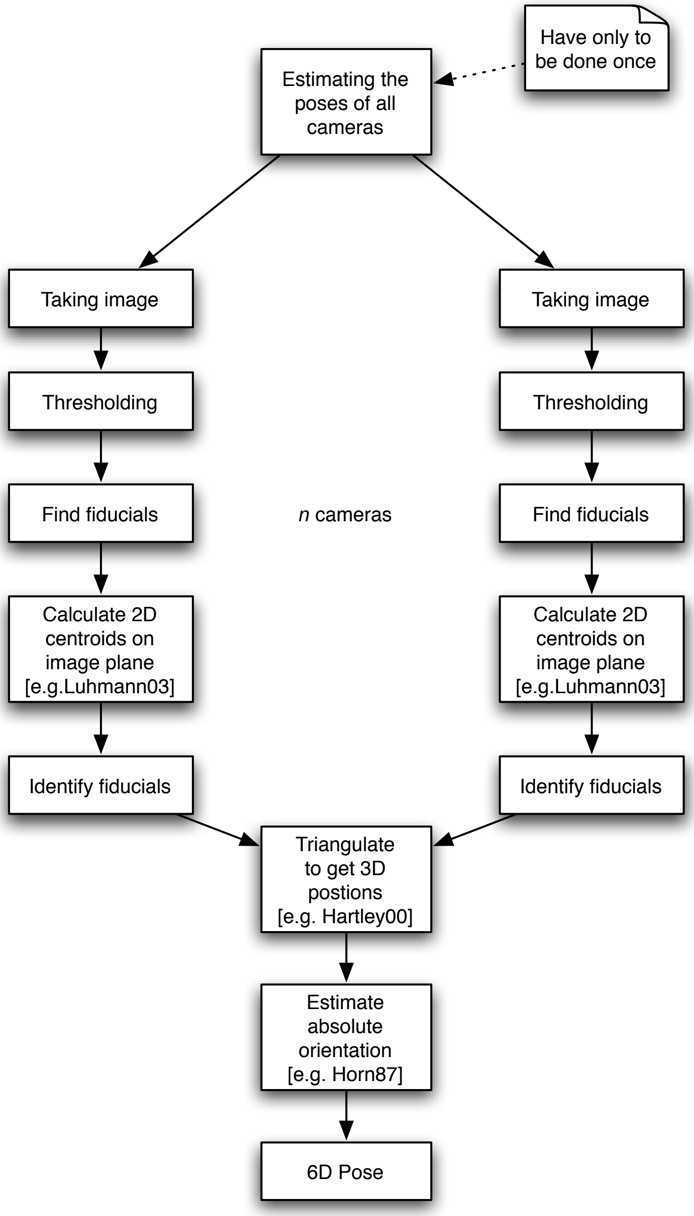


Figure 5.2: N-ocular optical tracking

Sensor elements	640×480
Chip size	4.87mm $\times 3.67$ mm
Pixel size	7.401μm $\times$ 7.399μm
Focal length	3.5 mm

Table 5.1: Specification of ARTrack1 Camera from A.R.T. Germany

5.1.1 Survey of N-Ocular Tracking Systems

There are several commercial n-ocular tracking systems available. Most of them using optical artificial features which are also called *optical landmarks*, *optical beacons* or markers [8]. Passive markers are illuminated with an infrared light source in contrast to active markers which are emitting infrared light by itself. The usage of the infrared spectrum makes the markers stand out against the background.

A comprehensive overview of available commercial tracking systems is presented in [44]. The most common products will be introduced now.

Advanced Realtime Tracking GmbH The A.R.T.[1] System consists of up to six precision cameras working in the near infrared light spectrum and the tracking software *DTack*. The system uses passive retro-reflective or active LED markers to provide 3DoF for single feature points and 6DoF for rigid bodies. The maximum frame rate is 60Hz and up to 20 rigid bodies can be tracked. Table 5.1 shows the specifications of an ARTtrack1 tracking camera manufactured by A.R.T..

Northern Digital Inc. Northern Digital[2] produces two popular tracking systems: *Optotrak* and *Polaris*.

The Optotrak system uses synchronized infrared LED markers which are tracked by three infrared cameras mounted on a rigid base. The tracking software delivers spatial position, angles, moments and rotation. The update rate for tracking spatial positions of three markers is up to 750Hz and for tracking the poses of three markers up to 145Hz.

The Polaris tracking system is available in a passive and a passive/active configuration. Both systems use a stereo rig with frame rate up to 60 Hz. The hybrid system can track up to 12 targets and the passive system up to 9 targets simultaneously.

NaturalPoint NaturalPoint[3] provides *Optitrack FLEX3* cameras. The cameras are working in the near infrared spectrum with a frame rate of 120Hz and a resolution of 355×288 pixels. For developers NaturalPoint provides a software development kit (SDK).

5.1.2 Derivation of Covariance Formulas

In the following the 3D covariance $\Sigma_{\vec{p}}$ of the detection of a single marker ball at position $\vec{p}$ in a given setup will be estimated. It is assumed for now that the intrinsic and extrinsic camera

---

[1] Advanced Realtime Tracking GmbH (A.R.T.): `http://www.ar-tracking.de/`

[2] Northern Digital: `http://www.ndigital.com`

[3] NaturalPoint: `http://www.naturalpoint.com`

parameters are known exactly and no other systematic errors are occurring, so only noisy errors are present. Further it is assumed that the covariance matrix $\Sigma_{\vec{u}} \in \mathbb{R}^{2\times 2}$ for estimating the 2D position $\vec{u}$ on the image plane is known. The covariance matrix $\Sigma_{\vec{u}}$ can be different for each camera.

The camera projection function assuming a pinhole camera model is in homogeneous coordinates

$$\rho \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{KT}\vec{x}$$

where ρ is the normalization factor, i.e. the inverse of the third row of the camera matrix equation and $\mathbf{K}$ are the intrinsic and $\mathbf{T}$ the extrinsic camera parameters.

When using an n-ocular stereo system detecting the same point, the *triangulation* is given by a set of nonlinear camera equations p:

$$\mathrm{p} : \mathbb{R}^3 \to \mathbb{R}^{2n}, \vec{x} \mapsto \mathrm{p}(\vec{x}) = \begin{pmatrix} \vec{u}_1 \\ \vdots \\ \vec{u}_n \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_1}\mathbf{K_1 T_1}\vec{x} \\ \vdots \\ \frac{1}{\rho_n}\mathbf{K_n T_n}\vec{x} \end{pmatrix} \tag{5.1}$$

as the projection function, with $\mathbf{K_i}$ and $\mathbf{T_i}$ being the respective parameters of the i-th camera. So the 3D position is projected onto the image plane of all n involved cameras.

The position estimation covariance from each camera is known and the covariance of the 3D position $\vec{x}$ is desired, so the backward propagation formula has to be applied. Therefore the Jacobian of p with respect to $\vec{x}$ is needed

$$J_p = \frac{\partial\,\mathrm{p}}{\partial x}\,|_{\vec{x}} \quad \in \mathbb{R}^{2n\times 3} \quad . \tag{5.2}$$

The Jacobian J_p then is applied to the backward propagation formula (3.41)

$$\Sigma_x = \left({J_p}^t \begin{pmatrix} \Sigma_{u_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Sigma_{u_n} \end{pmatrix}^{-1} J_p \right)^{\ddagger} \quad . \tag{5.3}$$

With this formulas the 3D position estimation covariance Σ_x can be calculated for an arbitrary position in the working volume.

The resulting equations are analytically computed using a computer algebra system and are then evaluated for each position in space.

Figure 5.3 shows a visualization of the error covariances in a two-camera setup. For this calculation an isotropic standard deviation of $\frac{1}{115}$ pixel on the image plane in each direction is assumed; the cameras have a 50cm baseline and a focal length of 3.5mm. In section 5.1.3 the estimation of this covariance will be shown.

For real world setups like the one presented here, the field of view of the cameras is considered and only that cameras are used, which can see the point for the error estimation [12].

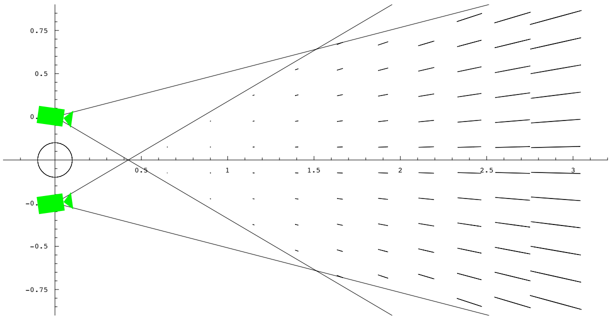


Figure 5.3: Error Covariances for tracking a single marker ball in a two-camera setup (Magnified by factor 100 for visualization)

5.1.3 Experimental Estimation of Errors

In this section a series of experiments will be presented, that were conducted to validate the model and to experimentally get an estimate of the covariance on the image plane of a specific optical tracking system.

The error on the image plane has mainly two sources, image noise and artefacts from the subpixel algorithms used for the detection of the marker balls in the camera image.

Image Noise

To estimate the error realistically from image noise, retro-reflective marker balls are placed as features in a regular grid on a table in the tracking volume. Both the cameras and the markers on the table are fixed throughout the experiment. The measured locations of the features in space are captured for several minutes at 60 Hz resulting in a total of 71553 sample points for each feature. A small baseline of about 20 cm is used to emphasize the effects in the images, see figure 5.4 (error for visualization exaggerated by a factor 50)

The covariance for each single feature point is calculated from this dataset as seen in figure 5.5. The covariance is displayed as an ellipse at confidence level 75% 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.

Now the actual error covariance on the image plane is estimated using the experimental measurements for a single feature point and apply the model to predict the parameters for the other fifteen points; the prediction is then compared with the measurements of these points.

To estimate the parameters, the model with symbolic parameters for the covariances on the image plane is evaluated and use Newton's method to fit the parameters to the measurements. Figure 5.6 shows the measured errors in red together with the computed errors in black first horizontally on the grid (x-z-plane) and then parallel to the image plane (y-z-plane, projected onto the grid). The parameter estimation was done for the lower most

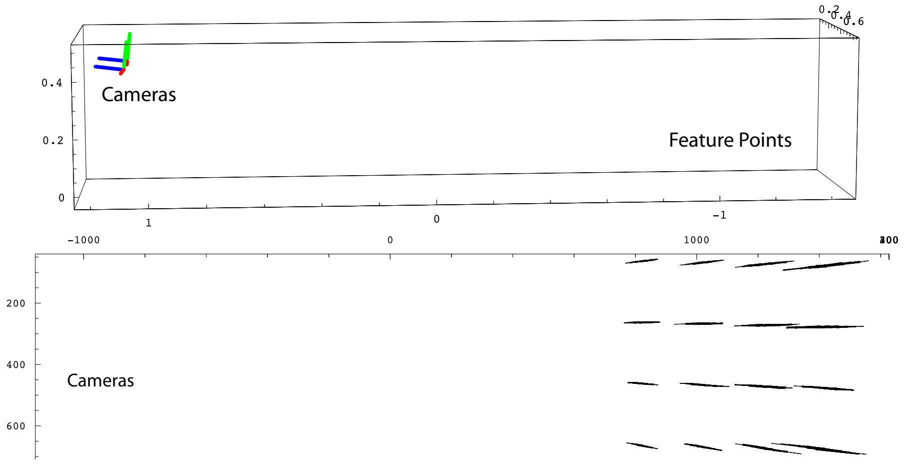


Figure 5.4: Error in position estimation for a regular grid (side & top view of the setup)

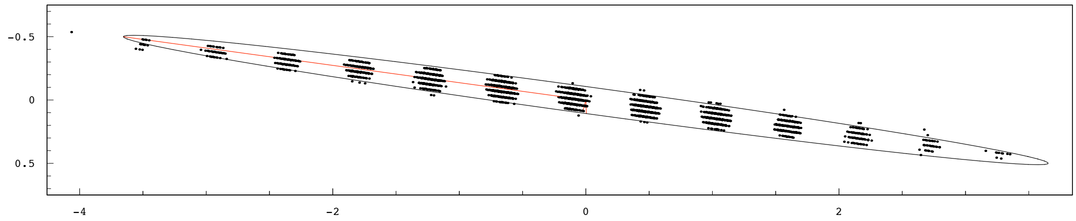


Figure 5.5: Error in position estimation together with estimated covariance for a single marker from figure 5.4

left point and then applied to all other points. The predicted values fit the measured values quite well. In the experiment, a standard deviation of $\frac{1}{115}$ pixel on the image plane for the detection of the center of the features was estimated.

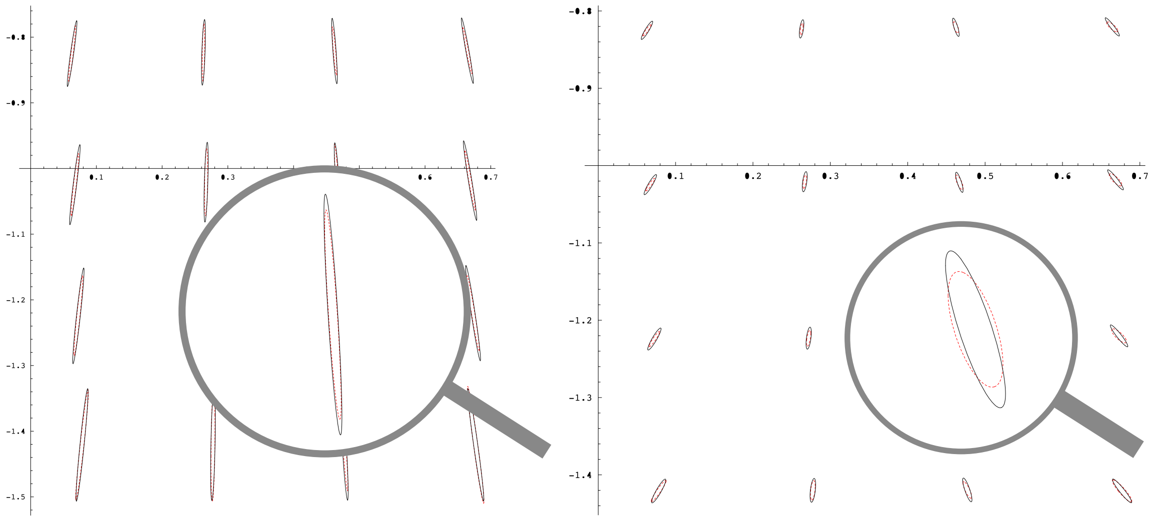
Image Noise and Subpixel Algorithm Noise

To estimate the accuracy of optical coordinate measurement systems, VDI/VDE 2634/1 [52] recommends to measure differences between single features in several directions. This measurement is extended with a rotating two-ball target (cf. figure 5.7(a)) for which the distance between the balls is measured.

