Predicting and Estimating the Accuracy of n-occular Optical Tracking Systems

Martin Bauer* Michael Schlegel† Daniel Pustka Nassir Navab Gudrun Klinker

Technische Universität München, Fakultät für Informatik
Boltzmannstraße 3, Garching bei München, Germany

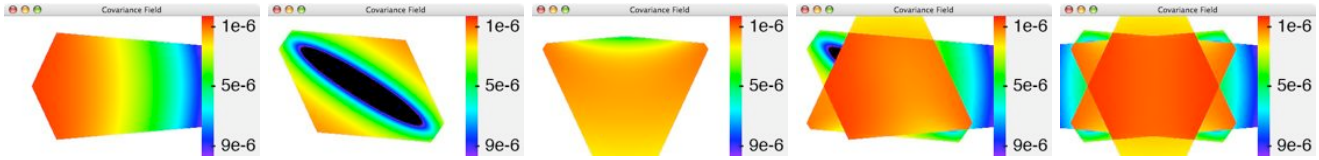


Figure 1: Screenshots for predicted position estimation error (trace norm of covariance matrix) for a single feature in multi-camera arrangements as presented in section 4.4. Images show (a) two cameras in upper and lower left corner, (b) two cameras in upper left and lower right corner, (c) two cameras in upper left and right corner, (d) three cameras in upper left, lower left and right corner and (e) four cameras, one in each corner. Red color means high accuracy and dark blue means low accuracy.

Abstract

Marker-based optical tracking systems are widely used in augmented reality, medical navigation and industrial applications. We propose a model for the prediction of the target registration error (TRE) in these kinds of tracking systems by estimating the fiducial location error (FLE) from two-dimensional errors on the image plane and propagating that error to a given point of interest. We have designed a set of experiments in order to estimate the actual parameters of the model for any given tracking system. We present the results of a study which we used to demonstrate the effect of different sources of error. The method is applied to real applications to show the usefulness for any kind of augmented reality system. We also present a set of tools that can be used to visualize the accuracy at design time.

Keywords: Optical Tracking, Accuracy Estimation, Error Propagation, Error Prediction, Target Registration Error

1 Introduction

Estimating the pose of an object in augmented reality systems always includes errors. While the general goal of an application designer should be to make the errors as small as possible, sometimes it is necessary to know how large the error of a tracking system actually is.

Experimental evaluations of tracking accuracy exist for a variety of different tracking systems but the result is usually an estimation of the expected RMS accuracy inside a specified working volume of a particular tracking setup, rather than an error estimate for each single measurement. In our work we consider n-ocular optical tracking systems using tracking targets that consist of several feature points (fiducials) with known locations. Two example targets are shown in figure 2.

*e-mail: martin.bauer@in.tum.de
†e-mail: michael.schlegel@in.tum.de

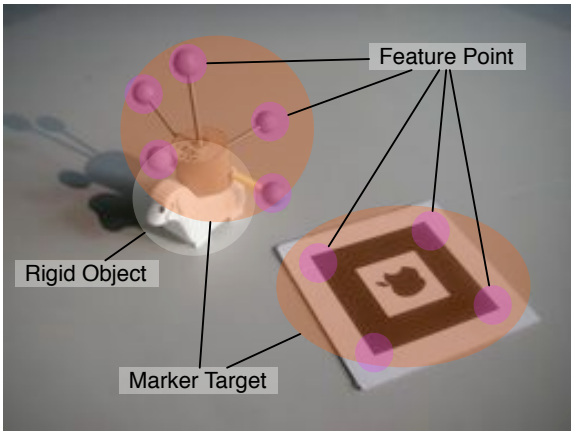


Figure 2: Marker targets consisting of several single feature points

We predict the overall *target registration error* (TRE) by propagating errors on the image plane (IPE) through the different steps in the tracking process, which have different influences on the TRE. Therefore, we need to carefully model each step and the way errors are propagated. In particular, we are looking at the following kinds of errors:

Image Plane Error (IPE) The tracking algorithm detects the point features in the image plane. The accuracy of this feature detection is limited by factors such as the image noise or the algorithm used and can be well approximated by a two-dimensional Gaussian error distribution, as we show later. Mistakes made in this step are propagated and amplified in the following steps and accumulate in the TRE. Therefore, we can consider the IPE as the source of all the other errors we are dealing with.

Fiducial Location Error (FLE) After the 2D locations of feature points in the image plane of two or more cameras are known, the 3D position of these features can be computed in the world. The

way errors in the image plane propagate to the FLE is influenced by the position of the feature and the arrangement of the cameras. We will investigate this further in section 4. Allen et. al. [19] present a general framework for predicting the estimated performance of arbitrary tracking systems. Their work is similar to what we propose in section 4 and will be discussed in more detail there. Mitschke et. al. [11] showed that it is crucial to know about the general shape of the error covariances in a given camera setup at design time.

Marker Target Error (MTE) In most cases, rigid arrangements of three or more fiducials are used to construct a marker target, of which both position and orientation can be determined. In this case the FLE error distributions of the individual fiducials influence the error in the 6D pose, but also the way they are arranged. Davis et. al. [16] have presented a method to predict the accuracy of a tracking target for optimized target design. We follow in section 3.1 the method proposed by Hoff et. al [10].

