UNIVERSITÄT AUGSBURG

Robust Optical User Motion Tracking Using a Kalman Filter

Klaus Dorfmüller-Ulhaas

Report 2003-06 May 2003

INSTITUT FÜR INFORMATIK
D-86135 AUGSBURG

Robust Optical User Motion Tracking Using a Kalman Filter

Klaus Dorfmüller-Ulhaas*
Multimedia Concepts and Applications
University of Augsburg
Eichleitnerstr. 30
86159 Augsburg, Germany
dorfmueller-ulhaas@informatik.uni-augsburg.de

ABSTRACT

Optical tracking has a great future in applications of virtual and augmented reality. It will assist to enhance the acceptance of virtual reality user interfaces, since optical tracking allows wireless interaction and precise tracking. Existing commercial motion capture systems are neither working reliably in real-time. Additionally, only few optical trackers can smooth and predict motion and include a motion estimator supplying similar results to the presented approach.

A Kalman filter formulation is proposed, providing the robustness required by most virtual reality applications. The filter is evaluated with respect to different rigid body motions and provides precise prediction. Thereby, the frequency and reliability of the optical tracker is enhanced. Predicted motion can be used to cope with display lags of complex virtual scenes or with acquisition or communication delays. The proposed filter may also be used with non-optical based trackers providing the pose of an object with six degrees of freedom.

Keywords

Optical tracking, Kalman filter, motion capture, user motion tracking

1. INTRODUCTION

Virtual environments (VE) immerse users in a fantastic world and enable them to take advantage of the interaction metaphors people are used to for manipulating objects. Within non-desktop virtual reality (VR) applications, users are animated to physically move around and to explore the virtual world. A new view provides a new perspective of the scene, obtaining new details and thus new information about the objects placed in the virtual world. Typically, head movements are the simplest form of interaction that a well designed application supports. The most fascinating applications in virtual environments are highly interactive. Observing humans using entertainment applications, we see that users want to touch objects and manipulate them like children do in exploring the real world.

In order to enable users to move and to interact in this fashion, the virtual reality interface needs precise information about where users stand and in which direction they are looking. This is due to the fact that each view of a user is comparable to a virtual camera and its images need to be calculated and displayed with high frequency and low latency to not cause motion sickness when viewed through a head mounted display (HMD). These are some of the requirements a tracking sensor has to fulfill. Thus, developing a motion tracker is a highly sophisticated task, and different tracking principles do exist using acoustic, optic, magnetic or inertial sensors. In fact, the future of tracking technology is likely in hybrid tracking. This means different sensors are combined to overcome the disadvantages each sensor has and to thereby get an ultimate solution for the tracking problem. It is not surprising that an ultimate tracker does not exist since each VR application poses its own requirements.

There is a wide range of optical based tracking methods, but there are only few systems that are reliable and near product stage so that designers of virtual reality interfaces are able to make use of them. Commonly used optical trackers of human motion capture for computer animation do not meet the real-time constraints of virtual reality applications. Those systems store either a captured image sequence of moving objects or a sequence of cluster centers obtained through hardware implemented segmentation of marker images. Afterwards, the images of markers are matched through user intervention to obtain an initial pose for tracking, which is then done off-line. During motion capturing those optical tracking systems provide high update rates (over 200 Hz) with extremely high accuracy, but also at high cost. Since real-time tracking of the human pose is not a must for computer animation applications new tracking technology needs to be developed.

Recently, inertial sensors with low drift were introduced to track motion for virtual environments [12], but nevertheless, additional sensors are needed to refresh the system with global position data. This can be done using e.g. optical trackers. Therefore, optical tracking provides high accuracy at low-cost due to the fact that more and more cameras are sold at a low price. Connected with a standard PC one can make use of image processing to extract data for tracking human motion in real-time. However, it is well known that image processing is quite computational expensive so

*Parts of this work have been done at ZGDV Darmstadt and Vienna University of Technology

that highly sophisticated algorithms are not applicable in real-time. Nowadays, real-time computer vision uses short and fast algorithms to track motion either by using artificial landmarks or even less restraining by tracking natural features. In order to manage high frequency motion capture, the tracking system developer should be very careful in choosing the landmarks being tracked. For segmenting natural landmarks, more computational time is needed. As a consequence, trackers that utilize natural landmarks are less reliable. Many algorithms have been published recently on natural feature tracking, but since object recognition is still an unsolved problem for real-time processing, those systems are mostly capable to track certain frames after initialization, given an approximate object pose. It is obvious that there is need for a real-time tracking system that is able to perform a self-initialization, that offers a reliable tracking with high precision and high update rate, which can be used in different application environments where dim light is a rule rather than an exception.

This paper proposes an optical tracking system that provides reliable motion tracking results in dim light environments. The robustness of the tracking system is due to the motion estimator and predictor incorporated into our optical tracking procedure. We first give an overview about the tracking system itself and then depict formulas for motion estimation with an extended Kalman filter. Experimental results and simulated examples illustrate the well-performed filtering of the proposed algorithm.

2. OPTICAL TRACKING SYSTEM

As we started to develop the proposed tracking system, we acted from necessity. The application scenario allows for using a responsive workbench as a semi-immersive display, but available tracking technology fails either at tracking with sufficient accuracy in environments containing metallic objects or at enabling wireless interaction. Therefore, optical tracking has become a promising alternative, since it provides both, high precision and wireless user motion tracking. It should be mentioned here that we started to implement a first version of our optical tracker in 1997 [11] and about that time, commercial optical tracking systems (like e.g. the ART-Optical Tracking System [1]) were not available or did not fulfill the requirements when using them at the responsive workbench. But nevertheless, our tracking system has many advantages in comparison to other optical trackers. One advantage is the integration of a Kalman filter process that makes tracking more reliable, enhances frequency and optimizes latency. Thus, we want to discuss its implementation in this paper and emphasize that its interface and filter formulation is not limited to optical tracking. Any other tracker that provides the pose of a rigid body with six degrees of freedom can be used as input for the Kalman filter process. Another advantage of our tracking system is a camera calibration that estimates intrinsic and extrinsic camera parameters in one step. Generally, commercial optical trackers are calibrated internally by the manufacturer. Afterwards, external calibration is performed by the customer. However, in case lenses need to be replaced, cameras must be send back to the manufacturer for internal calibration. The calibration method we have developed does not require an explicit internal calibration and is thus very easy to use and user friendly. Additionally, the tracking system is affordable for most public utilities like museums and research institutes.