Applying the theoretical error prediction model to this kind of test, it is necessary to build the Jacobian J_d from the distance function,

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

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



x-z-plane y-z-plane

Figure 5.6: Measured errors (red dotted line) vs. computed errors (black line)

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

The variance σ_d is then compared with the measurements, as shown in figure 5.7(b). The horizontal axis shows the angle of the two balls and the vertical axis the respective measured distance.

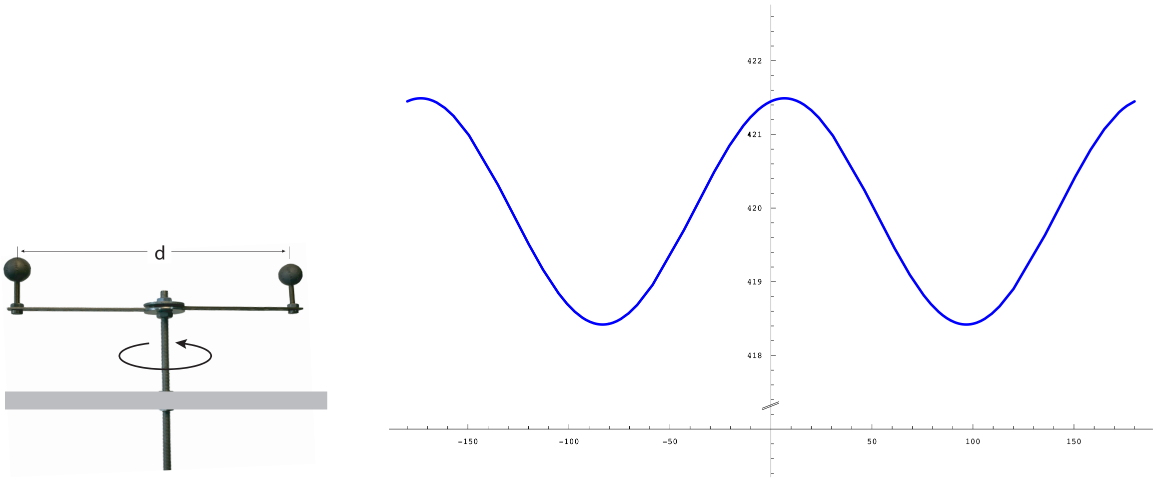
In the errors first a large sinusoidal error in the angular data is denoted. This error comes from a wrong scaling of the three room axes (cf. section 4.1.1). For the analysis that error was removed manually from the data by assuming independent scaling on the three axes.

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, it is assumed for now that they can be approximated with a Gaussian distribution and they can be modeled as a zero-mean noise on the image plane.

Figure 5.8 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 prediction fits the measurements still well.

5.2 Monocular Tracking Systems

Monocular tracking systems are operating with only one sensor. Due to the lack of the second camera it is not possible to track a single spatial point because no triangulation is



(a) (b)

Figure 5.7: Measuring the distance of a rotating two-ball target of rotating balls which are rigidly connected. Figure (a) shows the setup of this experiment; Figure (b) shows the error in length estimation for a rotating two-ball target, the blue line shows assumed room calibration error

possible. By combining more than one feature point rigidly to a marker target the pose of the marker target can be determined if the model of the tracked target is known.

There are two common approaches for monocular tracking, which differ only in the way how the point feature points on the image plane are estimated. The first group of tracking systems using planar targets, which are in general flat squares with identifiable corners and cameras which operate with visible light. By knowing the geometrical model it is possible to obtain all six spatial constraints, by the correspondences of the four points ($M_i \leftrightarrow m_i$) [35]. The other common approach is to use an optical tracking system operating with infrared light and active or passive markers, like described in section 2.2. The presented model for theoretical accuracy of monocular tracking systems does not depend on a special system, if the system can estimate 2D positions of the projected feature points.

Some example of tracking systems using planar targets are:

- ArToolKit marker system [31].
- Institut Graphische Datenverarbeitung marker system[4]
- Siemens Corporate Research (SCR) marker system [63]
- Hoffman marker system used by SCR and Framatome ANP [3]
- Sony Computer Science Laboratory [43]

The pose estimation of planar targets can be done as described in [35]: First the image taken from the camera is thresholded, then regions of connected black pixels are searched.

---

[4] ARVIKA: `http://www.arvika.de/www/`

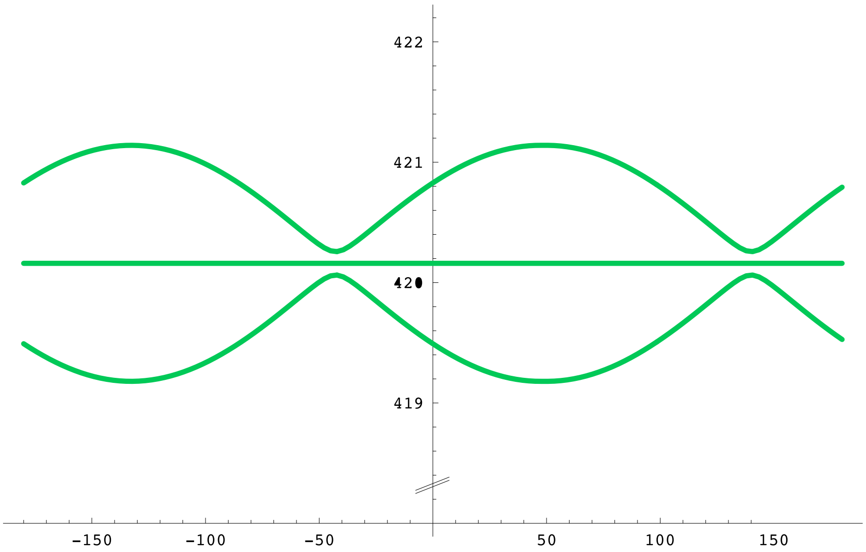


Figure 5.8: Measured errors corrected for room scaling (blue dots) vs. predicted error standard deviation (green line)

Figure 5.9: Examples for ARToolKit markers

That regions which have the shape of a square are assumed to be a target. By intersecting the estimated edges, the positions of the corners on the image plane can be calculated with subpixel accuracy. The position of the edges can then be treated as four feature points, which is the same situation as when tracking four retro-reflective markers for example with an ARTtrack1 camera. The four feature points allow the pose estimation of the target as shown in [23].

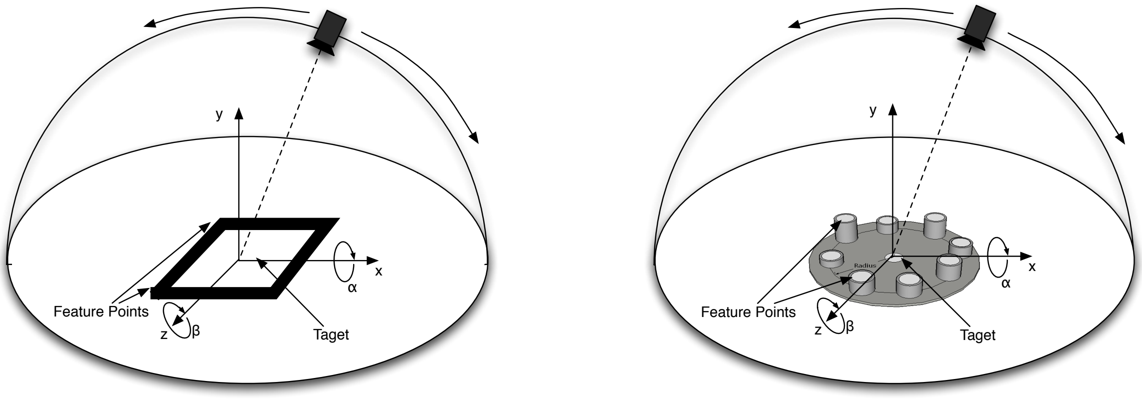
Tracking systems using natural lightning conditions are vulnerable to bad influences like direct sunlight. For example it can happen that the maker target is in the field of view of the camera but it can not be detected in the image. From now on it is assumed that a target can always be recognized.

Zhang et al. [62] did a comparative study of tracking systems using planar targets. Representatively for the other tracking systems the specific features of the ARToolKit will be shown. It's target consists of a thick black border on a white background (cf. fig. 5.9). This increases the contrast and makes it easier to identify the edges, which are used to determine the feature points of the marker. For using more than one target in one tracking volume it is necessary to place a unique symbol or pattern on the target (see fig. 5.9). Owen et al. [41] presented a design for planar arranged feature points, based on a black square, using an image constructed from the discrete cosine transform (DCT) basis function as the identification patten. The resulting targets are optimized to be easy identified and to provide a high and balanced accuracy. For example Malzebin et al. measured in their paper [38] that the position estimation depends strongly upon the orientation of the marker. The approach of Owen tries to minimize that kind of tracking error.

5.2.1 Theoretical Accuracy

By looking at the theoretical accuracy the systematic (biased) errors shown in section 4.1.1 will not be considered, because they can be minimized by improving the environment conditions, the underlying algorithms and by diligent calibration.

In the model it is assumed that all the feature points or the target can be well recognized and that the 2D position estimation is unbiased but not noise free. So the 2D errors in position estimation can be interpreted as a two dimensional jointly gaussian distributed variable with zero-mean. A covariance matrix in two dimensions then describes the un-isotropic jointly normal distribution. Furthermore it is assumed that the topology of the target is known exactly and that the target is fixed in position and orientation and just the camera is moving on a hemisphere heading towards the centroid of the target. This approach is common for analyzing the accuracy of monocular tracking system; it is used for instance by Malzebin et al. [38] and Abawi et al. [1].



(a)

(b)

Figure 5.10: Setup for analyzing the theoretical accuracy of a planar target such as an ARToolKit marker (cf. fig (a)) and of a marker based target (cf. fig (b))

Let the $\mathbf{C}_{\alpha,\beta,t}$ be the extrinsic camera parameter matrix

$$\mathbf{C}_{\alpha,\beta,r} = \begin{pmatrix} & & & 0 \\ & \mathbf{R}_{\alpha,\beta} & & r \\ & & & 0 \\ & \mathbf{0} & & 1 \end{pmatrix} \quad , \tag{5.4}$$

where $\mathbf{R}_{\alpha,\beta} \in \mathbb{R}^{3\times3}$ describes the orientation of the camera depending on the rotation angles α,β and the distance r. So the target is fixed and the camera is moving on a sphere with diameter r always looking towards the target (cf. fig. 5.10).

If $u \in \mathbb{R}^3$ is a spatial point in the world coordinate frame then the multiplication of u with the inverse of $C_{\alpha,\beta,r}$ results in the corresponding point $v \in \mathbb{R}^3$ in the camera coordinate frame (cf. eq. (5.5)).

$$x_{\text{cam}} = C^{-1}_{\alpha,\beta,r} x_{\text{world}} \tag{5.5}$$

To project the feature points onto the image plane the pinhole camera model is used. The projection $\mathrm{p}_{\mathrm{f}}(v)$ is shown in equation (5.6) where f represents the focal length of the camera.

$$\mathrm{p}_f : \mathbb{R}^3 \to \mathbb{R}^2, x_{\text{cam}} \mapsto \mathrm{p}_f(v) = \frac{f}{v_y}(v_x, v_z)^t \tag{5.6}$$

The complete projection $\mathrm{g} : \mathbb{R}^3 \to \mathbb{R}^2$ can then be written as

$$\mathrm{g}(u) := (\mathrm{p}_f \circ C^{-1}_{\alpha,\beta,r})(u) \quad . \tag{5.7}$$

Assuming the target consists of n feature points $(u_1, \ldots, u_n)$ where the centroid coincides with the origin of the world coordinate frame then these features are projected to the points $(w_1, \ldots, w_n)$ on the image plane by the function $\mathrm{f} : \mathbb{R}^{3n} \to \mathbb{R}^{2n}$

$$\mathrm{f}_{\alpha,\beta,r}(u_1, \ldots, u_n) := (\mathrm{g}_x(v_1), \mathrm{g}_y(v_1), \ldots, \mathrm{g}_x(v_n), \mathrm{g}_y(v_n)) \tag{5.8}$$

The covariance for estimating the position of projected feature points on the image plane is assumed to be known and the covariance of the target pose is desired, so the backward propagation formula has to be applied (cf. section 3.4.2). Estimating the covariance for detecting a feature point on the image plane can be experimentally determined [53]. Because f is not linear, the Jacobian of f has to be calculated and evaluated at the centroid of the target.

$$J_{\mathrm{f}}|_{\bar{u}} = \frac{\partial\, \mathrm{f}_{\alpha,\beta,\mathrm{r}}}{\partial(\alpha, \beta, f)} \in \mathbb{R}^{3\times2n} \tag{5.9}$$

If $\Sigma_{w_i} \in \mathbb{R}^{2\times2}$ is the covariance for detecting the position of a feature point on the image plane then the covariance for estimating the distance r and the angles α and β is given by

$$\Sigma_{\alpha,\beta,r} = (J^t_{\mathrm{f}} \begin{pmatrix} \Sigma_{w_1} & & \\ & \ddots & \\ & & \Sigma_{w_n} \end{pmatrix}^{-1} J_{\mathrm{f}})^{-1} \in \mathbb{R}^{3\times3} \tag{5.10}$$

Now the covariance of the angles α, β and the distance r can be predicted for any position of the camera.