Target Registration Error (TRE) In most applications, the pose of the marker target is not used directly, but instead the position of some point of interest in the target coordinate frame is determined, such as the tip of a pointing device. By propagating the MTE errors in position and orientation of the marker target to this point of interest, the TRE error relevant for the application can be estimated. Figure 3 shows an example visualization for the accuracy of a coordinate measurement tool. Fitzpatrick et. al. [8] give a simple formula to predict the target registration error (cf. section 3.1), but assume not only gaussian zero-mean errors in the 3D fiducial detection (fiducial location error, FLE), but additionally requires the error to be independent, isotropic and identical for all fiducials. We show in section 4 does not hold for most common setups.

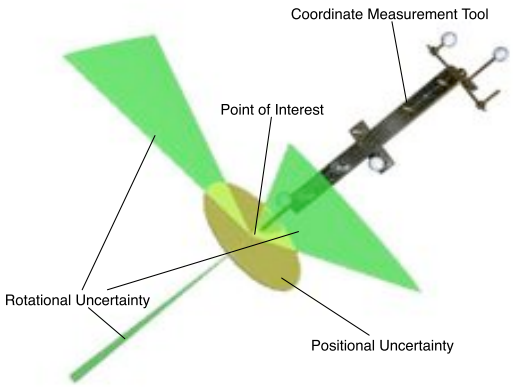


Figure 3: Visualization of predicted positional and rotational accuracy for a coordinate measurement tool

By combining the error propagations of all four steps, we can give a correct estimate of the target registration error for every measurement at runtime [17]. This is crucial for safety-critical augmented reality applications and desirable for any kind of augmented reality system. Coelho et. al. [17] have presented a way to propagate tracking error at runtime through a complete augmented reality application. Accurate runtime estimates can only be given by the tracking system itself since information is needed on how many cameras and how many feature points were used for the particular measurement.

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

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

This paper starts in a top-down approach in section 3 with a derivation of the error propagation formulas. First we show (section 3.1) how to estimate the 6D covariance of a specific marker target (marker target error, MTE) from known 3D covariances for every fiducial or feature (fiducial location error, FLE), as seen in figure 3. This MTE is further propagated to a point of interest to reveal the actual target registration error (TRE) relevant for the application. We continue (section 4) with a method to estimate the 3D FLE covariance for a single feature from the camera setup and 2D IPE detection covariances on the image plane of the cameras. Examples are seen in figure 1. After that we present in section 4.2 a set of experiments that we conducted on the one hand to validate the model and on the other hand to estimate the necessary parameters for the 2D IPE covariances for a specific camera and detection algorithm. We finish with the complete estimation of 6D covariances at points and regions of interest (section 5) from 2D covariances in the image planes and show the usefulness with a real life example in section 5.2.

2 Error Propagation Rules

The error in optical tracking systems comes from a large variety of different sources, most of them not being normally distributed. We will show later in real world experiments that modeling them as if they were Gaussian distributed is still a valid assumption for many of the errors. We will also show how to deal with the other kinds of errors.

In general it is most useful to provide Gaussian error estimates in terms of covariance matrices for each measurement [17]. In this section we shortly review the general error propagation rules for Gaussian errors that we use for the estimation of the final errors, the *forward propagation* and *backward propagation* of covariance matrices.

2.1 Forward Propagation

The *forward propagation of covariance* rule for an affine function f is defined as [9]: Let v be a random vector in $\mathbb{R}^M$ with mean $\bar{v}$ and covariance matrix Σ, and suppose that $f : \mathbb{R}^M \mapsto \mathbb{R}^N$ is an affine mapping defined by $f(v) = f(\bar{v}) + A(v - \bar{v})$. Then $f(v)$ is a random variable with mean $f(\bar{v})$ and covariance matrix Σ_f:

$$\Sigma_f = A\Sigma A^T \tag{1}$$

More specifically, the non-linear propagation using first order approximation: Let v be a random vector in $\mathbb{R}^M$ with mean $\bar{v}$ and covariance matrix Σ, and suppose that $f : \mathbb{R}^M \mapsto \mathbb{R}^N$ is differentiable and approximately linear in a neighborhood of $\bar{v}$. Then, up to a first order approximation, $f(v)$ is a random variable with mean $f(\bar{v})$ and covariance matrix Σ_f:

$$\Sigma_f = J_f \Sigma {J_f}^T \tag{2}$$

where J_f is the Jacobian matrix of f evaluated at $\bar{v}$.

We present the respective functions and their Jacobians in the following sections.

2.2 Backward Propagation

In some cases we do know the covariance of the function $f(v)$ and want to estimate the covariance of the parameters v. We could apply forward propagation on the inverse function f^{-1}, but what if the inverse function is hard to compute? Instead of computing $J_{f^{-1}}\Sigma {J_{f^{-1}}}^T$ we can show [9] that this is equal to computing

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

which is easier to achieve, in particular when we replace the matrix inversion by the *pseudoinverse*

$$\Sigma_{f^{-1}} = ({J_f}^T \Sigma^{-1} J_f)^{+} \tag{3}$$

to get a solution for the overparametrized case.