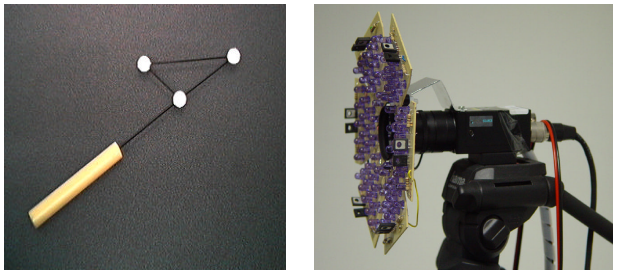


Figure 1: 6 DoF Interaction tool and camera with infrared spot light and infrared pass filter for retro-reflective marker tracking

When using the tracking system at the responsive workbench, lighting conditions around the responsive workbench have to be subdued to not decrease the brightness of the projection. This limits an optical tracking system to using infrared light, in which case one has two possibilities to choose from: either the system may use active infrared beacons, or the tracking area has to be illuminated with infrared light while the objects to be tracked are fitted with reflective markers. Our first implementation used active infrared beacons [11] that are self-powered. Because of that, shutterglasses fitted with active markers became heavy making them uncomfortable. Additionally, the amount of light emitted by the beacons is restricted by its LED specific radiance angle. As we wanted to enable interaction with six degrees of freedom, we extended our tracker to a more elaborated passive marker tracking system as depicted in Fig. 1.

In order to enable the users to walk around three sides of the responsive workbench, the camera positions were restricted to the far sides of the table, to the left and to the right of the projector. Figure 2 illustrates the constellation of the cameras at the responsive workbench. Mounting the cameras closer together (e.g. both on top of the projector cover) would create accuracy problems as the 3D position of the beacon can be best calculated when the cameras are arranged with a convergence angle of between 60 and 120 degrees between them.

The proposed optical outside-in tracker has been used in a

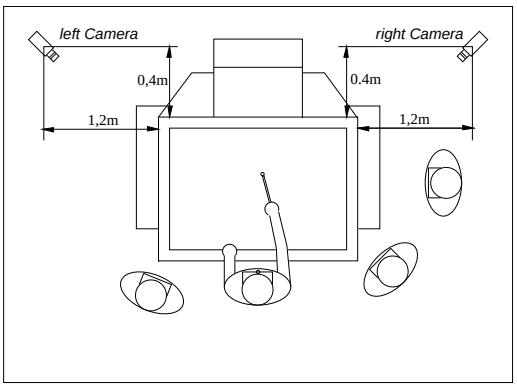


Figure 2: Camera positions at the responsive workbench

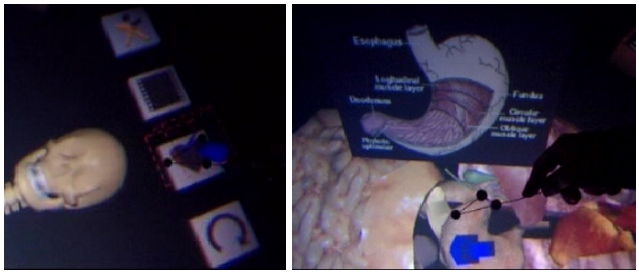
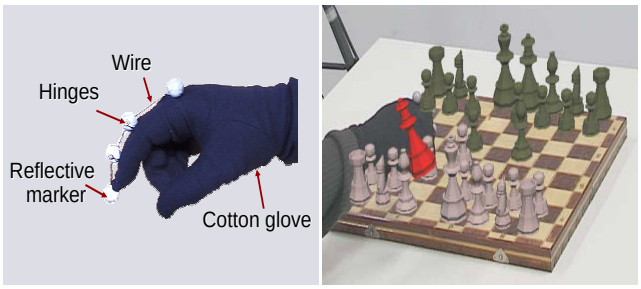


Figure 3: Interacting with virtual objects

medical education application as shown in Fig. 3. Through collision detection of a virtual representative of the pointing device, button-press-and-release events are simulated and cause changes of the virtual scene according to the source of a callback function. In order to enhance the robustness of emitting manipulation events, we have added a button to the interaction device that transmits button-press-and-release events via wireless radio link. The medical education application allows to grab and rotate objects. To learn more about the viscera of humans the user has to collide with their virtual representations whereupon a two-dimensional information card gives additional information in a specialized-book-like metaphor.

One of the perspectives of the optical tracker is to track the full body of a human to enable real-time character animation or full body interactions. This requires tracking of flexible objects and incorporates kinematic chains. One of our extensions to the tracking system includes the detection of a glove fitted with markers. Displaced balls on the back of the finger (Fig. 4) were used that have good retro-reflective properties and are not often significantly occluded by the



(a) (b)

Figure 4: Glove fitted with retroreflective markers to interact in an augmented reality chess game

fingers themselves. The use of displaced balls was enhanced by connecting the balls with short pieces of wire mounted to hinges in the balls to enforce a fixed known distance between the balls. Dimensions of these wire rods were chosen to match the distances between finger joints. While this "exoskeleton" looks awkward, it has the great advantage that it follows the behavior of the finger as a kinematic chain, but with easily detectable joint centers. Experiences confirmed that it does not affect finger movement or interaction in any noticeable way.

For reconstruction, a 3D finger model based on a kinematic chain of the finger joints was employed that directly maps onto the markers. As the distance of the markers is known, the system is independent of the actual dimensions of the user's finger (within certain limits), while the soft glove's material can be stretched to fit any user. The only remaining user specific parameter is the actual offset from the user's finger tip to the last marker in the chain which is used to determine the 6 DoF "hot spot". To enable a user to interact with his or her finger tip, this offset must be determined. However, most users showed to be willing to accept that the actual hot spot is deviant by a small amount from their finger tip, and interaction is not affected. In this paper, we limit ourselves to 6 DoF tracking and explain how motion estimation of a rigid body works within our optical tracking system.

In order to illustrate how Kalman filtering enhances the robustness of an optical tracking system, let us consider Fig. 5. The images for the left and right camera view are segmented to extract the markers' center of gravity. Herewith, the pose predicted by the Kalman filter is projected to the image planes. This improves the frequency of the tracking system, since information is provided about where marker images may be located in the current frame. For matching the marker images between camera image planes, the pose predicted by the Kalman filter provides additional information preventing wrong matches and undesired triangulations of three-dimensional marker positions not existent in reality. However, it can not completely assured that only marker po-

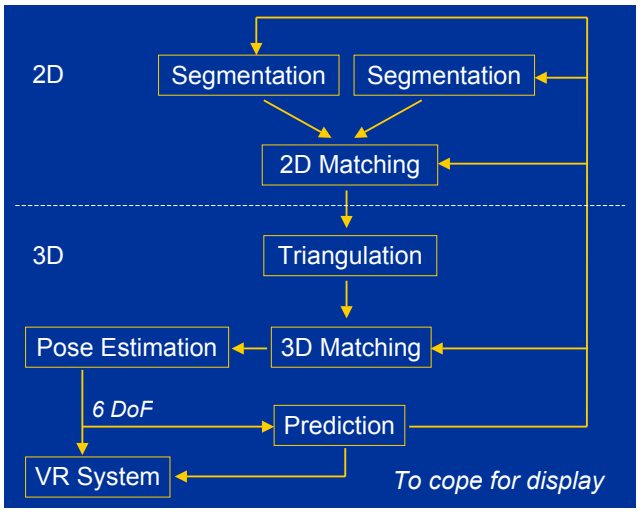


Figure 5: Image processing tasks

sitions are three-dimensionally reconstructed. Reflections of markers caused by glass or other reflective material in the environment can provide additional marker positions. This is why an additional matching algorithm was implemented to find known sub-structures in a 3D point cloud. This problem is NP-complete and could be solved much faster if heuristics are available. In fact, the Kalman filter provides the required information. Last but not least, pose prediction can be used to cope with display lags, communication delays and to reduce the latency of the optical tracker.