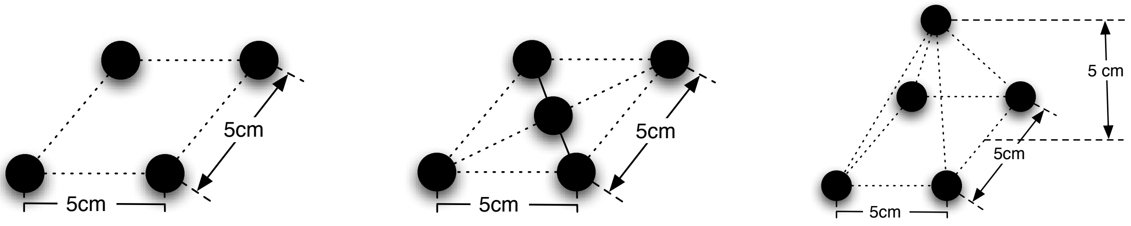
5.2.2 Visualization of Uncertainty

In this section three different targets are analyzed with the presented model. The first target is a coplanar target with four feature points (cf. fig. 5.11(a)), comparable to an ARToolKit marker, the second target is like the first but with an additional feature point in the centroid (cf. fig. 5.11(b)) and the third target is a non-coplanar target where the base coincides with the first target.

Camera moving along a Path

To visualize the accuracy the camera is moving on a path only changing the angle $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ leaving α fixed to zero. The distance from the camera to the centroid of the target is set to $r = 60\text{cm}$. Figure 5.10 is illustrating the chosen path. Along this path the uncertainties for detecting the three parameters α, β and r are estimated. The uncertainty for detecting a feature point on the image plane is set to the covariance matrix shown in eq. (5.11). These values are estimated with a two-camera setup with retro-reflective markers therefore the prediction of accuracy will be far too optimistic. So the uncertainties presented here can not be taken literally. For quantitative predictions realistic covariances have to be estimated experimentally.

$$\Sigma_{w_i} = \begin{pmatrix} 4.14 * 10^{-15}\text{m}^2 & 0 \\ 0 & 4.14 * 10^{-15}\text{m}^2 \end{pmatrix} \tag{5.11}$$



(a) (b) (c)

Figure 5.11: Figure (a) shows a coplanar target with four feature points which corresponds to an ARToolKit marker (*target1*), figure (b) shows a coplanar target with five feature points (*target2*) and figure (c) shows a non-coplanar target with five feature points (*target3*)

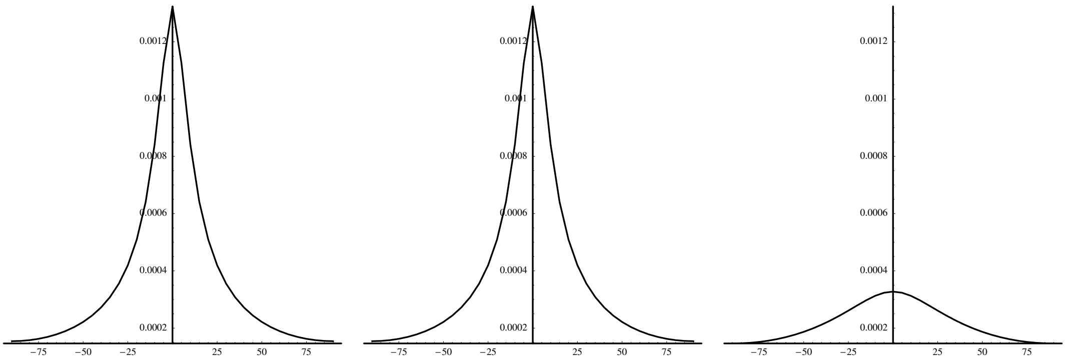
The plots in figure 5.12 are showing the rotational uncertainty and figure 5.13 the uncertainty for estimating the distance.

Rotational Uncertainty The accuracy for detecting the orientation is worst for all three targets if the camera is heading perpendicular to the target; this is caused by minimal changes in the projected geometry. The accuracy grows for all three targets as the camera views more angular to the target. Similar results are reported in [53, 1].

The non-coplanar design of target3 results in a significant greater accuracy (cf. fig. 5.12(c)) for detecting the orientation as the coplanar designs. The greater accuracy is presumably caused by the different distance between the feature at the apex of the pyramid and the camera against the distances between the coplanar features and the camera. As the camera is rotating this projected feature at the apex changes position on the image plane differently as the other features which leads to a better orientation estimation.

The better rotational accuracy of target3 is canceled out if the camera is viewing with about 90 degree to the target, but at this angle a real target would not be recognized anymore by the tracking system.

Vogt et al. [53] are also reporting the better performance of non-coplanar against coplanar targets in a monocular tracking system.

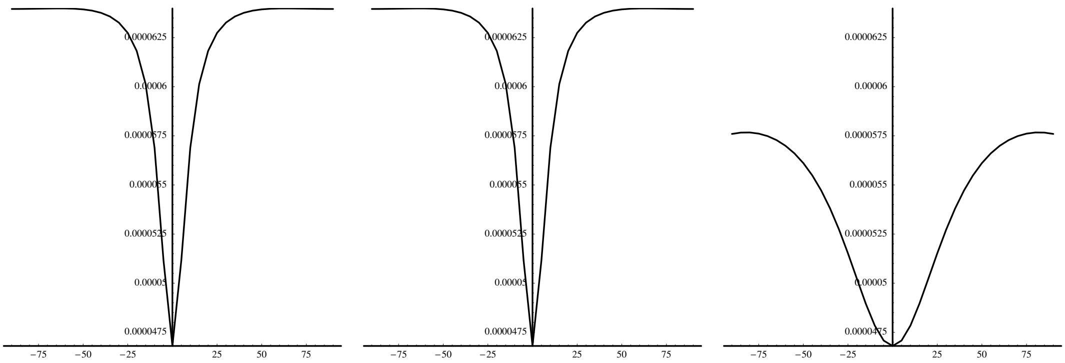


(a) (b) (c)

Figure 5.12: The figures (a), (b) and (c) showing the rotational uncertainty of target1, target2 and target3; the abscissae address the angle β and the ordinates correspond to the rotational rms error

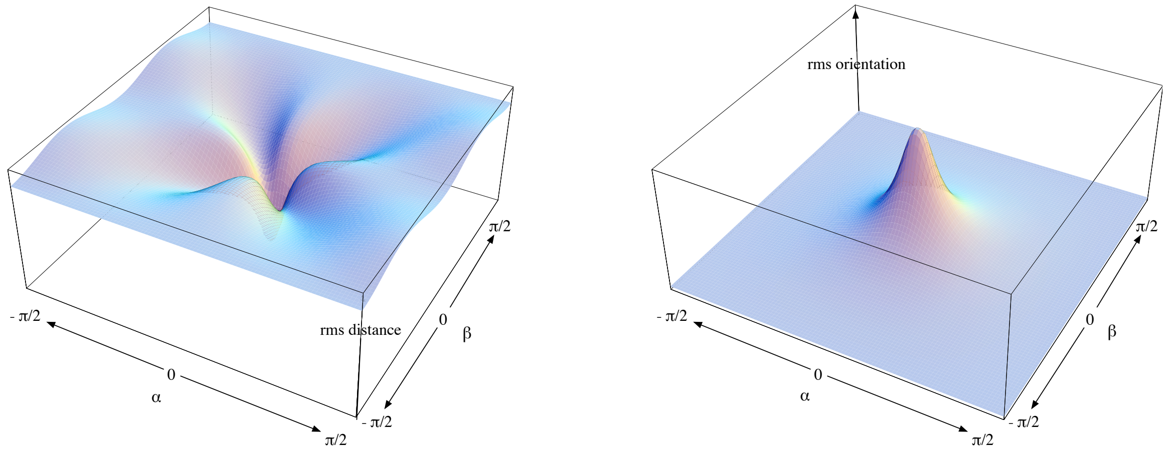
Distance Uncertainty The accuracy is best if the principal axis of the camera is perpendicular to the target for all three analyzed targets. In contrast to n-occular systems an additional feature point in the centrois is not increasing the positional accuracy (cf. fig.5.13(a) and fig. 5.13(b)). But as the plots are showing the errors in distance estimation, it is easy to accept that the additional feature in the centroid does not change geometrical attributes, as the distance between this feature and the camera is constant.

Vogt et al. are also reporting in [53] that the distance estimation accuracy is increased significantly when using non-coplanar targets.



(a) (b) (c)

Figure 5.13: The figures (a), (b) and (c) showing the uncertainty in detecting the distance to target1, target2 and target3; the abscissae address the angle β and the ordinates correspond to the standard deviation of errors in distance estimation



(a) (b)

Figure 5.14: 3D plots showing qualitative uncertainties with respect to the angles α, β at fixed distance r; figure (a) is visualizing the rms error of estimating the distance to the target and figure (b) shows the rms error of estimating the orientation

Camera moving on Hemisphere

The next experiment is showing the rms errors of detecting the distance and the orientation from several sample positions on the hemisphere with radius $r = 0.6$m. Figure 5.14 is showing the predicted accuracies for target1.

Rotational Uncertainty The rotational uncertainty in both angles is maximal if the camera is viewing perpendicular to the target (cf. fig. 5.14(b)). The symmetry of the rotational errors is caused by the symmetry of the target.

Distance Uncertainty The distance uncertainty of target1 can be seen in 5.14(a) which is showing an unexpected plot. In the origin – the camera is looking then perpendicularly to the target – the uncertainty is minimal which is equal to the previous results. The plot shows also that the uncertainty in distance is greater when the camera is viewing perpendicularly to the edges of the target as when viewing angularly to the target. This is again caused by the different projection of the feature points on the image plane.

6 Accuracy of Marker Targets

In this chapter it will be shown how to predict the pose accuracy of optical marker targets based on the accuracy of detecting the spatial position of a single marker ball.

6.1 Basics

A marker target consists of at least three [27] feature points which are combined rigidly (cf. figure 2.2(a)). With help of a n-ocular optical tracking system the spatial positions of these marker balls can be measured and if the geometrical model of the target is given, the pose of the whole target can be calculated. The process of finding this relationship is called 3D-to-3D pose estimation or absolute orientation problem. Solutions for this problem can be found in eg. [27] and [64]

But as the spatial position measurement is always erroneous, the pose estimation will also be erroneous. The idea is now to predict the pose accuracy of the whole target based upon the geometrical model and the known spatial position accuracy of the single feature points. Therefore a model is developed which takes the covariance matrices representing the accuracy of each spatial position measurement and delivers a covariance matrix representing the accuracy of the pose of the whole target. Other accuracy prediction models like [19] and [15] are only using one single rms for uncertainty in spatial position estimation, which corresponds to a isotropic error distribution.

Furthermore the covariance forward propagation is used to calculate the accuracy at special points of interest instead of in the centroid which is necessary for many applications.

Without loss of generality it is considered that the marker target consists of feature points $\{q_k\}$ in a local coordinate system with the origin in the centroid of the feature points, and their respective counterparts $\{p_k\}$ in the tracker coordinate system, which are additively disturbed by zero-mean uncorrelated errors $\{\Delta p_k\}$ per coordinate.

6.2 Uncertainty in the Centroid

From the points $\{q_k\}$ and $\{p_k\}$ the pose of the target is estimated by solving the *3D-3D Pose Estimation* problem using any kind of algorithm.

This estimation leads to a homogeneous transformation H which maps

$$p_k + \Delta p_k = H q_k$$

with some error Δp_k for every k. From the error function $\mathrm{f}(p,q) = Hq - p$ the Jacobian is calculated

$$J_{\mathrm{f}(q)} = \left.\frac{\partial\,\mathrm{f}(p,q)}{\partial \vec{x}}\right|_{\vec{x}=\vec{0}} \tag{6.1}$$

with respect to the 6D pose $\vec{x} = (x, y, z, \alpha, \beta, \gamma)^t$. For the representation of rotation fixed Z-Y-X euler angles are used (cf. section 3.1.2).