3 Propagation from Marker Balls (FLE) to Point of Interest (TRE)

We start our analysis with an estimation of the theoretical error of a tracking target consisting of several feature points. Such targets are commonly used in commercial tracking systems, where a target consists of several retroreflective marker balls [10] or planar paper-based markers as seen in figure 2. The same analysis can, with slight modifications in the measurement equations, also be applied to monocular visible light marker based tracking systems [5, 15].

3.1 Derivation of Covariance Formulas

We start with computing the 6D error covariance $\Sigma_{\vec{c}} \in \mathbb{R}^{6\times 6}$ of the marker target in the centroid $\vec{c}$ [10], assuming that we know the fiducial location error (FLE) for every feature point $\vec{p}_i$, not only as a single RMS value [8] but as an arbitrary covariance matrix $\Sigma_{\vec{p}_i} \in \mathbb{R}^{3\times 3}$. We will show later in section 4 a way how to estimate the FLE for a given camera setup.

Without loss of generality we consider a marker target defined as a set of feature point coordinates $\{\vec{q}_k\} \in \mathbb{R}^3$ in a local coordinate system with the origin in the centroid of the marker balls, and their respective counterparts $\{\vec{p}_k\} \in \mathbb{R}^3$ in the tracker coordinate system, which are additively disturbed by zero-mean Gaussian errors $\{\Delta\vec{p}_k\}$. Note that, for the propagation to work properly, the origin has to be defined at the centroid of the marker target. Otherwise, the positional error would be artificially increased due to its distance from the origin. We will take this effect into account separately when we propagate MTE errors to points of interest.

Covariance in the centroid of the target (MTE)

From the corresponding point sets we estimate the pose of the target by solving the *3D/3D Pose Estimation* problem using any kind of algorithm, for example [1]. This estimation leads to a homogeneous transformation $[R|\vec{t}]$ which maps

$$R\vec{q}_k + \vec{t} = p_k + \Delta\vec{p}_k$$

with some error $\Delta\vec{p}_k$ for every k. By combining position and orientation at the centroid in a single vector $\vec{c} = (x, y, z, \alpha, \beta, \gamma)^T$, we can treat $\vec{c}$ as a random variable that represents the MTE. Euler angles [10] are used to represent orientation here. To apply the error propagation, we use the function $f(\vec{p}, \vec{q}) = R\vec{q} + \vec{t} - \vec{p}$ and build the Jacobian

$$J_f(\vec{q}) = \left.\frac{\partial f(\vec{p}, \vec{q})}{\partial \vec{c}}\right|_{\vec{c}=\vec{0}} \tag{4}$$

with respect to the 6D pose $\vec{c}$ at the target centroid. Without loss of generality we can assume that the estimated transformation $\vec{c}$ is zero — we can transform the coordinate system such that all coordinates $\vec{p}_i$ and the respective covariances are given in the marker target coordinate system by propagating the individual covariances with $R^T \Sigma_{\vec{p}_i} R$. In this coordinate system, the mean pose is zero, but the gaussian errors have still nonzero distribution — and evaluate the Jacobian at the pose $\vec{c} = \vec{0}$,

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

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

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

Using the backward propagation formula (3) we get

$$\Sigma_{\vec{c}} = \left(M^T \begin{bmatrix} R^T \Sigma_{p_1} R & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & R^T \Sigma_{p_n} R \end{bmatrix}^{-1} M \right)^{+} \tag{5}$$

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

To visualize this error covariance in the original world coordinate system, we could again retransform the covariance matrix by computing $R\Sigma_{\vec{c}} R^T$.

Covariance at a given Point of Interest (TRE)

From this 6D MTE covariance in the centroid we can compute the 3D TRE covariance $\Sigma_{\vec{p}}$ at a point other than the centroid by applying the forward propagation formula (2).

Again we use the Jacobian J_f from equation 4 evaluated at the point of interest $\vec{p}$. The target registration error at the point of interest is then given by

$$\Sigma_{\vec{p}} = J_f \Sigma_{\vec{c}} {J_f}^T \tag{6}$$

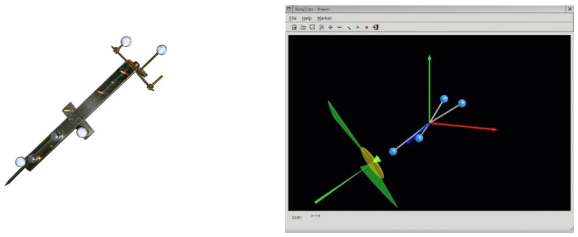
Assuming that the rotational part of the error in the centroid is independent of the positional part, it is then easy to see that the positional error at the point of interest is equal to the positional error at the centroid plus a positional error coming from the propagated rotational error [8], which is increasing proportionally to the distance from the centroid.

3.2 Interpretation of Covariances

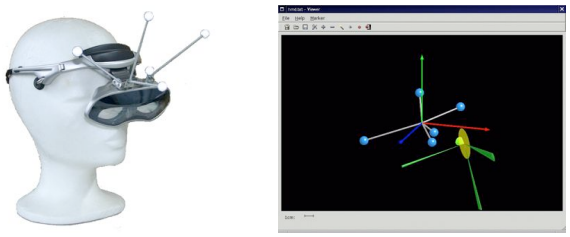
We assume for now that the tracking system is able to detect each marker ball with a certain accuracy expressed as a covariance matrix in three dimensions.