One low-cost optical tracking system that has inspired our work essentially was invented by Madritsch and Gervautz [16]. Disadvantages of their system include the use of LEDs for tracking, the use of unsynchronized cameras and thus, it was impossible for the system to prevent errors in pose esti-

mation during motion. Calibration is done with the cumbersome and error-prone Tsai [20] calibration. A system using synchronized cameras and an easy-to-use stereoscopic camera calibration was proposed by Dorfmüller and Wirth [11]. The use of retro-reflective markers for real-time tracking in VR and the invention of hand-held tools for interaction with six degrees of freedom was introduced by Dorfmüller [10]. Different versions of this optical tracking system are currently in use at ZGDV in Darmstadt, Germany, at the University of Münster, Germany, at Ewha Womans University Seoul, South Korea, at the Vienna University of Technology, Austria, and at the University of Augsburg, Germany. Application-related articles have been published by some of these institutes [17, 15]. A re-implementation of and extensions to the proposed tracking system were developed and published [19]. A similar optical tracker was created by Mulder and Liere [18] and Chung *et al.* [9]. None of those outside-in motion trackers has incorporated a motion estimator to support a robust tracking in virtual environments.

3. MOTION ESTIMATION

Motion tracking is one of the most important parts of a robustly working system. Many optical tracking systems estimate the pose of an object at one frame more or less independent of a previous frame. Prediction extrapolates motion and can be used for motion smoothing and provides heuristic information for optimization and matching problems. The filter formulation proposed in this paper also addresses virtual reality system developers who want to cope with latency problems of their systems while displaying complex virtual scenes. We propose here an elegant way to predict and smooth motion of a rigid body with six degrees of freedom.

For motion estimation we have chosen to use an extended Kalman filter, because Kalman filtering has the feasibility to model noise, even allowing the system to filter state values in noisy environments. Figure 6 shows how measurements on the camera image planes can be used as indirect measurements for the proposed EKF. First, function F_1 backprojects

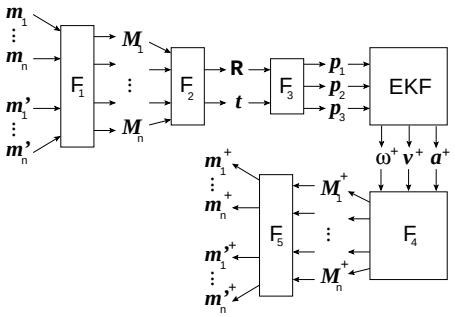


Figure 6: Predicting the measurements on the image planes of a stereo rig

the measurements to 3D space (compare pages 295ff in [14]). Method F_2 calculates the rotation and translation of a rigid body given multiple 3D point measurements corresponding to known ideal 3D points of the rigid body. It was shown by Arun *et al.* [3] and Umeyama [21] how two 3D point sets are related by rotation and translation. Umeyama has refined the closed-form solution introduced by Arun for the case of false matches. A nonlinear estimation algorithm refines the rotation and translation values by comparing real measurements of marker images with ideal ones obtained through projection while incorporating the rigid bodies pose. After rigid body's rotation matrix $\mathtt{R}$ and translation vector $\boldsymbol{t}$ have been precisely estimated, three imaginary points on the rigid body can be computed to provide the input for the proposed EKF. Motion kinematics as described later are used to predict the velocity and angular values of the rigid body. Function F_3 calculates three arbitrary points located in the coordinate system of the rigid body. Angular velocity $\boldsymbol{\omega}$, translational velocity $\boldsymbol{v}$, and translational acceleration $\boldsymbol{a}$ are filtered from these measurements and can be predicted through an EKF. Module F_4 transforms each 3D object point to be measured corresponding to the predicted pose of the rigid body using the estimated prediction values of the EKF. Function F_5 is a projection of predicted object points in 3D space on the camera image planes. If the Kalman filter is used with other tracking sensors, only function F_3 is necessary, the extended Kalman filter itself, and a postprocessing function F_4 that calculates the predicted pose $\mathtt{R}^+$ and $\boldsymbol{t}^+$ using the predicted state vector values of the Kalman filter $\boldsymbol{\omega}^+$, $\boldsymbol{v}^+$ and $\boldsymbol{a}^+$. This can be easily implemented using Theorem 1, which is introduced in the following.

Much work has been done recently for determining 3D motion and structure of moving rigid objects in computer vision [24, 8, 4]. The extraction of motion and shape parameters of a moving rigid 3D object from a 2D image sequence is often named the *structure from motion problem*. However, the number of unknown parameters is directly related to estimation performance and data requirements. If many parameters like e.g. structure parameters are unknown, the estimation process can be delicate and difficult. If fewer parameters, e.g. only kinematic parameters are involved, like in [5, 26], the estimation process is more robust and applicable for real-time tracking. Most motion estimators are based on EKF which has been the subject of much recent work [5, 6, 7, 13, 22, 25], but it should be mentioned that other approaches like the Levenberg-Marquardt technique do exist that provide better and faster convergence under some circumstances (compare [23]). Since Kalman filtering is robust for measurements that suffer from noise, it is in the focus of this paper.

3.1 Motion Model

Motion of a rigid body in three-dimensional space is considered a transformation including translation and rotation. Translation has three degrees of freedom and can be represented as a linear function. Rotations represented as matrices are also linear functions if applied to a point in space. However, a rotation matrix is not a minimal rotation representation. A rotation matrix is defined to be orthonormal and thus, a rotation has only three degrees of freedom. Using three parameters to represent a rotation results in nonlinear functions and in addition singularities are obtained through the use of a three parameter representation. Quaternions have four degrees of freedom and are free from singularities. If derivatives of rotation known as angular velocities are needed quaternion representations do not satisfy. Since quaternions use four parameters to represent a three degree of freedom rotation, there is always a direction of change in calculating the partial derivative that is not a rotation. If a

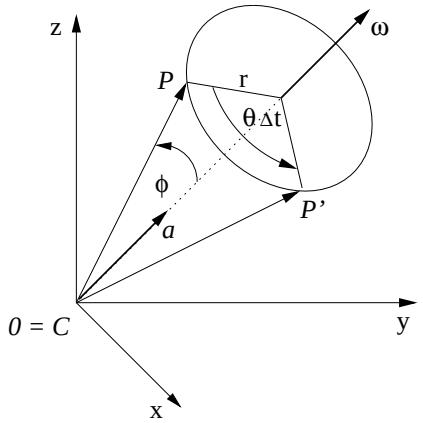


Figure 7: The local coordinate system of the rigid body, where C was chosen to be the origin. The velocity of point C and point P is the same up to a rotation of P around C.

rotation is changed in the direction of its partial derivative, the quaternion gets non-unit length and does no longer represent a rotation. A re-normalization can be performed, but the result is not exact and numerically unstable. Therefore, we propose to use a minimal rotation representation that has its origin in mechanics and is known as the *exponential map* which does not have the disadvantages of a quaternion representation.