Without loss of generality it can be assumed that the estimated pose H is the identity, i.e. zero translation and rotation — the coordinate system can be transformed in such way that all coordinates p_i and the covariances Σ_{p_i} are given in the target coordinate system — then the Jacobian is evaluated at the pose $\vec{0}$,

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

This Jacobian maps the 6D pose error Δx of the target to the respective 3D feature errors

$$\Delta P = \begin{pmatrix} \Delta p_1 \\ \vdots \\ \Delta p_n \end{pmatrix} = \begin{bmatrix} J_f(q_1) \\ \vdots \\ J_f(q_n) \end{bmatrix} \Delta x = M \Delta x$$

As this system of equations is over determined it can not be solved directly and so the pseudoinverse (cf. section 3.4.2) is used to get a least square solution for Δx

$$\Delta x = M^{\ddagger} \Delta P \quad . \tag{6.2}$$

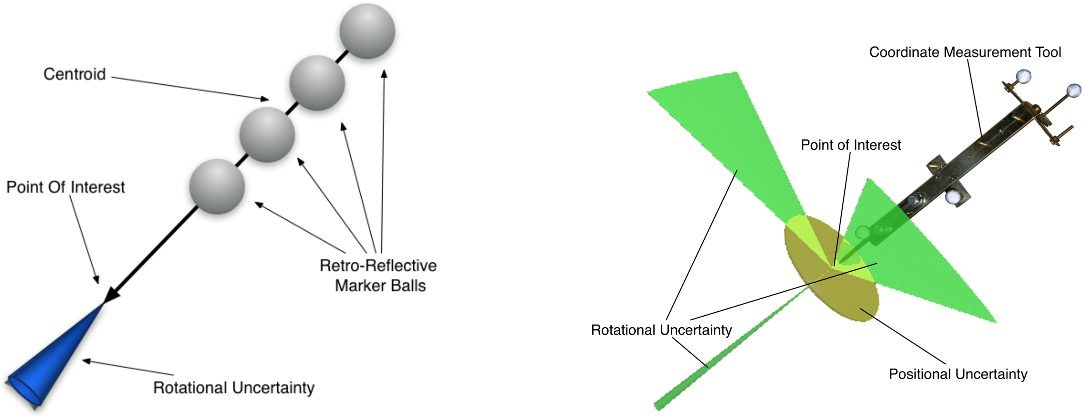
Using the linear forward propagation formula 3.35 leads to

$$\begin{aligned} \Sigma_{\vec{c}} &= \mathrm{E}(\Delta x \Delta x^t) = \mathrm{E}((M^{\ddagger}\Delta P)(M^{\ddagger}\Delta p)^t) \\ &= \mathrm{E}((M^{\ddagger}\Delta P)(\Delta P^t {M^{\ddagger}}^t)) \\ &= M^{\ddagger}\,\mathrm{E}(\Delta P \Delta P^t) {M^{\ddagger}}^t \\ &= M^{\ddagger} \begin{pmatrix} \Sigma_{p_1} & & \\ & \ddots & \\ & & \Sigma_{p_n} \end{pmatrix} {M^{\ddagger}}^t \in \mathbb{R}^{6\times 6} \end{aligned} \tag{6.3}$$

for the covariance $\Sigma_{\vec{c}}$ in the centroid of the marker target, where Σ_{p_i} are the covariance matrices in the target coordinate system. The upper-left 3×3 matrix represents the positional uncertainty and the lower-right 3×3 matrix the rotational uncertainty. The upper-right and lower-left 3×3 matrices represent the correlation between the positional and rotational uncertainty which is in general small and negligible.

6.3 Point Of Interest

The equation (6.3) can only predict the accuracy in the centroid of the target, but in most applications it is insufficient to know the target uncertainty in the centroid. A surgeon for example uses a point device to mark a point on the patient's body, then the uncertainty is desired at the tip of device. Figure 6.1(a) showing the accuracy of such a point device superimposed with an image of the device and figure 6.1(a) shows a drawing of a point device where the features are aligned in a row, which is only providing two DoF in rotation.



(a)

(b)

Figure 6.1: Figure (a) shows a pointing device with rotational uncertainty; Figure (b) shows the visualization of predicted positional and rotational uncertainty of a pointing device

From this 6D covariance in the centroid the 3D error covariance $\Sigma_{\vec{p}}$ can be computed at a point other than the centroid by applying the forward propagation formula ((3.39)). Again the Jacobian equation 6.1 evaluated at the point of interest $\vec{p}$ is used.

The target registration error at the point of interest is then given by

$$\Sigma_{\vec{p}} = J_f \Sigma_{\vec{c}} {J_f}^T \tag{6.4}$$

Assuming that the rotational part of the error in the centroid is independent from 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 an positional error coming from the rotational error [19]. As the negative influence of the rotational error on the positional error grows with the distance to the centroid, the target should be placed as close as possible to the object to minimize the error [24]. In the centroid the target location error is minimal [58].

6.4 Validation of the Pose Accuracy

In this section the prediction of accuracy of marker targets is validated with a Monte Carlo simulation. Monte Carlo simulations are a common way to proof a model, for example Vogt et al. [53] and Hoff et al. [24] used Monte Carlo simulations to validate their models.

For the validation a spatial position detection uncertainty given by a covariance matrix Σ_{pos} (cf. fig 6.2) is assumed equal for each marker. Then the pose accuracy Σ_{pose} of the target consisting of the marker at positions $\{q_i\}$ is calculated with the formulas presented in this chapter. Since the used computer algebra system could not create random numbers with a realistic covariance matrix the uncertainty Σ_{pos} for detecting a single feature point had to be scaled by a factor $1\;10^3$.

$$\Sigma = \begin{pmatrix} 4.73\;10^{-7} & 1.15\;10^{-7} & 6.71\;10^{-7} \\ 1.15\;10^{-7} & 1.97\;10^{-6} & 2.41\;10^{-6} \\ 6.71\;10^{-7} & 2.41\;10^{-6} & 4.56\;10^{-6} \end{pmatrix}$$

Figure 6.2: Assumed spatial position estimation uncertainty for a single marker

For the simulation the set of marker positions $\{q_i\}$ is tainted with normal distributed zero-mean errors ($\sim \mathrm{N}(0, \Sigma_{\text{pos}})$) which leads to a simulated set of measurements $\{p_i\}$. Then the absolute orientation of this noisy target $\{p_i\}$ is calculated with the closed-form solution of Horn [24]. The step of creating erroneous and calculating the pose is repeated several times. Then the covariance matrix can be calculated from the set of poses gained from the simulation process. The positional and rotational parts of the simulated and predicted covariance matrices for a regular tetrahedron with edge length 10cm can be seen in figure 6.4.

$$\begin{pmatrix} 2.821\;10^{-4} & -2.720\;10^{-6} & -3.279\;10^{-5} \\ -2.720\;10^{-6} & 2.345\;10^{-4} & -9.270\;10^{-5} \\ -3.279\;10^{-5} & -9.270\;10^{-5} & 1.012\;10^{-4} \end{pmatrix}$$

(a)

$$\begin{pmatrix} 2.941\;10^{-4} & -3.376\;10^{-6} & -3.365\;10^{-5} \\ -3.376\;10^{-6} & 2.389\;10^{-4} & -9.685\;10^{-5} \\ -3.365\;10^{-5} & -9.685\;10^{-5} & 1.028\;10^{-4} \end{pmatrix}$$

(b)

$$\begin{pmatrix} 1.183\;10^{-7} & 2.875\;10^{-8} & 1.678\;10^{-7} \\ 2.875\;10^{-8} & 4.925\;10^{-7} & 6.025\;10^{-7} \\ 1.678\;10^{-7} & 6.025\;10^{-7} & 1.140\;10^{-6} \end{pmatrix}$$

(c)

$$\begin{pmatrix} 1.182\;10^{-7} & 3.073\;10^{-8} & 1.703\;10^{-7} \\ 3.073\;10^{-8} & 4.898\;10^{-7} & 6.044\;10^{-7} \\ 1.703\;10^{-7} & 6.044\;10^{-7} & 1.148\;10^{-6} \end{pmatrix}$$

(d)

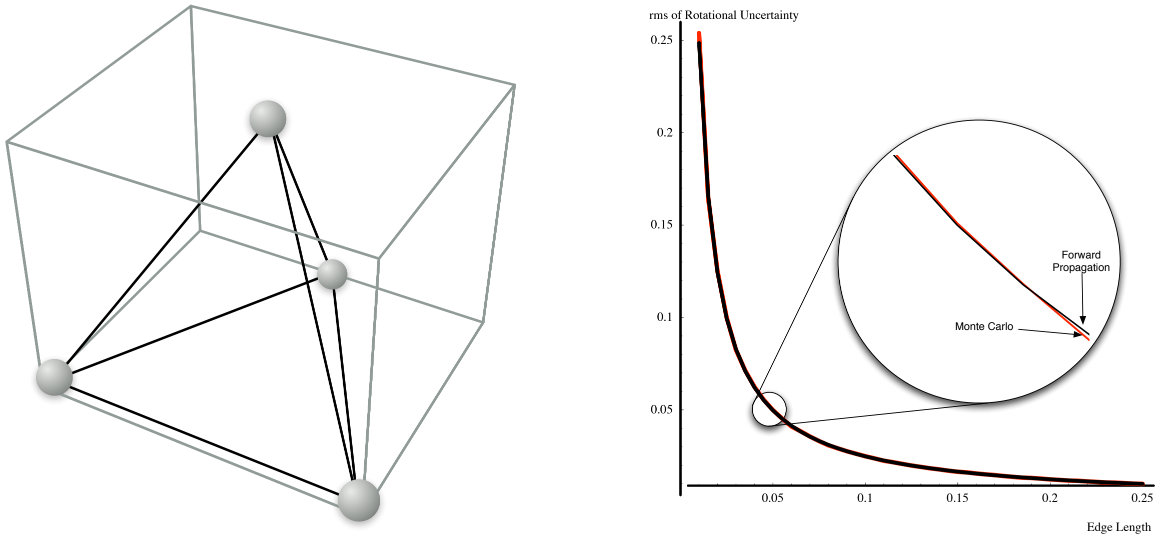
Figure 6.3: Rotational and positional uncertainty covariance matrices of a tetrahedron with edge length of 10cm; Figure (a) and (c) showing the predicted rotational and positional accuracy; Figure (b) and (d) showing the corresponding accuracies determined with Monte Carlo simulation (100000 samples)

To validate the prediction of the target accuracy regarding to geometric layout and the number of markers, the validation is now done for different target configurations.

6.4.1 Rotational Uncertainy

For the validation of rotational uncertainty a target with a shape of a regular tetrahedron with feature points are mounted on the vertices and variable edge length is chosen (cf. figure 6.4(a)). This allows to validate the predicted rotational uncertainty of a target at different sizes. As the rotational uncertainty is independent from the number of feature point [58], the amount of feature points will not be varied.

The plot in figure 6.4(b) shows the root mean square of the rotational uncertainty calculated with the forward propagation formula (black line) and with Monte Carlo method (red line). Figure 6.4.1 shows the positional rms errors for some target configurations. The average difference between these two root mean square errors is $\approx 0.74\%$.



(a)

(b)

Figure 6.4: Validation of the rotational accuracy is done with a target with the shape of a regular tetrahedron with variable edge length (see fig.6.4(a)); Figure 6.4(b) shows the plot of the rotational rms at different edge lengths. The red line represents the rotational rms determined by Monte Carlo simulation and the black line the predicted rms.

	Edge length of regular tetrahedron						
	0.1	0.125	0.15	0.175	0.2	0.225	0.25
predicted rms	0.02486	0.01989	0.01657	0.0142	0.01243	0.01105	0.009943
simulated rms	0.02489	0.01987	0.01649	0.01418	0.01238	0.01102	0.009933

Table 6.1: The rms of the predicted and simulated rotational uncertainty in degrees

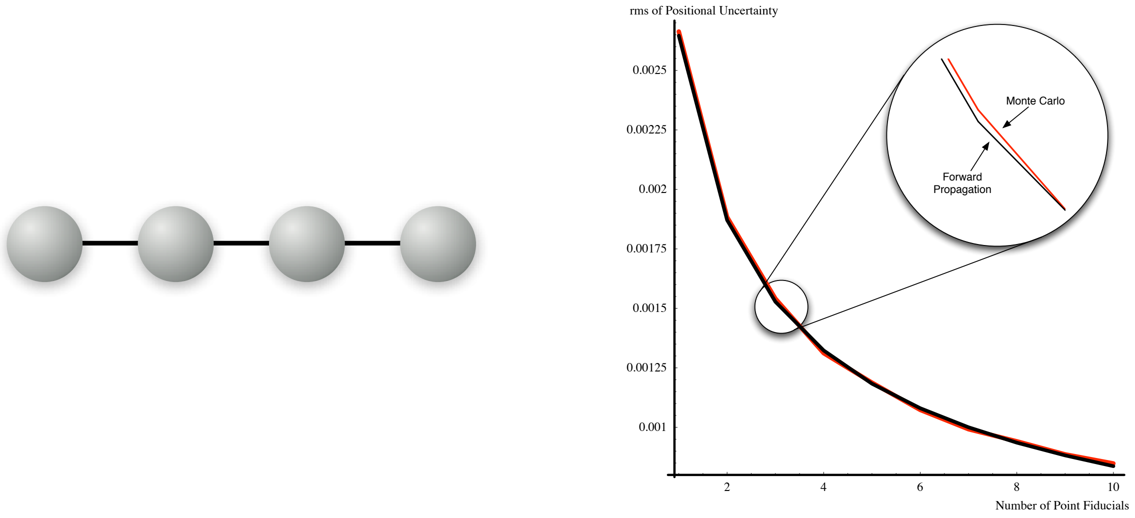
6.4.2 Positional Uncertainty

The positional uncertainty of a marker target is only depending on the amount of feature points used but not on their geometrical distribution [58]. For the validation of the predicted positional part a target is chosen where the amount of feature points can be varied. The feature points of the target are arranged equidistant in a row, which allows to add new feature points by and by (cf. 6.5(a)).

The plot in figure 6.5(b) shows the dependency of positional uncertainty and the amount of features. The red line in the plot shows the rms of positional uncertainty simulated with

Monte Carlo and the black line shows the rms of positional uncertainty predicted with the forward propagation. Table 6.2 is showing the predicted and simulated positional rms errors.

The average difference between these two errors is $\approx 0.43\%$, so it can be stated that there is only a minimal difference between the simulation and the prediction.



(a)

(b)

Figure 6.5: For the validation of the positional uncertainty a target is chosen where the markers are arranged in a row (cf. fig. 6.5(a)); The rms of the predicted positional uncertainty (black line) and the rms of the simulated uncertainty (red line) can be seen in figure6.5(b). This figure is showing also the decrease of uncertainty with the increasing number of feature points.