Although the model allows arbitrary covariances for each marker ball, for visualization of MTEs only, we additionally assume that this covariance is independent of the location in the tracking volume and even isotropic, i.e. the same variance in each direction. While common in tracking error analysis [8], in our work this assumption is only used for visualization reasons in the offline analysis of marker target geometries; for the further analysis of overall accuracy we allow arbitrary covariances.

We will show in section 4 that this assumption is acceptable under certain circumstances. But also for setups where this is not the case this analysis can be used to estimate the inherent error characteristics of a specific marker layout [16]. This is especially interesting since we are usually not interested in the pose of the marker target itself but rather in the pose of an object rigidly connected to that marker. Note also that this kind of geometric analysis of constellation is used in other areas as well, for example to estimate the accuracy of GPS-based measurements [7].



Coordinate measurement tool



Head-mounted display

Figure 4: Pose Error Visualization Tool; blue balls denote the target geometry, yellow ellipsoid the positional uncertainty at a point of interest and green cones the rotational uncertainty.

While it is possible to use this kind of error estimation to automatically design *optimal* marker targets [18], our main goal is the analysis of the errors in existing setups. We have implemented a tool that can be used to load the description of a marker target and then visually explore the error characteristics of this target at a user-defined point of interest. The tool can furthermore be used to evaluate the effect of changes in the marker geometry on the resulting error. For the visualization (cf. figure 4) we show the positional TRE uncertainty as a yellow ellipsoid around the point of interest at a user-definable confidence level. For the rotational uncertainty we use the same error propagation as in eq. 6 but propagate only the rotational part of the error in the centroid along three orthogonal axes. Due to singularities this results in three flat discs that we use to draw cones from the center to these discs [10].

Figure 4 shows two different marker targets and the according visualization of the error. For the coordinate measurement tool it is possible to see that the rotational accuracy along the pointing axis is quite good while rotation around that axis does not give high accuracy. For the HMD the tool reveals that there is significantly less accuracy in the pitch direction of the users head than in the two other directions.

3.3 Limitations

The presented model assumes that the geometry of the target is exactly known, although in real world setups the geometry is actually calibrated and might be affected by some error. We can however overcome this limitation by diligent calibration.

Dynamic Errors

Dynamic errors are caused by end-to-end system delays when the tracked object moves [3, 4]. In our work we ignore dynamic errors, although we are aware that these errors can be more significant than the static errors for some setups. We are only interested in the accuracy of the actual measurement. This measurement covariance could be used together with an appropriate motion model [19] to provide error estimates even between measurements. 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 [13].

Recognizing vs. Tracking

In our model we try to stay independent of the actual implementation of the system itself. In particular, we assume that a feature point can always be detected uniquely. We do not consider that some marker targets would not be uniquely detectable due to symmetries, visibility [12, 20], or other properties the specific tracking algorithms relies on. The detection rate can be increased by better design — we predict only the accuracy once a feature is correctly identified.

4 Accuracy of a Single Feature Point (FLE) in an n-Ocular System

In the previous section we have assumed that the fiducial location error (FLE) is given as a covariance for each feature point. Now we dive a little bit deeper into the details of an n-ocular system and explain how the FLE can be computed, given the camera setup and the image plane error of the feature point for each camera. Allen et.al. [19] provide a general framework for estimating the accuracy of tracking systems. Their work is based on rather similar concepts, namely error propagation through linearisations of the measurement functions. They include a motion model to cover also movement and different frame rates. In contrast to that we want to provide an estimate of a particular measurement at the time of the measurement.

4.1 Derivation of Covariance Formulas

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

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

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

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.

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

nonlinear camera equations p:

$$p: \quad \begin{matrix} \vec{u_1} \\ \vdots \\ \vec{u_n} \end{matrix} = \begin{matrix} \rho_1 \mathbf{K_1} \mathbf{T_1} \vec{x} \\ \vdots \\ \rho_n \mathbf{K_n} \mathbf{T_n} \vec{x} \end{matrix}$$

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

In order to compute the FLE, we build the Jacobian $J_p = \frac{\delta p}{\delta \vec{x}}$ and apply the backward propagation formula (3).

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

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

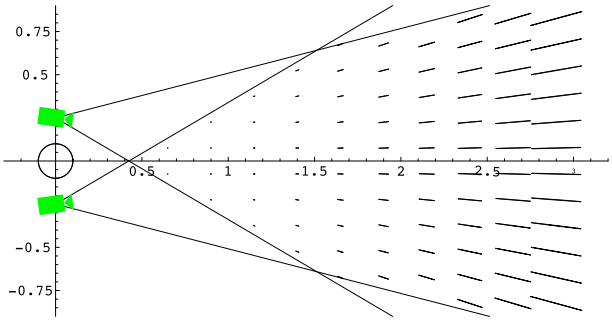


Figure 5: Error Covariances for tracking a single marker ball in a two-camera setup (Magnified by factor 100 for visualization)

Figure 5 shows a visualization of the error covariances in a two-camera setup. For this image we assumed an isotropic covariance with a standard deviation of of $\frac{1}{115}$ pixel on the image plane in each direction; the cameras have a 50 cm baseline and a focal length of 3.5 mm. We will show in section 4.2 how we have estimated this covariance. For real world setups like the one presented here, we additionally consider the field of view of the cameras and use only the cameras that are able to see the point for the error estimation [20].