Consider Figure 7, depicting a 3D rotation in space by an angle $\theta\,\Delta t$ about an axis of rotation $\boldsymbol{\omega}$. The rotation representation defines the instantaneous angle of rotation θ with $\Delta t = 0$ to be the length of vector $\boldsymbol{\omega}$, hence $\theta\,\Delta t = \|\boldsymbol{\omega}\|\,\Delta t = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}\,\Delta t$ and the direction of $\boldsymbol{\omega}$ is that of the rotation axis.

The general motion of a point $\boldsymbol{P}$ on a rigid body may be estimated by using Taylor series in order to approximate the translational velocity $\boldsymbol{v}_t$ and the angular velocity $\boldsymbol{\omega}_t$ at time t. This would give the following equations:

$$\begin{aligned} \boldsymbol{v}_t &= \sum_{i=0}^{n} \boldsymbol{v}_i \frac{(t-t_0)^i}{i!} \qquad (1) \\ &= \boldsymbol{v}_0 + \boldsymbol{v}_1 \Delta t + \boldsymbol{v}_2 \frac{(\Delta t)^2}{2} + \cdots + \boldsymbol{v}_n \frac{(\Delta t)^n}{n!} \\ \boldsymbol{\omega}_t &= \sum_{i=0}^{m} \boldsymbol{\omega}_i \frac{(t-t_0)^i}{i!} \qquad (2) \end{aligned}$$

$\boldsymbol{v}_0$, $\boldsymbol{\omega}_0$ are known as the translational and rotational velocity at time $t-1$ and $\boldsymbol{v}_1$, $\boldsymbol{\omega}_1$ are the translational and rotational acceleration parameters at time $t-1$. $\boldsymbol{v}_2$, $\boldsymbol{\omega}_2$ are the physical jerk. In practice it is sufficient to approximate the motion of a rigid body by the first one or two terms of the *Taylor series*. As long as Δt is small, this linearized kinematic model fits well. Now, we can introduce Theorem 1 which represents a linearized motion model used within our implementation:

THEOREM 1. *The trajectory of a point $\boldsymbol{P}$ transfomed from $\boldsymbol{p}_0$ to $\boldsymbol{p}_t$ in time Δt is given in the case of constant angular velocity $\boldsymbol{\omega}$, linear translational velocity $\boldsymbol{v}$ and constant translational acceleration $\boldsymbol{a}$, by the following equation:*

$$\boldsymbol{p}_t = \mathtt{W}\boldsymbol{p}_0 + \mathtt{V}\boldsymbol{v} + \mathtt{A}\boldsymbol{a} \qquad (3)$$

where $\mathtt{W}$, $\mathtt{V}$ *and* $\mathtt{A}$ *are given as:*

$$\begin{aligned} \mathtt{W} &= \mathtt{I}_3 + \frac{\sin(\theta\Delta t)}{\theta}\tilde{\omega} + \frac{1-cos(\theta\Delta t)}{\theta^2}\tilde{\omega}^2 = e^{\tilde{\omega}\Delta t} \qquad (4) \\ \mathtt{V} &= \mathtt{I}_3\Delta t + \frac{1-\cos(\theta\Delta t)}{\theta^2}\tilde{\omega} + \frac{\theta\Delta t - \sin(\theta\Delta t)}{\theta^3}\tilde{\omega}^2 \qquad (5) \\ \mathtt{A} &= \frac{\Delta t^2}{2}\mathtt{I}_3 + \frac{\theta\Delta t - \sin(\theta\Delta t)}{\theta^3}\tilde{\omega} \\ &+ \frac{(\theta\Delta t)^2 - 2(1-\cos(\theta\Delta t))}{2\theta^4}\tilde{\omega}^2 \qquad (6) \end{aligned}$$

where

$$\tilde{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

is a skew symmetric matrix and $\tilde{\omega}^2 = \tilde{\omega} \cdot \tilde{\omega}$

Herewith, the rotation of a point $\boldsymbol{P}$ is given by the exponential map of vector $\boldsymbol{\omega}\Delta t$, which is denoted as $e^{\tilde{\omega}\Delta t}$ and describes a rotation matrix. Equation 4 is well known as the Rodrigues formula [2]. The translation of a point $\boldsymbol{P}$ is given by $\boldsymbol{t} = \mathtt{V}\boldsymbol{v} + \mathtt{A}\boldsymbol{a}$ [1].

3.2 Tracking

There are different possibilities how to solve the tracking problem when rotation and translation data are used as sensor data input. Often the translational data and the rotational data are treated independently and stored in the state vector of a Kalman filter process,

$$\begin{pmatrix} \mathtt{R} - \hat{\mathtt{R}}^+ \\ \boldsymbol{t} - \hat{\boldsymbol{t}}^+ \end{pmatrix} = \mathbf{0} \qquad (7)$$

where $\hat{\mathtt{R}}^+$ and $\hat{\boldsymbol{t}}^+$ are a predicted rotation matrix and translation vector, respectively. In Augmented Reality the registration is best performed when virtual and real data align so that the observer does not perceive any offset. This alignment is dependent on the measured rotation and translation obtained through the tracking system. Thus, we assume that modeling rotation and translation independently does not assure that the final pose of a rigid body in VR aligns best with the real one. We propose to use equations that minimize the error of transformed points which is perceived by the user of a VR system instead of treating those parameters independently and use a minimization as follows:

$$\begin{pmatrix} \boldsymbol{M}_1 - \hat{\boldsymbol{M}}_1(\mathtt{R}, \boldsymbol{t})^+ \\ \boldsymbol{M}_2 - \hat{\boldsymbol{M}}_2(\mathtt{R}, \boldsymbol{t})^+ \\ \boldsymbol{M}_3 - \hat{\boldsymbol{M}}_3(\mathtt{R}, \boldsymbol{t})^+ \end{pmatrix} = \mathbf{0} \qquad (8)$$

[1] This formula can also be used for calculating a predicted translation vector using the predicted acceleration and velocity parameters of the Kalman filter

where $\hat{\boldsymbol{M}}_i(\mathtt{R}, \boldsymbol{t})^+$ is the i^{th} predicted point located on the rigid body using predicted pose parameters $\mathtt{R}$ and $\boldsymbol{t}$. It is left to future work to evaluate this statement by comparing these two different approaches. It is recommended in this paper that a relative movement of at least three 3D points should be modeled in order to enhance the accuracy of a predicted rigid body motion.