	Number of markers				
	1	2	3	4	5
rms predicted	$1.53\ 10^{-3}$	$1.32\ 10^{-3}$	$1.18\ 10^{-3}$	$1.08\ 10^{-3}$	$1.00\ 10^{-3}$
rms simulated	$1.54\ 10^{-3}$	$1.31\ 10^{-3}$	$1.19\ 10^{-3}$	$1.07\ 10^{-3}$	$9.92\ 10^{-4}$

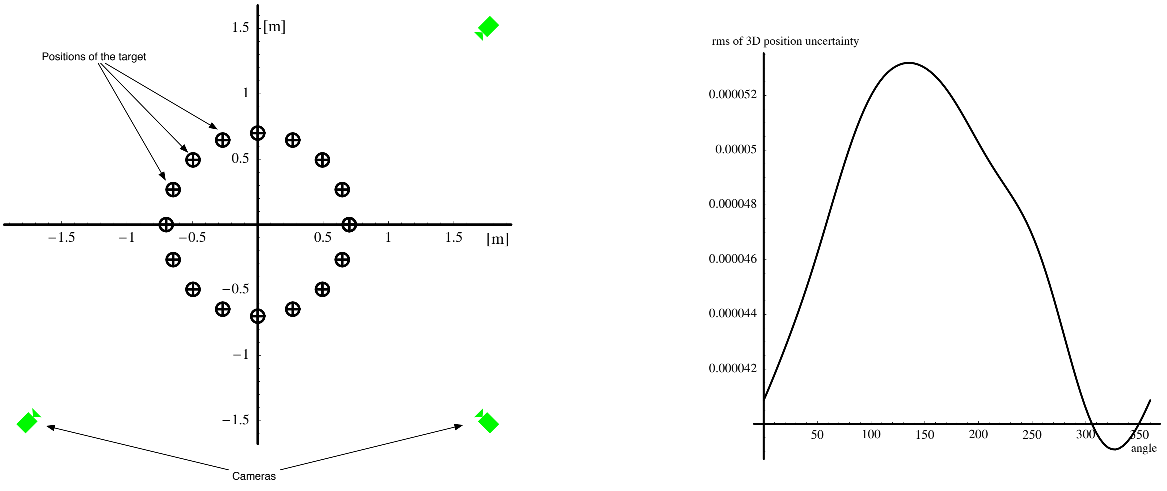
Table 6.2: The rms of the predicted and simulated rotational uncertainty in meters

6.5 One Uncertainty for all Features

In this section the assumption that all features are detected with the same accuracy is analyzed with respect to the prediction of target accuracy.

When predicting the accuracy of a marker target the uncertainty for estimating the spatial positions of the feature points is needed for each feature separately. Because the spatial position estimation uncertainty is changing over the working volume all features points will be located with different accuracy and so the spatial position estimation covariance matrix has to be known for each feature separately. The covariance matrices can be predicted with methods presented in section 5.1.2. As the difference of spatial position estimation uncertainties is small within the radius of a typical marker target, it is valid taking the uncertainty of one position eg. the centroid and assuming the same uncertainty for all other features. For predicting the accuracy of a target in real time during the tracking process this simplification is not necessary because predicting the spatial uncertainty for each marker can be done fast, but when assessing tracking setups offline this assumption can save much computational time.

To show that this assumption does only create only small errors, the rms of the positional and rotational uncertainties calculated with both methods will be compared. Therefore a setup with three cameras is chosen and the target is moving on a circular path in the middle of the working volume (cf. 6.6(a)). Again a regular tetrahedron with edge length 10cm and feature points at the vertices is used as target.



(a)

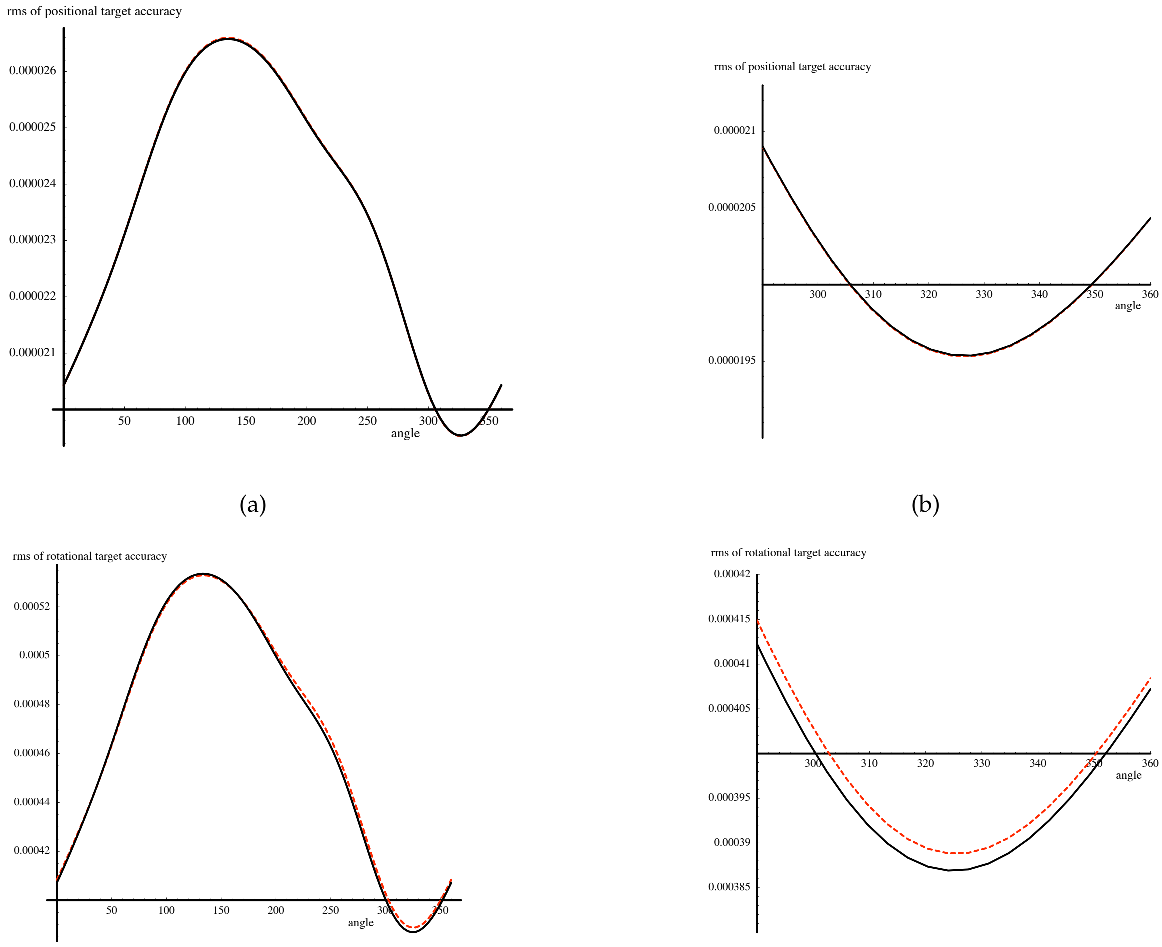
(b)

Figure 6.6: The setup for validating the assumption of section 6.5 can be seen in figure (a). The target will be moved along the path in the middle of the tracking volume; The plot in figure (a) shows the rms of the positional uncertainty detecting a single feature along the circular path

The plot in figure 6.6(b) shows how the rms of positional uncertainty for detecting the position of single feature points is varying along the circular path.

In figure 6.7(a) and 6.7(c) plots of the rms errors of the positional and rotational uncertainty can be seen, because the differences between the two estimation methods are so small, the figures 6.7(b) and 6.7(d) are showing the plots only in the range from 290° to 360°. The two positional rms errors are nearly identical, the difference between the two rotational rms errors is visible but still small. The mean difference of the two positional rms errors is only about $-7.28\ 10^{-9}$m where the mean of rms errors is about $2.34\ 10^{-5}$m, and the mean of the rotational rms errors is $9.48\ 10^{-7\circ}$ where the mean of both rotational rms errors is about

$4.67\ 10^{-4\circ}$. So the differences between these two error estimation methods can be ignored using a usual target.



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

Figure 6.7: The rms of the positional and rotational uncertainty of a target along a circular path. Figure (b) and (b) is showing the positional errors. Figure (c) and (d) showing the rotational errors. The red dashed lines represent the rms when assuming the same 3D accuracy for all feature points, the black lines show the rms with separate accuracies for each feature.

6.6 Visualization of the Pose Uncertainty

As shown above, a covariance matrix can be interpreted as quadric, but as the covariance matrix of a pose is $\in \mathbb{R}^{6\times 6}$ this would yield a quadric in six-dimensional space which is impossible to visualize. So the covariance matrix Σ_p has to split up into two submatrices $\in \mathbb{R}^{3\times 3}$; One for the positional and the other for the rotational uncertainty. Equation eq. (6.5) shows the separation.

$$\Sigma_p = \begin{pmatrix} \Sigma_{\text{pos}} & * \\ * & \Sigma_{\text{rot}} \end{pmatrix} \tag{6.5}$$

6.6.1 Positional Uncertainty

The positional uncertainty Σ_{pos} can be visualized with a ellipsoid, as described in section 3.3.

6.6.2 Rotational Uncertainty

Visualizing the rotational uncertainty with an ellipsoid would be technically possible but it would not be intuitive for the user at all.

One idea for visualizing the rotational uncertainty is to view how the rotational error is affecting a single point. Let be $e = (e_x, e_y, e_z)^t$ a point in the target coordinate frame and R_{err} depending on the euler angles (α, β, γ) an error in rotation then $\tilde{e} = R_{\text{err}}e$ would be the point e affected by a rotational error. As the rotational error given by a covariance is normally distributed with zero-mean the forward propagation rule can be applied. Therefore the Jacobian of mapping $R_{\text{err}}e$ is needed which can be calculated as

$$J_{R_{\text{err}}e}\big|_{\alpha,\beta,\gamma=0} = \left.\frac{\partial R_{\text{err}}e}{\partial(\alpha,\beta,\gamma)}\right|_{\alpha,\beta,\gamma=0} = \begin{pmatrix} 0 & e_z & -e_y \\ -e_z & 0 & e_x \\ e_y & -e_x & 0 \end{pmatrix} \quad . \tag{6.6}$$

Now the covariance forward propagation can be applied:

$$\Sigma_e = J_{R_{\text{err}}e} \Sigma_{rot} J^t_{R_{\text{err}}e} \tag{6.7}$$

The resulting covariance matrix Σ_e is of rank two. Decomposing the covariance matrix with eigenvalue decomposition leads to a rotation matrix and a diagonal matrix of the eigenvalues $\lambda_i >= 0$. One and only one of these eigenvalues is then zero, because the matrix is of rank 2. So there are two principal axes which represent an ellipse. With help of the rotation matrix the ellipse can be rotated into the right orientation.

To enhance this visualization the ellipse could be substituted with a elliptical cone, where the cone's axis coincides with the vector e and the base coincides with the ellipse represented by the covariance matrix Σ_e. The position of the cone is arbitrary, because the rotational error is invariant under translational operations.

Hoff et al. suggest in [24] to visualize the rotational error along the three coordinate axes. The visualization along three axes is useful for a non interactive presentation of the errors. By the usage of an interactive system one single cone would be sufficient as the cone could be rotated and therefore all errors could be visualized.

6.6.3 Covariance along a Path

In several applications, typical movements can be identified for which the uncertainty is desired at some point. To achieve this, the path of the target for the desired typical action can be recorded with a tracking system. Then the covariances along this path can be analyzed offline similar to the tool presented in section 7.3. Figure 6.8 shows an example of a prerecorded path (black) of the target centroid together with predicted covariances at discrete locations, as well as the respective propagated covariances at the path of the point of interest (dotted blue).

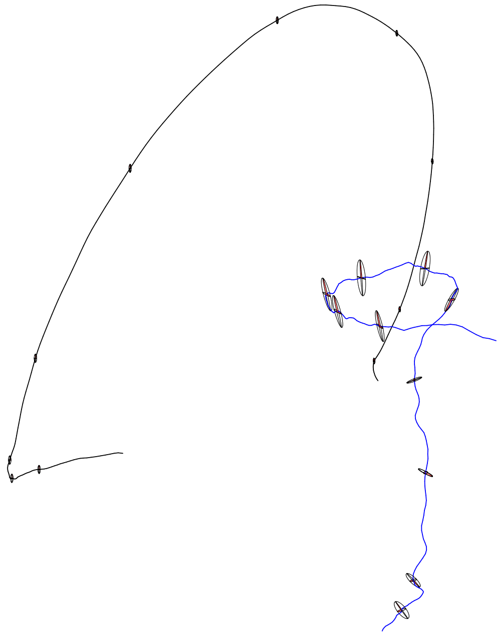


Figure 6.8: Example track of the covariance along a recorded path (black) and propagated to the point of interest (dotted blue)

It is easy to see that the propagated covariance is quite different for poses where the orientation of the target is different. While covariances in the centroid are quite similar along the whole path, the resulting covariance at the point of interest differs substantially along over time. This is due to the fact that the already unisotropic covariances at the centroid get propagated again in an unisotropic way. For unfavorable constellations then the already larger error in one direction gets augmented above average.

In the four-camera setup this happens when the line from the centroid to the tip of the tool is vertical, i.e. orthogonal to the plane of the four cameras; in a stereocamerasystem this would be the case with the line oriented along the optical axes of the cameras.

7 Tools

In this chapter the developed tools will be described. The most basic tools are the methods developed for the used computer algebra system. It provides a set of general methods to calculate accuracies of optical tracking systems. It will be described in section 7.1.

A tool for visualizing the accuracy of a tracking volume, depending on the type of cameras and their spatial position and orientation, is discussed in section 7.2.

In section 7.3 a tool will be presented for designing an optical marker target consisting of multiple features. With help of this tool it is possible to study the different kinds of positional and rotational uncertainties at design time.

7.1 Mathematica Toolbox

Mathematica is a product of the Wolfram Research, Inc. First released in 1988 it allows to describe and solve mathematical problems in a symbol manner, in contrast to numerical tools.

By formulating the problems of forward, backward propagation and others in a symbolic way, allow to export the solutions to programming languages like *Java*, *C*, *C++* or to numerical tools. For the development of the *Covariance Field Viewer* (cf. sec. 7.2) this approach was chosen.

In the next the most useful methods will be described.