4.2 Experimental Estimation of Errors

In this section we present a series of experiments that we conducted to validate our model and to experimentally get an estimate of the IPE covariance on the image plane of a specific optical tracking system.

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

Image Noise

To realistically estimate the error from image noise, we placed retroreflective marker balls as features in a regular grid on a table in our tracking volume. Both the cameras and the markers on the table were fixed throughout the experiment. We captured the measured locations of the features in space for several minutes at 60 Hz resulting in a total of 71553 sample points for each feature. We used

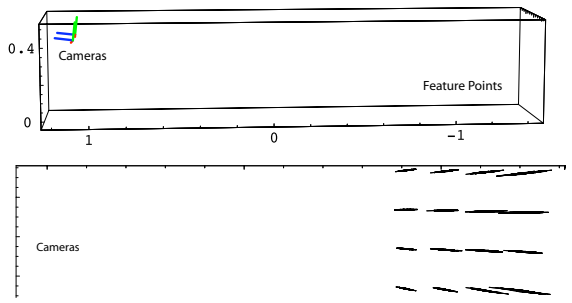


Figure 6: Error in position estimation for a regular grid (side & top view of the setup; error for visualization exaggerated by a factor 50)

a rather small baseline of about 20 cm to emphasize the effects in the images, see figure 6.

The results of this experiment are shown in figure 7, where the measured 3D locations of a single feature point are plotted as black dots. We also calculated the covariance for each single feature point from this dataset, which is displayed as an ellipse at a 75% confidence level around the centroid. The particular regular pattern in the 3D reconstruction of the point results from discretization in the camera coordinate system. However, the general shape of the measurements in space is in fact approximated by the covariance ellipsoid.

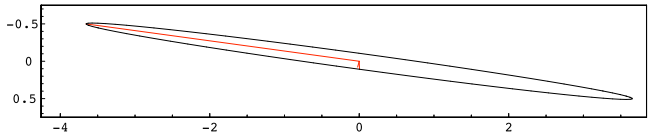


Figure 7: Error in position estimation together with estimated covariance for a single marker from figure 6

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

To estimate the error in the image plane, we evaluated our model with symbolic parameters for the covariances on the image plane and used Newton's method to fit the parameters to the measurements. Figure 8 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 leftmost point and then applied to all other points. The predicted values fit the measured values quite well. In our experiment, we estimated a standard deviation of $\frac{1}{115}$ pixel on the image plane for the detection of the center of the features. We use this estimation for the error estimates in the examples in section 5.2.

Image Noise and Subpixel Algorithm Noise

To estimate the accuracy of optical coordinate measurement systems, VDI/VDE 2634/1 [14] recommends measuring differences between single features in several directions. We extended this measurement with a rotating two-ball target (cf. figure 9) for which we measured the distance between the balls.

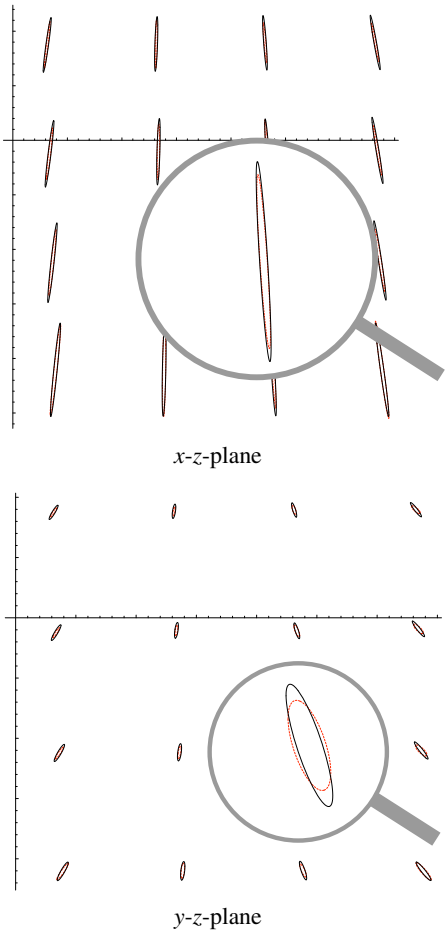


Figure 8: Measured errors (red dotted line) vs. predicted errors (black line)

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

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

which we use 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

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

We compare the variance σ_d with our measurements, as shown in figure 10. The horizontal axis shows the angle of the two balls and the vertical axis the respective measured distance.

In the errors we first note a large sinusoidal error in the angular data. This error comes from a wrong scaling of the three room axes during the calibration of the system. This systematic error from the calibration process provided by the manufacturer of the tracking system needs to get eliminated independently, as it cannot well be modeled as Gaussian noise. For our analysis we have removed that error manually from the data by assuming independent scaling on the three axes.