The Kalman filter requires a state vector holding the following variables in our application:

$$\boldsymbol{s}_t = \begin{pmatrix} \boldsymbol{\omega}_t & \boldsymbol{v}_t & \boldsymbol{a}_t \end{pmatrix}^T \tag{9}$$

where $\boldsymbol{\omega}_t$ is the constant angular velocity, $\boldsymbol{v}_t$ is the translational velocity, and $\boldsymbol{a}_t$ is the constant translational acceleration. In fact, we store only relative motion parameters in the state vector since global orientation and translation can be calculated from relative motion.

The filter formulation foresees a set of six 3D measurements in the measurement vector, where three variables denote the current measurements at time t and three variables give the previous measurements at time $t-1$. This formulation leads us to the estimation of relative motion.
The measurement vector is given as

$$\boldsymbol{x}_t = \begin{pmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \boldsymbol{P}_3 & \boldsymbol{P}'_1 & \boldsymbol{P}'_2 & \boldsymbol{P}'_3 \end{pmatrix}^T \tag{10}$$

In case only rotational $\mathtt{R}_t$ and translational data $\boldsymbol{t}_t$ are available from the sensor, these three 3D points may be calculated using three non-coplanar vectors, for instance the unit vectors of the coordinate system $\boldsymbol{e}_1 = (1,0,0)^T$, $\boldsymbol{e}_2 = (0,1,0)^T$, and $\boldsymbol{e}_3 = (0,0,1)^T$. The points needed for the measurement vector are then obtained by the following transformation.

$$\boldsymbol{P}_i = \mathtt{R}_t \boldsymbol{e}_i + \boldsymbol{t}_t \tag{11}$$

For prediction of the state vector, the EKF uses the following equation:

$$\hat{\boldsymbol{s}}_{t|t-1} \quad = \quad h_t(\hat{\boldsymbol{s}}_{t-1})$$

Since the rotation parameters have been omitted from the state vector, the state prediction is a linear function and can thus be simplified as

$$\hat{\boldsymbol{s}}_{t|t-1} \quad = \quad \mathtt{H}_t \hat{\boldsymbol{s}}_{t-1}$$

where $\mathtt{H}$ is a matrix and is given as

$$\mathtt{H} = \begin{pmatrix} \mathtt{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathtt{I}_3 & \mathtt{I}_3\,\Delta t \\ \mathbf{0} & \mathbf{0} & \mathtt{I}_3 \end{pmatrix} \tag{12}$$

From Theorem 1, the following function which is minimized by the Kalman filter, may be derived.

$$f\left(\boldsymbol{x}'_t, \hat{\boldsymbol{s}}_{t|t-1}\right) = \begin{pmatrix} \mathtt{W}\boldsymbol{P}_1 + \mathtt{V}\boldsymbol{v} + \mathtt{A}\boldsymbol{a} - \boldsymbol{P}'_1 \\ \mathtt{W}\boldsymbol{P}_2 + \mathtt{V}\boldsymbol{v} + \mathtt{A}\boldsymbol{a} - \boldsymbol{P}'_2 \\ \mathtt{W}\boldsymbol{P}_3 + \mathtt{V}\boldsymbol{v} + \mathtt{A}\boldsymbol{a} - \boldsymbol{P}'_3 \end{pmatrix} = \mathbf{0} \tag{13}$$

This function is a residual between the measurement determined by using the motion model applied with the state vector and the current measurement. If the state vector and the motion model are exact and the measurement does not include any noise, this function accumulates to a nine-dimensional zero vector. Function $f(\boldsymbol{x}, \boldsymbol{s})$ is a nonlinear function since the matrices $\mathtt{W}$, $\mathtt{V}$, and $\mathtt{A}$ are nonlinear expressions. In order to apply the EKF algorithm, it is necessary to compute the derivatives of $f(\boldsymbol{x}, \boldsymbol{s})$ with respect to $\boldsymbol{x}$ and $\boldsymbol{s}$. However, these derivatives include singularities as we will see later in this section. We will provide a solution to this problem.
The derivative $\frac{\partial f}{\partial \boldsymbol{x}}$ can be computed as:

$$\frac{\partial f}{\partial \boldsymbol{x}} = \begin{pmatrix} \mathtt{W} & -\mathtt{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathtt{W} & -\mathtt{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathtt{W} & -\mathtt{I}_3 \end{pmatrix} \tag{14}$$

Singularities occur due to the derivative with respect to $\boldsymbol{s}$:

$$\frac{\partial f}{\partial \boldsymbol{s}} = \begin{pmatrix} \frac{\partial f_1}{\partial \boldsymbol{\omega}} & \mathtt{V} & \mathtt{A} \\ \frac{\partial f_2}{\partial \boldsymbol{\omega}} & \mathtt{V} & \mathtt{A} \\ \frac{\partial f_3}{\partial \boldsymbol{\omega}} & \mathtt{V} & \mathtt{A} \end{pmatrix} \tag{15}$$

where

$$\frac{\partial f_i}{\partial \boldsymbol{\omega}} = \frac{\partial(\mathtt{W}\boldsymbol{P}_i)}{\partial \boldsymbol{\omega}} + \frac{\partial(\mathtt{V}\boldsymbol{v})}{\partial \boldsymbol{\omega}} + \frac{\partial(\mathtt{A}\boldsymbol{a})}{\partial \boldsymbol{\omega}} \tag{16}$$

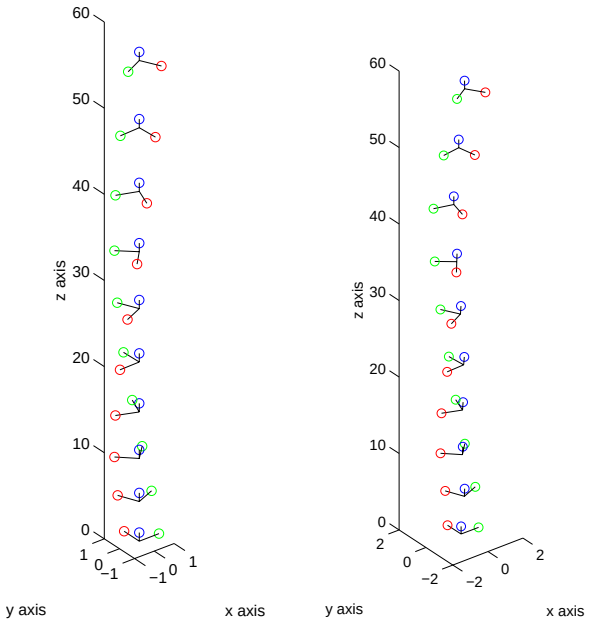
Here, the derivatives of matrices $\mathtt{W}$, $\mathtt{V}$, and $\mathtt{A}$ are needed and determined as follows:

$$\begin{aligned}
\frac{\partial(\mathtt{W}\boldsymbol{P}_i)}{\partial \boldsymbol{\omega}} &= -\frac{\sin(\theta\Delta t)}{\theta}\tilde{\boldsymbol{P}}_i \\
&+ \frac{\theta\Delta t\cos(\theta\Delta t) - \sin(\theta\Delta t)}{\theta^3}(\tilde{\omega}\boldsymbol{P}_i)\boldsymbol{\omega}^T \\
&+ \frac{\theta\Delta t\sin(\theta\Delta t) - 2(1-\cos(\theta\Delta t))}{\theta^4}(\tilde{\omega}(\tilde{\omega}\boldsymbol{P}_i))\boldsymbol{\omega}^T \\
&+ \frac{1-\cos(\theta\Delta t)}{\theta^2}\left[-\widetilde{\omega\boldsymbol{P}_i} + (\boldsymbol{\omega}^T\boldsymbol{P}_i)\mathtt{I}_3 - \boldsymbol{P}_i\boldsymbol{\omega}^T\right] \\
\frac{\partial(\mathtt{V}\boldsymbol{v})}{\partial \boldsymbol{\omega}} &= -\frac{1-\cos(\theta\Delta t)}{\theta^2}\tilde{\boldsymbol{v}} \\
&+ \frac{\theta\Delta t\sin(\theta\Delta t) - 2(1-\cos(\theta\Delta t))}{\theta^4}(\tilde{\omega}\boldsymbol{v})\boldsymbol{\omega}^T \\
&+ \frac{3\sin(\theta\Delta t) - \theta\Delta t(2+\cos(\theta\Delta t))}{\theta^5}(\tilde{\omega}(\tilde{\omega}\boldsymbol{v}))\boldsymbol{\omega}^T \\
&+ \frac{\theta\Delta t - \sin(\theta\Delta t)}{\theta^3}\left[-\widetilde{\omega\boldsymbol{v}} + (\boldsymbol{\omega}^T\boldsymbol{v})\mathtt{I}_3 - \boldsymbol{v}\boldsymbol{\omega}^T\right] \\
\frac{\partial(\mathtt{A}\boldsymbol{a})}{\partial \boldsymbol{\omega}} &= -\frac{\theta\Delta t - \sin(\theta\Delta t)}{\theta^3}\tilde{\boldsymbol{a}} \\
&+ \frac{3\sin(\theta\Delta t) - \theta\Delta t(2+\cos(\theta\Delta t))}{\theta^5}(\tilde{\omega}\boldsymbol{a})\boldsymbol{\omega}^T \\
&+ \frac{4(1-\cos(\theta\Delta t)) - (\theta\Delta t)^2 - \theta\Delta t\sin(\theta\Delta t)}{\theta^6}(\tilde{\omega}(\tilde{\omega}\boldsymbol{a}))\boldsymbol{\omega}^T \\
&+ \frac{(\theta\Delta t)^2 - 2(1-\cos(\theta\Delta t))}{2\theta^4}\left[-\widetilde{\omega\boldsymbol{a}} + (\boldsymbol{\omega}^T\boldsymbol{a})\mathtt{I}_3 - \boldsymbol{a}\boldsymbol{\omega}^T\right]
\end{aligned}$$

Even if $\boldsymbol{\omega} = \mathbf{0}$, the derivatives are not defined since θ become zero. We may cope with this property if the limes near $\boldsymbol{\omega} = \mathbf{0}$ can be determined. Most terms of the derivative becomes zero since $\boldsymbol{\omega}$ becomes zero. For the others, the limes of θ becoming zero is calculated by applying the rule of *de l'Hospital*:

$$\begin{aligned}
\lim_{\theta\to 0}\frac{\partial(\mathtt{W}\boldsymbol{P}_i)}{\partial \boldsymbol{\omega}} &= \lim_{\theta\to 0} -\frac{\sin(\theta\Delta t)}{\theta}\tilde{\boldsymbol{P}}_i \\
&= \lim_{\theta\to 0} -\frac{\Delta t\cdot\cos(\theta\Delta t)}{1}\tilde{\boldsymbol{P}}_i = -\Delta t\tilde{\boldsymbol{P}}_i \\
\lim_{\theta\to 0}\frac{\partial(\mathtt{V}\boldsymbol{v})}{\partial \boldsymbol{\omega}} &= \lim_{\theta\to 0} -\frac{1-\cos(\theta\Delta t)}{\theta^2}\tilde{\boldsymbol{v}} \\
&= \lim_{\theta\to 0} -\frac{\cos(\theta\Delta t)\Delta t^2}{2}\tilde{\boldsymbol{v}} = -\frac{\Delta t^2}{2}\tilde{\boldsymbol{v}} \\
\lim_{\theta\to 0}\frac{\partial(\mathtt{A}\boldsymbol{a})}{\partial \boldsymbol{\omega}} &= \lim_{\theta\to 0} -\frac{\theta\Delta t - \sin(\theta\Delta t)}{\theta^3}\tilde{\boldsymbol{a}} \\
&= \lim_{\theta\to 0} -\frac{\Delta t^3\cos(\theta\Delta t)}{6}\tilde{\boldsymbol{a}} = -\frac{\Delta t^3}{6}\tilde{\boldsymbol{a}}
\end{aligned}$$

So far, the filter formulation is complete and we may per-



(a) (b)

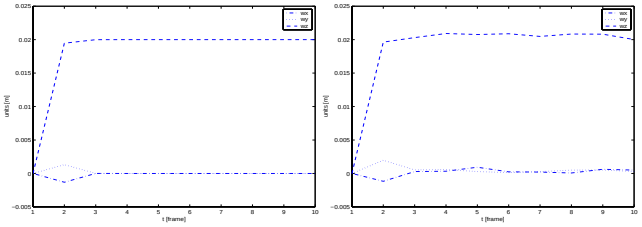
Figure 8: Motion simulation of a rigid body. (a) The rigid body is translated and rotated so that the origin is still moved on a straight line. (b) The measured translation and rotation of the rigid body is corrupted by noise. The origin of the rigid body's coordinate system is no longer located on a straight line, causing the coordinate system to be enlarged in x-y direction.

form some experiments with simulated data to evaluate a successful working of the EKF.

3.3 Experimental Results

We first have implemented the proposed extended Kalman filter formulation using MatLab^{TM} and then translated the code to C++ to integrate it into our optical tracker. The MatLab^{TM} application serves us to perform simulations and to evaluate the filtering of rigid body motion. In the following, we depict three of our simulations and show plots of filtered motion data to see the correct working of the filter and the weaknesses if the tracker's frequency does not fulfill Shannon's sampling theorem.