Absolute Orientation Problem In this method `Horn3d3d` Horn's closed-form solution for the absolute orientation problem using quaternions [27] is implemented. It calculates a coordinate frame transformation from a set of measured 3D positions to a given set of model points in a least square manner.

This method can be used to calculate the rotational and positional uncertainty of a marker target with a Monte Carlo simulation. Assuming the 3D position estimation accuracy is known in form of a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ then the true positions p_i of the model will be tainted with errors $\{\Delta p_i\}$ ($\sim \mathrm{N}_3(0,\Sigma)$) which leads to simulated measured positions $\{q_i\}$. Using Hoff's solution a quaternion and a translations for mapping $\{q_i\}$ onto $\{p_i\}$ can be calculated.

Positional uncertainty of tracking a single feature The method `CalcCovariance` calculates the accuracy of estimating a 3D position of a feature point with a n-ocular tracking system. The accuracy is depending on the position and orientation of the participating cameras, their focal lengths and the 2D position estimation accuracies on the image plane. The implementation is based on the covariance backward formulas shown in chapter 5.1. Figure 7.1 shows a setup with three cameras; the positional uncertainties calculated with the `CalcCovariance` module are plotted as ellipses. For visualization purposes the ellipses are magnified.

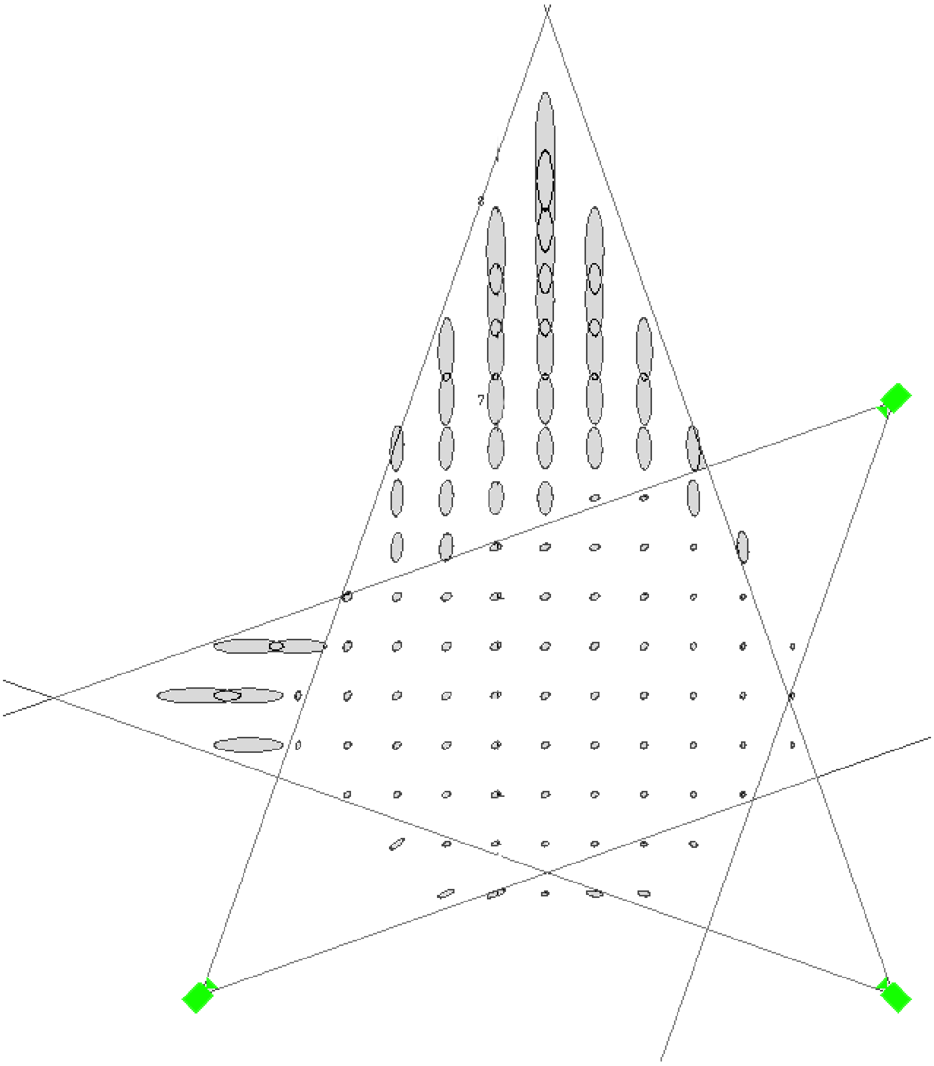


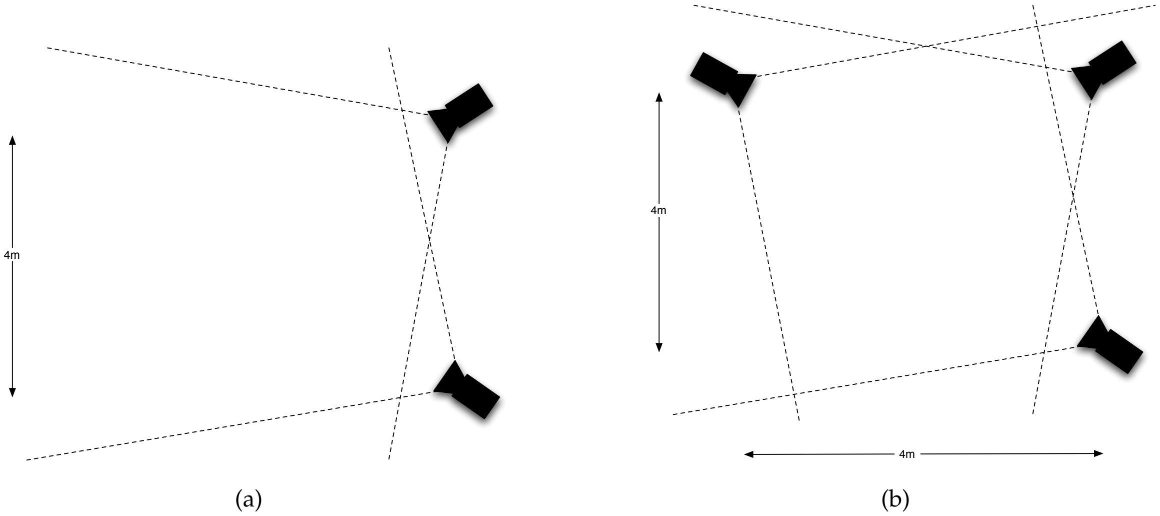
Figure 7.1: 3-ocular tracking setup; the ellipses representing the spatial uncertainty of detecting a single feature point.

Pose Covariance of a Marker Target The module `Hoff` can be used to predict the full 6DoF pose covariance matrix of a marker target consisting of multiple features with given spatial position uncertainty. The prediction is based on the formulas presented in chapter 6.

7.2 Covariance Field Viewer

With the help of the *Covariance Field Viewer* it is possible to visualize the positional accuracy of a n-ocular optical tracking setup.

The tracking setup is defined by the poses of the cameras, their focal length, the view frustums and the 2D position estimation accuracies on the image plane. The spatial position accuracy of a feature point is then calculated with the formulas presented eq. (3.41) and chapter 5. Each point of the defined working volume is tested for the visibility of all participating cameras, so only that cameras are used for the accuracy calculation which can see the feature.



(a) (b)

Figure 7.2: Camera setups used for visualizing the error for tracking a single feature point

The tool is implemented in the programing language Java and Abstract Window Toolkit (AWK) for Java was used to build the user interface. The formulas for calculating the accuracy of one, two, three or four cameras were derived in Mathematica and exported with help of a third party Mathematica package[1]. The maximal amount of four cameras is not a constraint of the design and can be extended to more cameras by exporting a new formulas from Mathematica.

To visualize the 3×3 positional covariance it is necessary to find a suitable mapping which maps the covariance matrix Σ to a single scalar value ($f : \mathbb{R}^{3\times3} \mapsto \mathbb{R}$). Several mappings were examined and visualized with two camera configurations which can be seen in figure 7.2.

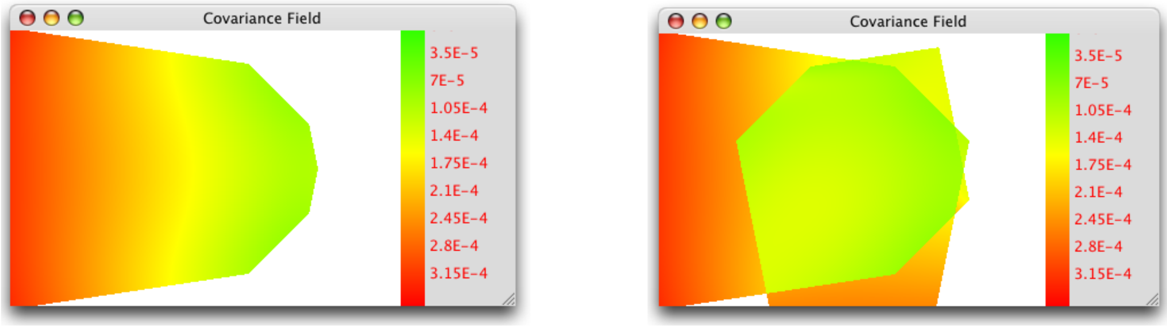
Using the rms error for visualizing the accuracy in the working volume is fast and useful for most applications. The other mappings are either nearly equal to the rms (volume) or only suitable for special purposes (maximum).

RMS The root mean square is a common measurement for errors and the calculation for covariance matrices can be done in a fast and easy way by taking the square root of the

---

[1] Mark Sofroniou: *C, FORTRAN77, Maple and TeX Code Generation Package*; `http://library.wolfram.com/infocenter/MathSource/60/`

trace of covariance matrix (cf. eq. 4.13). The figure 7.3 shows the resulting plots using rms with a two and three camera setup.



(a) (b)

Figure 7.3: Rms error for tracking a single feature point in a two-camera (a) and three-camera (a) setup

Maximum The square root of the maximal eigenvalue of the covariance matrix corresponds to the maximal standard deviation along the principal axes. By using the maximum of the three standard deviations only the greatest uncertainty is visualized, so this mapping is usefully if an application demands an upper bound for uncertainty. The plots resulting from this mapping (cf. figure 7.4) are showing smaller uncertainties as with other mappings because only one of three uncertainties is used for visualization.

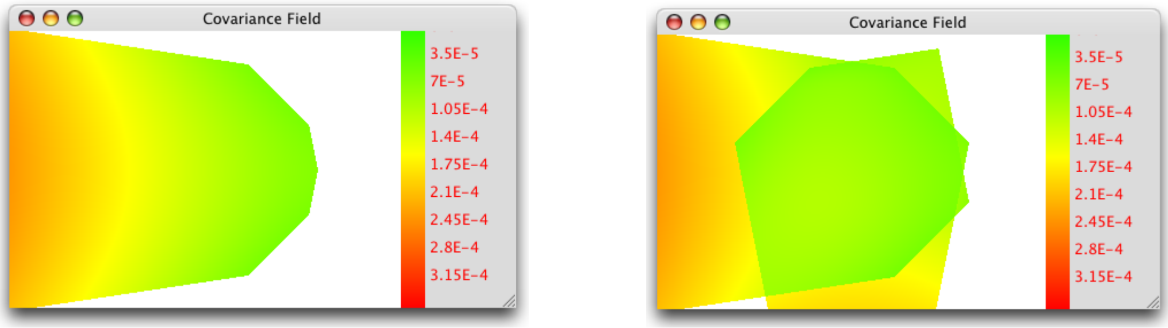
(a) (b)

Figure 7.4: The greatest standard deviation along the principal axes is used to visualize the error for tracking a single feature point in a two-camera (a) and three-camera (a) setup

A simplification of this mapping would be to use just the biggest value of the covariance matrix and not of the eigenvalues. The differences between these two visualizations are in general not visible.

Volume Another idea for mapping the covariance matrix onto a single scalar number is using the volume of the ellipsoid which corresponds to the covariance matrix. The volume of an ellipsoid is defined as $V = \frac{4}{3}\pi abc$, where a, b, c are the radii of the ellipsoid. As the radii of the covariance ellipsoid are the square roots of the eigenvalues of Σ the volume of the ellipsoid can be written with the eigenvalues λ_i as

$$V_\Sigma = \frac{4}{3}\pi\sqrt{\lambda_1\lambda_2\lambda_3} \tag{7.1}$$



(a)

(b)

Figure 7.5: The volume of the covariance ellipsoid is used to visualize the error for tracking a single feature point in a two-camera (a) and three-camera (a) setup

Instead of calculating the eigenvalues of the matrix, the volume can be calculated with help of the matrix determinant. The determinant of a covariance matrix is only determined by the eigenvalues of the matrix Σ:

$$\sqrt{|\Sigma|} = \sqrt{|RDR^t|} = \sqrt{|R||D||R^t|} = \sqrt{|D|} = \sqrt{\lambda_1\lambda_2\lambda_3} \tag{7.2}$$

It is easy to see that the volume of the ellipsoid can then calculated by

$$V = \frac{4}{3}\pi\sqrt{|\Sigma|} \quad . \tag{7.3}$$

But since the volume of the ellipsoid is a cubic measure, the cube root of the volume has to be used to get comparable results as eg. with rms. The rms and the cube root of the volume are only differing in a constant factor if the error is isotropically distributed. With an isotropic error the corresponding uncertainty ellipsoid is a sphere with radius r, then the rms is r and the cube root of the volume is $(\frac{4}{3}\pi)^{\frac{1}{3}}r$. So the cube root of the volume and the rms are only differing in a constant factor.

The mapping for using the volume to visualize the tracking accuracy can be written as

$$\mathrm{f}(\Sigma) = (\frac{4}{3}\pi\sqrt{|\Sigma|})^{\frac{1}{3}} \tag{7.4}$$

and the resulting images for two and three cameras can be seen in figure 7.5.