The remaining error consists of a random part coming from system noise as analyzed above, and an additional irregular error coming from subpixel effects in the 2D detection algorithms. Although

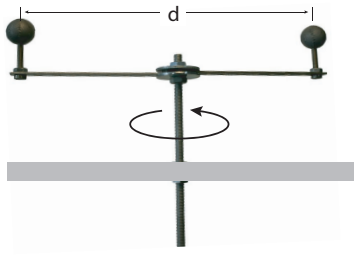


Figure 9: Two rotating balls, rigidly connected

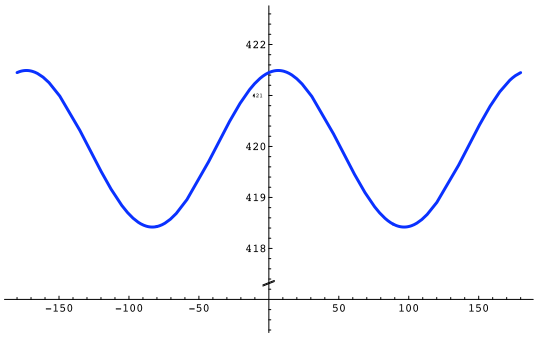


Figure 10: Error in length estimation for a rotating two-ball target, blue line shows assumed room calibration error

these errors are in fact systematic, we assume for now that we can approximate them with a Gaussian distribution and model them as a zero-mean noise in the image plane. This is justified by the relatively high frequency of the artifacts.

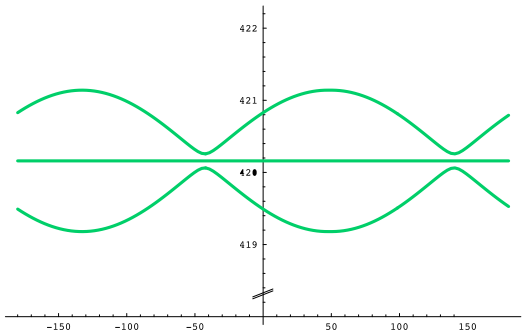


Figure 11: Measured errors corrected for room scaling (blue dots) vs. predicted error standard deviation (green line)

Figure 11 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.

4.3 Higher Order Camera Models

In our model we consider the camera system as pinhole cameras. The usage of more advanced camera calibration [2, 5] reduces the error from distortions in the image plane. We assume that the covariance on the image plane is not affected too much by the distor-

tion correction. The validity of this assumption needs to be analyzed further [6], though.

4.4 Visualization of Covariances

We have built a tool that can be used to explore the predicted errors for a general setup in three dimensions. Figure 12 shows a screen shot for a 3 camera setup similar to the one we have in our lab. The cameras are mounted in the upper left and the lower left and right corners of the image. Figure 1 shows screen shots for a variety of other common setups.

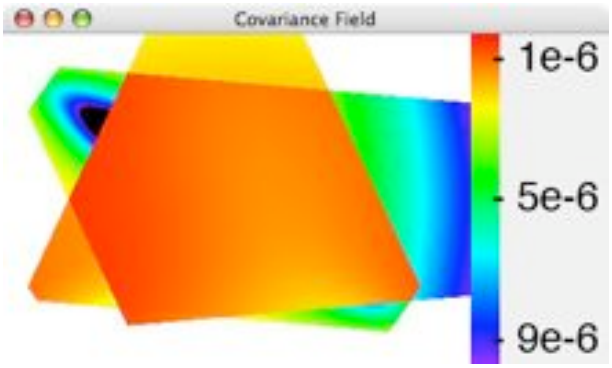


Figure 12: Error Covariances (trace norm) for tracking a single marker ball in a three-camera setup; cameras are mounted in the upper left and the two lower corners

The displayed color denotes the FLE error for each location in the room, where the viewing volumes of the individual cameras are taken into account. We have implemented various different matrix norms to convert the 3×3 covariance matrix to a color map, e. g. the largest eigenvalue (maximum norm) or the trace of the covariance matrix (see figure 13).

In the future, we plan to integrate more advanced rendering techniques like volume rendering [19] for the visualization of the errors.

5 N-Ocular System Detecting Multiple Feature Target

Now we know from section 4 how to estimate the FLE covariance for single features at known locations in space, and from section 3 how to propagate this error into a target registration error at a given point of interest.

Putting all this together, we can compute the covariance of a target at some point of interest for a given camera setup directly from our estimation of covariances in the image plane. When integrated into the tracking system itself, the model also has the possibility to take the number of points that were used for reconstruction into account as well as the number of cameras that were able to see the target.