For a first experiment, let us assume the constant angular velocity of a moving object is given by $\boldsymbol{\omega} = (0,\ 0,\ 0.02)^T$ and the constant acceleration is given by $\boldsymbol{a} = (0,\ 0,\ 0.001)^T$. The translational velocity of the previously introduced motion model is not constant, but linear, so the velocity accumulates over time to $\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \boldsymbol{a}\Delta t$. The initial velocity motion of a rigid body is assumed to be $\boldsymbol{v}_0 = (0,\ 0,\ 0.2)^T$. The time between successive frames is taken to be constant for simplification and defined as $\Delta t = 20\,ms$. Figure 8 (a) shows this experimental rigid body motion. Three points located imaginarily on the rigid body are displayed with different color circles. Ten time steps of this rigid body motion are captured. A Kalman filter assumes that the measurements may be perturbed by white noise. The measurements of the second experiment we may consider are rectangularly distributed. In contrast of using a gaussian distribution we will see how the Kalman filter behaves if the error induced is different from white noise. The range for varying the angular velocity $\boldsymbol{\omega}$ is $\pm 10^{-3}$, for varying the translational velocity $\pm 10^{-2}$ and finally the translational acceleration is perturbed in the range of $\pm 10^{-5}$. A motion of a rigid body suffering from this random noise is depicted in Fig. 8 (b). The EKF formulation was applied to the artificial 3D point data of Fig. 8 (a) and (b). The kinematic parameters are extracted by the filter and predicted for $\Delta t = 20\,ms$. Figure 9 shows the prediction of angular velocity, where (a) is obtained using ideal and (b) using corrupted data. It can

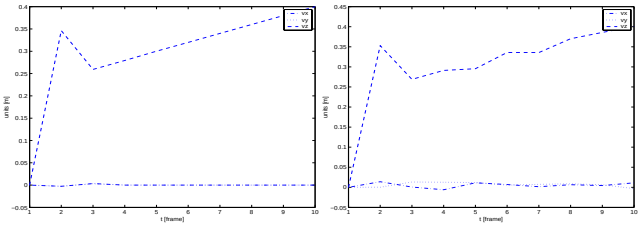


(a) (b)

Figure 9: Prediction values of angular velocity. (a) Ideal measurement data are used. (b) Corrupted data are used. After three iterations, the prediction of the rotational velocity is stable and converges to the true values $\boldsymbol{\omega} = (0,\ 0,\ 0.02)^T$.

be seen in Fig. 9 that for both cases the EKF converges to the true values of the angular velocity after three iterations. After three iterations the covariance matrix P has been adapted to the specific motion of the rigid body and the angular velocity can be precisely predicted. In the first case the angular velocity is $\boldsymbol{\omega} = (0,\ 0,\ 0.02)^T$ and for the second case it is close to these values.

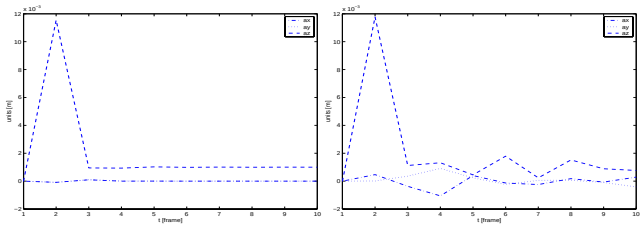
The translational velocity is predicted as illustrated in Fig. 10. Remember that the translational velocity is not constant. In the underlying motion model it is linear which



(a) (b)

Figure 10: Prediction of translational velocity. (a) Ideal measurement data are used. (b) Corrupted data are used. After three iterations, the prediction of the translationasl velocity is stable. The velocity is a linear function $\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \boldsymbol{a}\Delta t$ what can be seen by the value of v_z.

can be seen in Fig. 10 (a) after three iterations for the value v_z. Noise does not influence the convergence of the Kalman filter. Fig. 11 depicts the prediction of translational acceleration derived from this motion prediction experiment. The following experiment shows how the filter behaves if the sampling rate is of low frequency. Shannon's sampling theorem says that the measurement or sampling frequency



(a) (b)

Figure 11: Prediction of translational acceleration. (a) Ideal measurement data are used. (b) Corrupted data are used. After three iterations, the prediction of the translational acceleration is stable. It can be seen that the translational acceleration is more sensitive to noise than e. g. translational velocity.

should be at least twice the true target motion. We now assume an abrupt change of target motion of the rigid body at time frame 10. For such generated measurements the sampling does not fulfill the previously mentioned requirements for the change of direction from frame 9 over 10 to 11 (compare Fig. 12). The rigid body motion shown in Fig. 8 is supplemented with an abrupt change in motion defined by the following kinematic vectors $\boldsymbol{\omega} = (0.005,\ -0.01,\ -0.02)^T$, $\boldsymbol{v} = (0.2,\ 0.1,\ -0.2)^T$ and $\boldsymbol{a} = (0.001,\ 0.0008,\ -0.001)^T$. This situation is shown in Fig. 12. It can be seen in Fig. 13 and Fig. 14 that the filter realigns after three to four

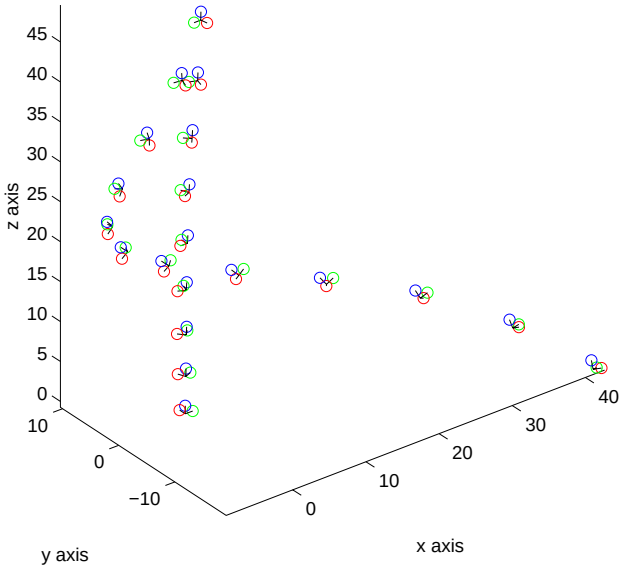


Figure 12: Simulation of a sudden change of direction. The motion data of a rigid body are corrupted by noise. From frame 9 to 10 the direction of motion is abruptly changed to evaluate the behavior of the EKF for a too low sampling rate.

iterations. The translational acceleration is more sensitive to this abrupt change and needs about five iterations. If the sampling frequency is not high enough, the resulting predicted pose results in overshoots as can be seen in frame 11 of Fig. 15. Thus, it is crucial that the tracker fulfills this sampling theorem.

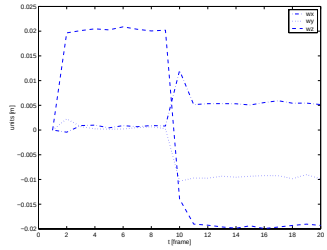
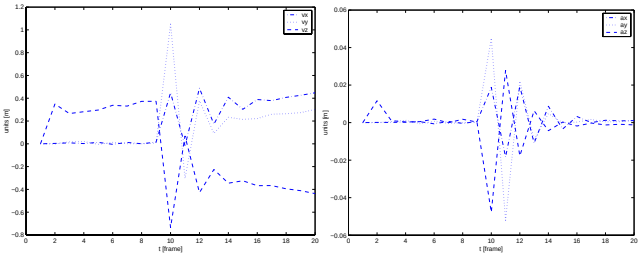


Figure 13: Prediction of angular velocities. The behavior of the EKF for frames 1 to 9 is similar to the previously carried out simulation. From frame 9 to 11 the prediction of the angular velocity is unstable.