The speed of the computation was analyzed to evaluate the usability of the prediction algorithms for real-time applications. For a two-camera setup, the estimation of the covariance of a single feature needs about 250 multiplications and 125 additions, running in about 1μs on a standard PC; computation for a three-camera setup needs about twice as long, for a four camera setup about four times as long. So the algorithms are fast enough to be used for real-time applications.

7.3 Pose Uncertainty Visualizer

The *Pose Error Visualizer* can be used to analyze the predicted accuracy of an optical marker targets consisting of several feature points.

The developed application does not attempt to optimize the target geometry in respect of accuracy automatically. It gives the user an instrument to view the accuracy while designing or changing an existing target. In contrast to this approach, Davis et al [16] developed a technique for optimizing a target with respect to the visibility of the features.

For assessing an existing marker target the geometrical model retrieved from the tracking system can be imported into the tool. With a given 3D tracking uncertainty covariance for each feature the 6D pose uncertainty of the whole target will be computed with the formulas from Hoff et al. [24] which are explained in detail in section 6. Based on the accuracy detecting the feature points the uncertainty for position and rotation is calculated in the centroid of the target. The user can define a point of interest which is then used for the prediction and visualization. The accuracy prediction at a point of interest is done by applying the forward propagation rules presented in section 6.3.

The visualization is implemented as shown in section 6.6; so the positional uncertainty is shown as a semi-see-through ellipsoid placed at origin or the point of interest. The rotational uncertainty is visualized with three cones along three perpendicular axes.

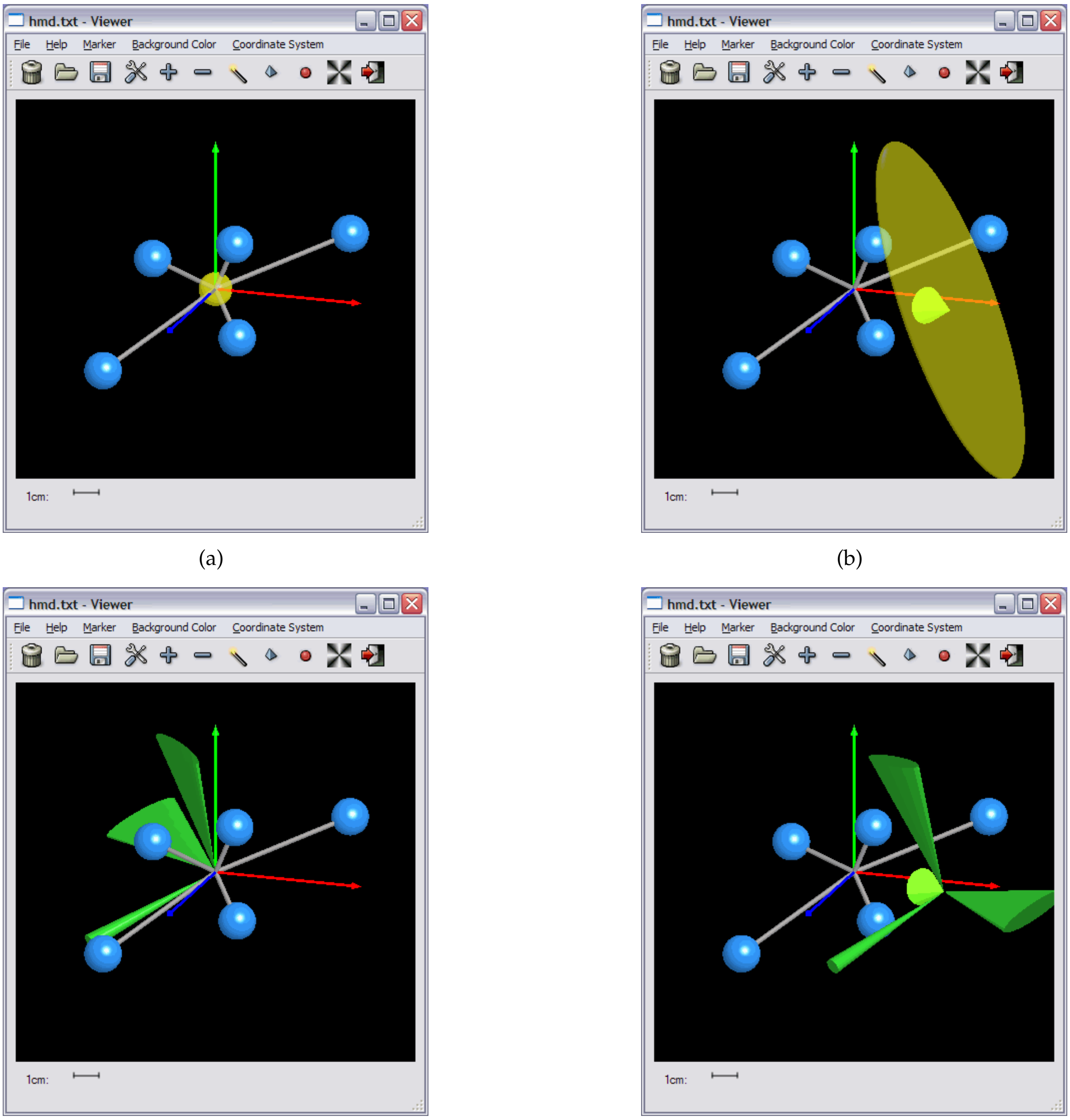
It is also possible to change the geometry of the target and to view the resulting uncertainty at real-time which makes it possible for the user to design targets which meet his demands of positional and rotational uncertainty. For example the target should be on one hand as small as possible but on the other hand the rotational uncertainty is increasing with the decreasing size of the target [58].

The positional accuracy in the centroid is minimal and it is only depending on the amount of point features used [58]. When using different locations for the accuracy prediction, the positional accuracy is also affected by the rotational uncertainty which is depending only on the spatial distribution of the point features [58]. This effect can be seen in the comparison between figure 7.6(a) which shows the positional uncertainty ellipsoid in the centroid and figure 7.6(b) which shows the positional uncertainty ellipsoid at a point of interest. In the first screenshot the positional uncertainty ellipsoid has the shape of a sphere because an isotropic distribution of position errors was used; in the second screenshot the geometry of the positional ellipsoid has changed due to the rotational errors. Figure 7.6(c) and 7.6(d) show the rotational uncertainty in the centroid and in a point of interest. The geometry of the cones has not changed as the rotational accuracy is not depending on the position.

The tool was developed using the *QT4* library[2] which allows portability to several plat-

---

[2] Trolltech AS, `http://www.trolltech.com/`



(a) (b)

(c) (d)

Figure 7.6: Screenshots of Pose Uncertainty Visualizer; the figures (a) and (a) show the positional uncertainty of the target in the centroid and in a point of interest; the figures (c) and (d) show the rotational uncertainty of the target in the centroid and in a point of interest

forms eg. *Linux*, *Windows* and *Mac OS X*. For the visualization OpenGL was chosen which provides also portability for many platforms.

8 Future Work and Conclusions

This last chapter contains an outlook to potential future applications and research directions as well as a short conclusion of the thesis.

8.1 Future Work

During work on this thesis many interesting ideas occurred how to improve the presented methods for predicting and visualizing the accuracy of optical tracking system which could not be realized. Some of these ideas will be presented here.

8.1.1 Monocular Accuracy Prediction

The model for predicting the pose estimation accuracy with a monocular tracking system could be improved to consider the third euler angle and a spatial position which would lead to a prediction of the full 6DoF pose uncertainty.

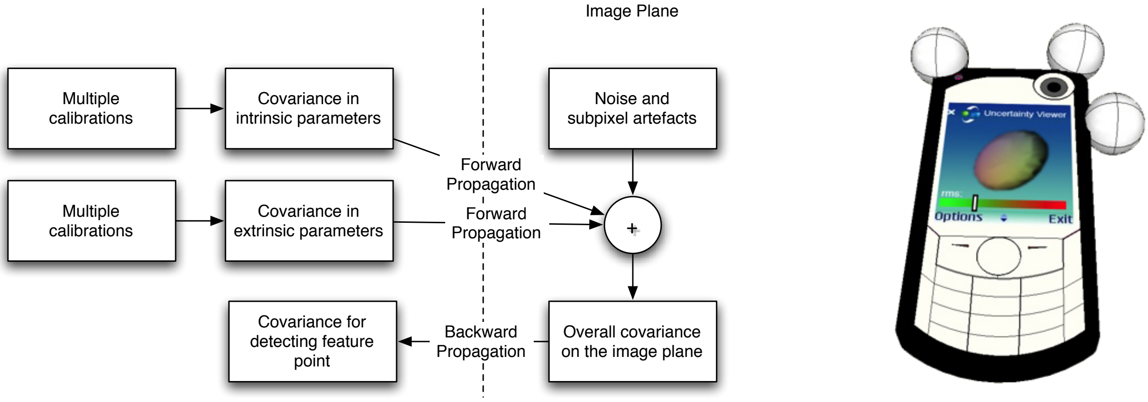
This could be achieved by using the rotation matrix from formula (3.1) and a spatial position in the extrinsic camera matrix and by adjusting the Jacobian to consider these six variables of the pose. A model for this was developed but not tested.

8.1.2 Volume rendering

The tool for visualizing the positional accuracy of a n-ocular optical tracking setup is only capable to show the uncertainty on the hyperplane of the tracking volume. Since the plane can be varied in position and orientation the volume can be assessed step by step which is not intuitive for the user assessing the system. By using volume rendering for the visualization of accuracy the working volume can be overseen easily. Cerfontaine et al. [12] and Allen et al. [2] used this approach for visualization of the working volume.

8.1.3 Field Covariance Measure System

Another approach for testing the accuracy tracking a single feature point within the working volume of a n-ocular tracking system would be the use of a cellular phone or another mobile device. A target is rigidly mounted to the device which allows to visualize the accuracy in front of the device as an ellipsoid and with the rms. The user of such an application could assess the accuracy within that regions he wants to use. As the computation abilities of mobile devices are limited, the uncertainty can be calculated on a workstation and sent to the device via arbitrary wireless technology. Some researchers already use mobile phone for augmented reality application such as [39]. Figure 8.1(b) shows a mock-up of a cellular phone visualizing a covariance ellipsoid and the corresponding rms.



(a) (b)

Figure 8.1: In (a) a suggestion for considering uncertainties in intrinsic and extrinsic camera parameters can be seen; Figure (b) shows a mock-up of an application for a mobile device visualizing the spatial tracking uncertainty in the working volume

8.1.4 Uncertainty in the Target Geometry

For the model which predicts the pose accuracy of a marker it is assumed that the geometrical distribution of the markers is known exactly. However it is impossible to get an exact model of the target, even with a perfect calibrated tracking system the features will be measured erroneous due to noise. The noise can be reduced by taking multiple measurements of the marker positions and assuming the mean of the measurement for the positions, but some error will still be present. So a way has to be found to predict the accuracy of the target with respect to errors in the target model. Davis et al. [15] and other researchers have developed models considering the uncertainty in the target geometry but these can not be integrated easily into the prediction framework presented in this theses.

8.1.5 Real-time Estimation of Accuracy

The models presented for the pose accuracy of marker targets and for the accuracy estimating feature points can be used to provide estimates for the accuracy of a specific measurement at runtime. The necessary data such as poses of cameras and the geometrical models of the used target have to be obtained somehow from the used tracking system. Such an estimate of tracking accuracy at runtime is useful for almost any augmented reality application especially for those which are safety-critical.

Instead of estimating the accuracy based on informations obtained from the tracking system it could be integrated into the tracking systems itself. This is especially interesting since such a system could include all the parameters that went into the calculation of the pose into the prediction for every frame. For instance the tracking system knows which of the feature points are used for the pose determination of a target.

8.1.6 Consider Uncertainty of Extrinsic and Intrinsic Parameters

If a n-ocular tracking system is used the poses of the used cameras have to be determined for a correct triangulation of the point features. Even with diligent calibration, the pose determination will be erroneous because the measurements used for the calibration are affected by noise. This error caused by noise can be reduced by using multiple measurements for the calibration but the result will never be exact. The same problem occurs when estimating the intrinsic camera parameters and the parameters correcting the lens distortion.

When estimating those parameters several times, the biased error in these parameters can be interpreted as multivariate random variables. These random variables are defined by a mean and covariance matrix where the mean represents the most accurate calibration. The forward propagation formulas can then be used to propagate the covariance matrix of the calibration parameters to the image plane which results in a two dimensional covariance matrix. This covariance matrix can be combined with the covariance matrix estimated as in section 5.1.3. Figure 8.1(a) shows a sketch how the combination of forward and backward propagations can be used to consider errors in the extrinsic and intrinsic camera parameters.

It has to be tested if the representation of biased errors with a zero-mean normal distribution is valid.

8.1.7 Occlusion

Since for the features of marker targets often retro-reflective marker balls are used, these markers can be hidden by other markers (cf. section 4.1.1). The probability of occlusions is depending on the target geometry, the size of the marker balls and the camera setup. It would be useful to determine the probability of occlusions at design time.

Davis et al. presented in [15] a *viewpoint algorithm*, which can be used with some changes to calculate the probability of occlusions. Figure 8.2 shows the principal idea behind this method. Assuming a imaginary sphere around the centroid of the target on which the cameras are placed. By taking a sufficient amount of sampling points on the sphere and testing the visibility from these sampling points, it would be possible to determine the ratio of points with occlusions to points without occlusions. This ratio could be used as a metric for the possibility of occlusions. As this model does not consider the real positions of the cameras, it cannot estimate the concrete possibilities for a specific setup, but it allows a statistical statement about the marker target in general.

Implementing a metric for occlusion in the Pose Uncertainty Visualizer would help the user to find the right balance between the size of the target and the requested accuracy.