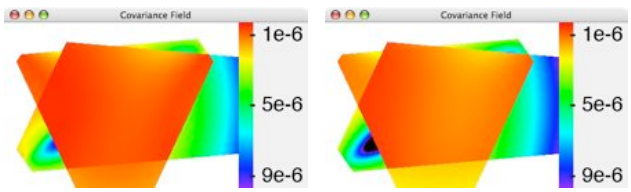


Figure 13: Different matrix norms used for visualization: maximum norm (left) and trace norm (right)

5.1 Visualization of Combined Covariance

Due to the large number of parameters for the calculation of the covariance – the result is different for every location and orientation – a general visualization tool similar to section 3.2 or section 4.4 is not possible. However, several possibilities exist to visualize the results according to the application:

Covariance in Real-Time

If we are interested in the covariance of a given tracking target in a given tracking system at real-time, we can use the concepts from section 3.2 to visualize the current covariance in a separate window or in 3D virtual/augmented reality. The user can then interactively explore the working space and analyze the resulting changes in accuracy. The pose of the target is estimated by the tracking system and the corresponding accuracy computed using the extrinsic and intrinsic camera parameters from the tracking system. This is in particular useful when integrated into the tracking system itself.

Covariance along a path

In several applications, typical movements can be identified for which we would like to know the covariances at some point. Therefore we can record with the tracking system the path of the target for the desired typical action and analyze the covariances along this path offline similar to the tool from section 3.2. Figure 14 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).

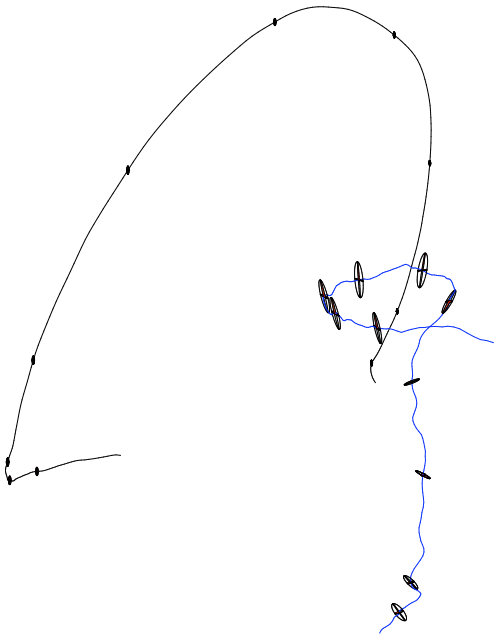


Figure 14: Example track of the covariance along a recorded path (black) and propagated to the point of interest (dotted blue)

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 (TRE) differs substantially along over time. This is due to the fact that the already anisotropic covariances at the centroid get propagated again in an anisotropic way. For unfavorable constellations the already larger error in one direction gets augmented above average. In our 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 stereo camerasystem this would be the case with the line oriented along the optical axes of the cameras. Note that this fact can not be modeled when assuming independent, isotropic and identical FLE for every fiducial [8].

Areas or Volumes of Interest

In many cases we are not really interested in the actual covariance of the measurement; rather we would like to give an error boundary that we want to guarantee for the system at design time. To reach this goal, we can define an area or volume together with some constraints on the orientations which our target is allowed to have in this area. We can then calculate statistics over the resulting error in the region of interest, like the mean error or maximum error – just as in section 4.4 for the single feature. An example for this usage is given in section 5.2.

Error Estimate in the User Interface of an Application

Probably the most obvious usage of our computations is to display the actual error online in the application itself. This could be done by showing a textual representation of the covariance or displaying a circle around the measured point, or by adjusting the user interface in a way suitable to the current accuracy [17].

Computation Time

We have analyzed the speed of the computation 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.

Even more time can be saved by limiting the amount of necessary calculations for every frame. This is in particular useful when for interactive design tools large volumes of predicted accuracies are needed for visualization. We have analyzed the difference of the covariances for a single feature in an area of about the size of normal targets, i.e. a diameter of 20cm in common multi-camera arrangements. Our estimations showed that in typical setups the difference of the covariances in such ranges do not differ widely. Therefore for real-time computations we can safely assume that the covariance for each single feature is approximately the error at the centroid of the marker and therefore we have to do this calculation only once. Taking this error estimation for every feature point we use equation (5) to get the six-dimensional error in the centroid.

5.2 Example Application: Navigated β-probe for tumor resections

In minimally invasive tumor resection, the desirable goal is to perform a minimal but complete removal of cancerous cells. A β-probe is used to detect nuclear labeled malignant cells. Recent work in our group [21] extends the one-dimensional signal of the β-probe to a surface map of the scanned activity.

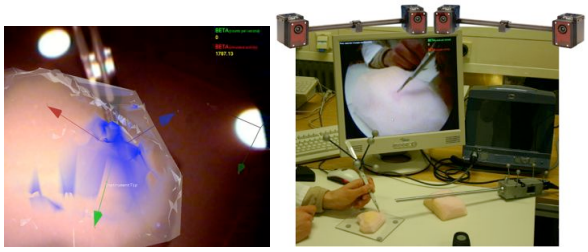


Figure 15: Navigated β-probe for tumor resections

The probe is tracked using a four-camera setup and a retroreflective 4 ball target. The cameras see the working volume from four different sides yielding almost isotropic covariances for the pose estimation of the features. The point of interest, the tip of the probe, is in a distance of about 30 cm from the centroid of the target. We recorded both the camera configuration and a typical movement path from this scenario. Applying our model to this setup, we predict the positional accuracy for a recorded path at the point of interest. Figure 16 shows a typical error covariance from that path at a 95% confidence level: approximately 0.5 mm along the probe (x-axis) and 1.2 mm to 2.4 mm orthogonal to the probe (y-z-axis). These values also correspond to the observed jitter in the augmentation. In this specific application, the values do not differ greatly along the recorded path since the possible movements of the probe are quite restricted by the application.