(a) (b)

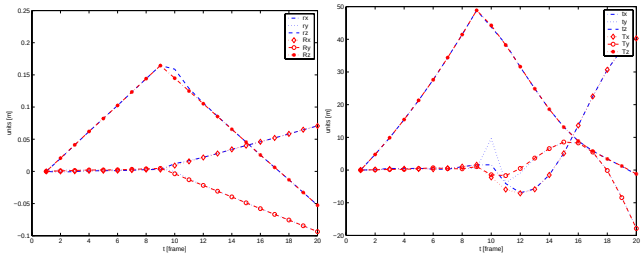
Figure 14: Prediction of translational velocity and acceleration. (a) Translational velocity (b) Translational acceleration. It can be seen that translational velocity and -acceleration are more sensitive to noise than angular velocities. The filter is stable only after frame 13.

The integration of the Kalman filter has made the optical tracker more reliable, since the predicted pose of a rigid body is used to solve NP complete matching problems. It speeds up our marker segmentation and thus, allows to achieve tracking with full frame rate of the cameras. The cameras we are using provide 30 frames per second so that in cases of very fast rigid body movements Shannon's theorem is broken. In order to track rigid bodies more reliably under such circumstances, we plan to combine the optical tracker with inertial sensors to create a hybrid tracking system which is able to share the strengths of each sensor technology and to fulfill the Shannon theorem in cases of very fast motion.

4. CONCLUSIONS

We have proposed an extended Kalman filter formulation for robust optical tracking. It has been illustrated how Kalman filtering enhances the reliability of an optical tracking system. A motion model based on exponential maps was proposed to track motion. It has its strengths in numerical robustness and fast convergence to the desired motion estimate. The implemented Kalman filter was evaluated with respect to different rigid body motions and provides filtering and precise prediction unless the Shannon theorem is fulfilled.

Our future work will compare our Kalman filter formulation of rigid body motion with formulations that treat rotation and translation independently from each other to predict motion. We want to show that our prediction method is more precise with respect to pixel error in the context of



(a) (b)

Figure 15: Overshoots of predicted rigid body rotation and translation. When calculating the global rotation (a) and global translation (b) of the rigid body using the state vector of the EKF, overshoots in frames 10 to 12 for (a) and in frames 9 to 13 for (b) can be recognized.

Augmented Reality applications, since our method reduces the displacement error of points in three-dimensional space.

Optical tracking is very precise in estimating the position of points in 3D space, but it suffers from line-of-sight problems. Also, if inexpensive standard cameras are used, the frequency of an optical tracker cannot be higher than the sensors' frequency. Since standard cameras provide sampling rates about only 30 Hz, a faster sensor technology would be welcome. The line-of-sight problem as well as a low sampling frequency can be overcome if inertial sensors are incorporated. Drift correction will be done with optical tracking and frequency enhancement and tracking in cases of occlusion will be done with inertial sensors. Sensor fusion will be done with an extension to our Kalman filter formulation.

5. REFERENCES

[1] Advanced realtime tracking. `http://www.ar-tracking.de`.

[2] S. Altmann. Hamilton, rodrigues, and the quaternion scandal. *Mathematics Magazine*, 62(3):291–308, 1989.

[3] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[4] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 17(6), 1995.

[5] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 15(6), 1993.

[6] R. Azuma. Predictive tracking for augmented reality. Technical Report TR95-007, UNC-Chapel Hill Department of Computer Science, February 1995.

[7] T. Broida and R. Chellapa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):90–98, 1986.

[8] T. Broida and R. Chellapa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):479–513, 1991.

[9] J. Chung, N. Kim, G. J. Kim, and C.-M. Park. Postrack: A low cost real-time motion tracking system for vr application. In *International conference on Virtual Systems and MultiMedia*, 2001.

[10] K. Dorfmüller. An optical tracking system for vr/ar-applications. In M. Gervautz, A. Hildebrand, and D. Schmalstieg, editors, *Virtual Environment'99*, pages 33–42. Springer Verlag, Wien, New York, 1999.

[11] K. Dorfmüller and H. Wirth. Real-time hand and head tracking for virtual environments using infrared beacons. In D. T. N. Magnenat-Thalmann, editor, *Modelling and Motion Capture Techniques for Virtual Environments*, volume 1537 of *Lecture Notes in Artificial Intelligence*, pages 113–127. Springer Verlag, Heidelberg, 1998.

[12] E. Foxlin. Motion tracking requirements and technologies. In K. M. Stanney, editor, *Handbook for Virtual Environments*, pages 163–210. Lawrence Erlbaum Associates Publishers, 2002.

[13] E. Foxlin, M.Harrington, and G. Pfeifer. Constellation^{TM}: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of ACM SIGGRAPH'98*, pages 371–378, Orlando, FL, July 1998.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[15] M.-H. Kim, S.-M. Choi, Y.-J. Choi, S.-M. Rhee, H.-R. Chung, and H.-S. Kim. Workbench vr - construction and application of a semi-immersive vr environment using the workbench. *Information Technology Research Center Forum*, May 2001.

[16] F. Madritsch and M. Gervautz. CCD-camera based optical beacon tracking for virtual and augmented reality. *Eurographics*, 15(3), 1996.

[17] J. Möltgen, B. Schmidt, and W. Kuhn. Landscape editing with knowledge-based measure deductions for ecological planning. In P. A. . A. Stefanidis, editor, *ISD'99 - Integrated Spatial Databases: Digital Images and GIS Lecture Notes in Computer Science, 1737*, pages 139–152. Springer, Berlin, 1999.

[18] J. D. Mulder and R. van Liere. The personal space station: Bringing interaction within reach. In *Proceedings of VRIC 2002*, Laval, June 2002.

[19] M. Ribo, A. Pinz, and A. Fuhrmann. A new optical tracking system for virtual and aug-mented reality applications. In *Proceedings of the IEEE Instrumentation and Measurement Technical Conference*, pages 1932–1936, 2001.

[20] R. T. Tsai. An efficient and accurate camera calibration technique for 3-d machine vision. *Proccedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1986.

[21] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[22] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information. *Computer*

Graphics, 31(Annual Conference Series):333–344, 1997.

[23] J. Weng, N. Ahuja, and T. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, 1993.

[24] J. Weng, T. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, 1989.

[25] G. Young and R. Chellappa. 3-d motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):735–759, 1990.

[26] Z. Zhang and O. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. Technical Report 1438, INRIA Sophia-Antipolis, France, July 1991.