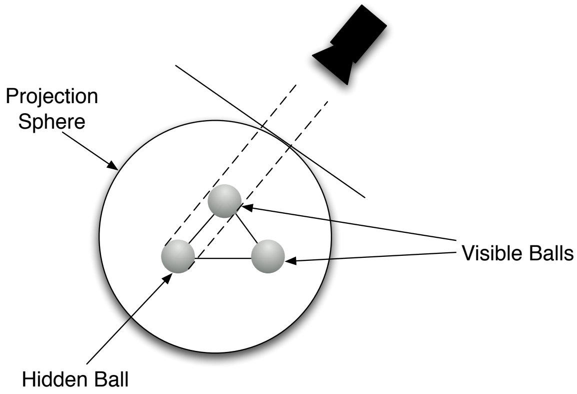


Figure 8.2: Finding occlusion of marker balls by testing multiple lines-of-sight

8.1.8 Areas or Volumes of Interest

In many applications only a specific part of the working volume is of interest. A surgeon for example wants that a lower bound for tracking accuracy is guaranteed in the surrounding area of his patient.

This can be achieved by defining a volume or area of interest. For this volume the accuracy of tracking a marker target can be calculated based on the model of the target on the camera setup. For resulting target accuracies in that region of interest statistics can be calculated – for example the mean of the rms or, most important for the surgeon, a lower bound for the accuracy.

Similar to this approach is the application introduced in section 7.2. In contrast to the resulting total target error desired in this concept, in the referred tool only the accuracy of detecting a single fiducial with multiple cameras is implemented. But the metrics for visualizing the positional covariance matrix can also be used for examining the region of interest statistically.

8.2 Conclusions

In this thesis the models for predicting and estimating the accuracy of optical tracking systems were derived.

Therefore a gaussian error model was introduced using multi-variate random variables to describe the errors caused by noise and it has been shown how the errors of this gaussian model can be interpreted and visualized.

Techniques for propagating the gaussian errors were shown and used for a set of theoretical models for accuracy prediction of tracking systems.

One model can predict the pose accuracy of tracking a target with monocular system based on the accuracy detecting a single feature on the image plane and on the geometry of the used target. For n-ocular optical feature tracking systems a model was developed to predict the expected accuracy tracking a single feature, based on the accuracy on the image plane and the geometrical distribution of the cameras. For predicting the accuracy of a marker targets a method was presented which is based on the accuracy how accurate the system can track the position of a single feature in 3D and on the geometrical layout of the target.

Based on these models two applications were developed. One can calculate and visualize the accuracy of the detecting spatial position of a single feature with a n-ocular tracking system. The other application can calculate the accuracy of a marker target the usage with a n-ocular tracking system.

Experiments and simulations have been used to prove the feasibility of the models. The Experiments were also used to estimate the parameters for real setups.

The presented model for predicting and estimating the accuracy of optical tracking systems can be used in future applications for instance providing real-time predictions of the expected accuracies.

Bibliography

[1] D. F. ABAWI, J. BIENWALD, and R. DÖRNER, *Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit*, in Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 04), 2004, pp. 260–261.

[2] B. D. ALLEN and 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, New York, NY, USA, 2005, ACM Press, pp. 201–210.

[3] M. APPEL and N. NAVAB, *Registration of technical drawings and calibrated images for industrial augmented reality*, Machine Vision and Applications, 13 (2002), pp. 111–118.

[4] K. B. ATKINSON, *Close Range Photogrammetry and Machine Vision*, Whittles Publishing, 1996.

[5] R. AZUMA, *A Survey of Augmented Reality*, Presence, 6 (1997), pp. 355–385.

[6] R. AZUMA, Y. BAILLOT, R. BEHRINGER, S. FEINER, S. JULIER, and B. MACINTYRE, *Recent advances in augmented reality*, Computer Graphics and Applications, IEEE, 21 (2001), pp. 34–47.

[7] R. AZUMA and G. BISHOP, *Improving static and dynamic registration in an optical see-through HMD*, in SIGGRAPH, 1994, pp. 197–204.

[8] D. K. BHATNAGAR, *Position trackers for Head Mounted Display systems: A survey*, tech. rep., Department of Computer Science, University of North Carolina - Chapel Hill, 1993.

[9] G. BISHOP, G. WELCH, and B. D. ALLEN, *Tracking: Beyond 15 Minutes of Thought*, SIGGRAPH Course Pack, (2001).

[10] D. C. BROWN, *Close-range camera calibration*, Photogrammetric Engineering, 37 (1971), pp. 855–866.

[11] T. CAUDELL and D. MIZELL, *Augmented reality: an application of heads-up display technology tomanual manufacturing processes*, in Proceedings of Hawaii International Conference on System Science, vol. 2, 1992.

[12] P. A. CERFONTAINE, M. SCHIRSKI, D. BÜNDGENS, and T. KUHLEN, *Automatic Multi-Camera Setup Optimization for Optical Tracking*, in Proceedings of IEEE Virtual Reality, 2006.

[13] X. CHEN and J. DAVIS, *Camera Placement Considering Occlusion for Robust Motion Capture*, Tech. Rep. CS-TR-2000-07, Stanford University Computer Science, 2000.

[14] J. J. CRAIG, *Introduction to Robotics Machanics & Control*, Addison-Wesley, 1986.

[15] L. DAVIS, E. CLARKSON, and J. P. ROLLAND, *Predicting Accuracy in Pose Estimation for Marker-based Tracking*, Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03), (2003), p. 28.

[16] L. DAVIS, F. G. HAMZA-LUP, and J. P. ROLLAND, *A Method for Designing Marker-Based Tracking Probes*, Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04), (2004), pp. 120–129.

[17] W. FELLER, *An Introduction to Probability Theory and Its Applications*, vol. 1,2, Wiley, 1968.

[18] G. FISCHER, *Lineare Algebra*, Friedr. Viewer & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1995.

[19] J. M. FITZPATRICK, J. B. WEST, and C. R. M. JR., *Predicting error in rigid-body, point-based registration.*, IEEE Transactions on Medical Imaging, 17 (1998), pp. 694–702.

[20] W. R. HAMILTON, *On Quaternions*, in Proceedings of the Royal Irish Academy, D. R. Wilkins, ed., vol. 3, 1847, pp. 1–16.

[21] R. M. HARALICK, H. JOO, C. LEE, X. ZHUANG, V. G. VAIDYA, and M. B. KIM, *Pose Estimation from Corresponding Point Data*, IEEE Transactions on Systems, Man and Cybernetics, 19 (1989), pp. 1426–1446.

[22] R. I. HARTLEY and P. STURM, *Triangulation*, Lecture Notes in Computer Science, 970 (1995), pp. 190–197.

[23] R. I. HARTLEY and A. ZISSERMAN, *Multiple View Geometry in Computer Vision*, Cambridge University Press, June 2000.

[24] W. HOFF and T. VINCENT, *Analysis of Head Pose Accuracy in Augmented Reality*, in IEEE Transactions on Visualization and Computer Graphics, vol. 6(4), IEEE Computer Society, 2000, pp. 319–334.

[25] W. A. HOFF, *Fusion of Data from Head-Mounted and Fixed Sensors*, in First IEEE International Workshop on Augmented Reality (IWAR '98), 1998.

[26] B. HOFMANN-WELLENHOF, H. LICHTENEGGER, and J. COLLINS, *GPS: theory and practice*, Springer, 1994.

[27] B. K. P. HORN, *Closed-Form Solution of Absolute Orientation Using Unit Quaternions*, Journal of the Optical Society of America, 4 (1987).

[28] R. JACOB, *Eye tracking in advanced interface design*, Virtual Environments and Advanced Interface Design, (1995), pp. 258–288.

[29] B. JÄHNE, *Digitale Bildverarbeitung*, Springer, Berlin, 2002.

[30] K. KANATANI, *Unbiased estimation and statistical analysis of 3-D rigid motion from two views*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15 (1993), pp. 37–50.

[31] H. KATO and M. BILLINGHURST, *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System*, in 2nd IEEE and ACM International Workshop on Augmented Reality, 1999, p. 85.

[32] K. R. KOCH, *Parameterschätzung und Hypothesentests in linearen Modellen*, Dümmler, Bonn, 1980.

[33] C. LANCZOS, *Applied Analysis*, Prentice-Hall Mathematices Series, Prentice Hall, 1956.

[34] P. LANG, A. KUSEJ, A. PINZ, and G. BRASSEUR, *Inertial tracking for mobile augmented reality*, in Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, vol. 2, 2002, pp. 1583– 1587.

[35] V. LEPETIT and P. FUA, *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey*, Foundations and Trends in Computer Graphics and Vision, 1 (2005), pp. 1–89.

[36] T. LUHMANN, *Photogrammetrische Verfahren in der industriellen Messtechnik*, in Publikationen der Deutsche Gesellschaft für Photogrammetrie und Fernerkundung eV (DGPF), vol. 9, 2001.

[37] T. LUHMANN, *Nahbereichsphotogrammetrie. Grundlagen, Methoden und Anwendungen*, Wichmann, Heidelberg, Germany, 2003.

[38] P. MALBEZIN, W. PIEKARSKI, and B. THOMAS, *Measuring ARToolkit Accuracy in Long Distance Tracking Experiments*, in 1st International ARToolkit Workshop, Darmstadt, Germany, October 2002.

[39] M. MOHRING, C. LESSIG, and O. BIMBER, *Video See-Through AR on Consumer Cell-Phones*, in Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 04), vol. 00, Los Alamitos, CA, USA, 2004, IEEE Computer Society, pp. 252–253.

[40] H. NAJAFI, N. NAVAB, and G. KLINKER, *Automated Initialization for Marker-Less Tracking: A Sensor Fusion Approach*, in ISMAR, 2004, pp. 79–88.

[41] C. OWEN, F. XIAO, and P. MIDDLIN, *What is the best fiducial?*, in Augmented Reality Toolkit, The First IEEE International Workshop, 2002, p. 8 pp.

[42] D. PUSTKA, *Handling Error in Ubiquitous Tracking Setups*, Master's thesis, Technische Universität München, 2004.

[43] J. REKIMOTO, *Matrix: a realtime object identification and registration method for augmented reality*, in 3rd Asia Pacific Computer Human Interaction, 1998, pp. 63–68.

[44] M. RIBO, *State of the art report on optical tracking*, Vienna Univ. Technol., Vienna, Austria, Tech. Rep, 25 (2001), p. 2001.

[45] F. SAUER, A. KHAMENE, and 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, London, UK, 2002, Springer-Verlag, pp. 116–124.

[46] M. SCHNEEBERGER, *Implementierung und Test eines videobasierten 3D-Messsystems zur Patientenpositionierung in der Praezisionsstrahlentherapie*, Master's thesis, Ruprecht-Karls-Universität Heidelberg, 2000.

[47] H. R. SCHWARZ, *Numerische Mathematik*, B. G. Teubner, Stuttgart, 1993.

[48] M. SHORTIS, T. CLARKE, and T. SHORT, *A comparison of some techniques for the subpixel location of discrete target images*, SPIE Videometrics III, 2350 (1994), pp. 239–250.

[49] C. TRÜBSWETTER, *Analyzing and Monitoring Tracking Accuracy of an A.R.T. System*, 2003.

[50] R. Y. 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, RA-3 (1987), pp. 323–344.

[51] H. L. VAN TREES, *Detection, Estimation and Modulation Theory*, Wiley, New York, 1968.

[52] *Optical 3D measuring system - imaging systems with point-by-point probing.* VDI/VDE guideline 2634/1, 2002.

[53] S. VOGT, A. KHAMENE, F. SAUER, and 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), 2002, pp. 127–136.

[54] F. WANG, *A simple and analytical procedure for calibrating extrinsic camera parameters*, Robotics and Automation, IEEE Transactions on, 20 (2004), pp. 121–124.

[55] F. WANG, *An Efficient Coordinate Frame Calibration Method for 3-D Measurement by Multiple Camera Systems*, Systems, Man and Cybernetics, Part C, IEEE Transactions on, 35 (2005), pp. 453–464.

[56] C. WARE and R. BALAKRISHNAN, *Reaching for Objects in VR Displays: Lag and Frame Rate*, ACM Transactions on Computer-Human Interactions, 1 (1994), pp. 331–356.

[57] E. W. WEISSTEIN, *Mathworld.* http://mathworld.wolfram.com/.

[58] H. J. . WOLTRING, R. HUISKES, A. DE LANGE, and F. E. VELDPAUS, *Finite centroid and helical axis estimation from noisy landmark measurements in the study of human joint kinematics*, in Journal of Biomechanics, vol. 18(5), 1985, pp. 379–389.

[59] K. WONG and W. HO, *Close-range mapping with a solid state camera*, Photogrammetric Engineering and Remote Sensing, 52 (1986), pp. 67–74.

[60] Y. YANAGIDA, S. SAITO, S. YANO, T. MAEDA, and S. TACHI, *A Head-Tracked, Live-Video-Based Telexistence System Using a Fixed Screen*, Proceedings of the 11th International Conference on Artificial Reality and Telexistence (ICAT 2001), (2001), pp. 42–47.

[61] S. YOU, U. NEUMANN, and R. T. AZUMA, *Hybrid Inertial and Vision Tracking for Augmented Reality Registration*, in IEEE Virtual Reality Conference 1999 (VR'99), vol. 00, Los Alamitos, CA, USA, 1999, IEEE Computer Society, p. 260.

[62] X. ZHANG, S. FRONZ, and 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'02), 2002, pp. 97 – 106.

[63] X. ZHANG, Y. GENC, and N. NAVAB, *Taking AR into Large Scale Industrial Environments: Navigation and Information Access with Mobile Computers*, in IEEE and ACM International Symposium on Augmented Reality (ISAR'01), vol. 00, Los Alamitos, CA, USA, 2001, IEEE Computer Society, p. 179.

[64] W. ZHENGYAN and A. JEPSON, *A new closed-form solution for absolute orientation*, in Computer Vision and Pattern Recognition, 1994, pp. 129–134.