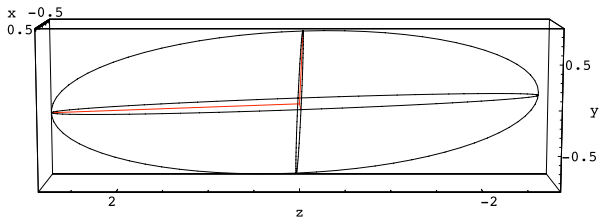


Figure 16: Typical predicted accuracy in the β-probe application, viewing direction along the probe, scale in mm, 95% confidence level

Here we assumed no systematic errors; if we also want to account for systematic errors as shown in figure 10, our experiments showed that we should include an additional factor of 8 to 10 to be safe. This is still sufficient for the application, since the accuracy of the probe itself is, due to construction, only in the range of 10mm.

6 Conclusion

We have presented a theoretical model to predict the accuracy (TRE) of an optical feature tracking system. A set of experiments has been used on the one hand to prove the feasibility of the model and on the other hand to estimate actual parameters for real setups.

The model can be used to evaluate proposed setups for potential augmented reality systems according to their error characteristics before building the system. A set of tools has been implemented to visualize the simulation results.

Likewise the model can be used by any manufacturer of tracking systems to provide at runtime estimates for the actual accuracy of a specific measurement. 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, like for example the number of features used.

Such an estimate of tracking accuracy at runtime is useful for almost any augmented reality application.

Acknowledgments

Part of this work was supported by the Deutsche Forschungsgemeinschaft (KL1460/2), the Bayerische Forschungsstiftung (project TrackFrame, AZ-653-05) and the European Commission (project PRESENCCIA, contract no. 27731). The authors would like to thank *A.R.T. GmbH* for the valuable discussions and for providing us all the technical specifications of their tracking system. Special thanks go to Jörg Traub for providing us real-world data from his scenario as described in the example application.

References

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

[2] WENG, J., COHEN, P., HERNIOU, M., *Camera Calibration with Distortion Models and Accuracy Evaluation*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(10) (1992) 965–980

[3] AZUMA, R. T., BISHOP, G., *Improving Static and Dynamic Registration in an Optical See-through HMD*, in *Proceedings of SIGGRAPH '94* (Orlando, Florida, 1994) 194 – 204

[4] WARE, C., BALAKRISHNAN, R., *Reaching for Objects in VR Displays: Lag and Frame Rate*, in ACM Transactions on Computer-Human Interactions, 1(4) (1994) 331–356, ISSN 1073-0516

[5] ATKINSON, K. (Ed.), *Close Range Photogrammetry and Machine Vision*, Engineering and Science (Whittles Publishing, Scotland, UK, 1996)

[6] FLOROU, G., MOHR, R., *What Accuracy for 3D Measurements with Cameras?*, in *International Conference on Pattern Recognition* (1996) I: 354–358

[7] MANOLAKIS, D. E., *Efficient solution and performance analysis of 3-D position estimation by trilateration*, in Aerospace and Electronic Systems, IEEE Transactions on, 32(4) (1996) 1239–1248

[8] FITZPATRICK, J. M., WEST, J. B., JR., C. R. M., *Predicting error in rigid-body, point-based registration.*, in IEEE Transactions on Medical Imaging, 17(5) (1998) 694–702

[9] HARTLEY, R. I., ZISSERMAN, A., *Multiple View Geometry in Computer Vision* (Cambridge University Press, 2000)

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

[11] MITSCHKE, M., NAVAB, N., *Recovering Projection Geometry: How a Cheap Camera Can Outperform an Expensive Stereo System.*, in *Conference on Computer Vision and Pattern Recognition (CVPR 2000)* (IEEE Computer Society, Hilton Head, SC, USA, 2000) 1193–1200

[12] CHEN, X., *Design of many-camera tracking systems for scalability and efficient resource allocation*, Ph.D. thesis 2002. Adviser-Patrick M. Hanrahan

[13] SAUER, F., KHAMENE, A., VOGT, S., *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* (Springer-Verlag, London, UK, 2002), ISBN 3-540-44225-1 116–124

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

[15] ZHANG, X., FRONZ, S., NAVAB, N., *Visual Marker Detection and Decoding in AR Systems: A Comparative Study*, in *First IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)* (Darmstadt, Germany, 2002)

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

[17] COELHO, E. M., MACINTYRE, B., JULIER, S., *OSGAR: A Scenegraph with Uncertain Transformations*, in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)* (IEEE, Washington, DC, 2004) 6–15

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

[19] ALLEN, B. D., WELCH, G., *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* (ACM Press, New York, NY, USA, 2005) 201–210

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

[21] WENDLER, T., TRAUB, J., ZIEGLER, S., NAVAB, N., *Navigated three dimensional beta probe for optimal cancer resection*, in *Proceedings of 9th International Conference on Medical Image Computing & Computer Assisted Intervention MICCAI* (Copenhagen, Denmark, 2006)