Finger tracking for interaction in augmented environments

Klaus Dorfmüller-Ulhaas[*,+], Dieter Schmalstieg[+]
[*]Imagination Computer Services, Austria
[+]Vienna University of Technology, Austria
{klaus, dieter}@ims.tuwien.ac.at

Abstract

Optical tracking systems allow three-dimensional input for virtual environment applications with high precision and without annoying cables. Spontaneous and intuitive interaction is possible through gestures. In this paper, we present a finger tracker that allows gestural interaction and is simple, cheap, fast, robust against occlusion and accurate. It is based on a marked glove, a stereoscopic tracking system and a kinematic 3-d model of the human finger. Within our augmented reality application scenario, the user is able to grab, translate, rotate, and release objects in an intuitive way. We demonstrate our tracking system in an augmented reality chess game allowing a user to interact with virtual objects.

(a) (b)

Figure 1. Manipulation of virtual objects by grab and release gestures: Natural interaction is possible using the finger tracker described in this paper together with augmented reality displays. In this image, a user plays chess against the computer by moving virtual chess men with his finger on a real board.

1 Introduction

In order to convey a sense of immersion, a virtual environment (VE) system must not only present a convincing visual rendering of the simulated objects, but also allow to manipulate them in a fast, precise, and natural way. Rather than relying on mouse or keyboard, direct manipulation of virtual objects is enabled by employing tracking with six degrees of freedom (6DOF). Frequently, this is done via handheld props (such as flying mouse or wand) that are fitted with magnetic trackers. However, this technology can only offer limited quality because it is inherently tethered, inaccurate and susceptible to magnetic interference. Early on, optical tracking has been proposed as an alternative. One main reason why optical tracking is so attractive is because unlike tethered magnetic tracking it supports capturing human movement without the need for active sensors, thus allowing interaction without the use of props. In particular, the tracking of hands is relevant, because it allows natural gesturing (Figure 1).

Figure 2 shows a taxonomy of gesturing after Quek [25, 26]. Intentional gestures can be roughly categorized into *manipulative* (object movement, rotation etc.) and *communicative*. While the expressive power of gestures is mostly attributed to the communicative family of gestures, it is the manipulative family that is mostly used in virtual environments. The reason is that choreographing 3-d events based on object manipulation is straightforward and immediately useful, while the meaning of communicative gestures is often more subtle and harder to exploit in applications. Also, communicative gesturing just like any form of communication relies on a common language that first needs to be mastered by the user before useful interaction is possible (with the exception of deictic gestures [3]). We will now examine how gestures fit into an interaction framework for VEs. We follow the 3-d interaction taxonomies developed by Hand [11] and Bowman [4] that categorize interaction into viewpoint control, selection, manipulation, and system control:

- Viewpoint manipulation in virtual environments is best

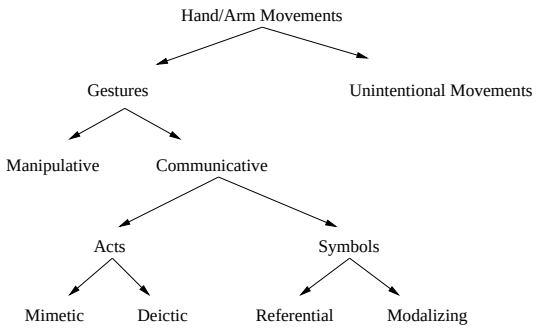


Figure 2. Intentional hand and arms movements can be classified as manipulative or communicative. Communicative gestures can be related to language (symbolic), or non-linguistic acts. Mimetic acts simulate actions, while deictic acts refer to a specific object. Symbolic gestures either stand for a referential action, or are used as modalizers, often for speech.

performed with direct head tracking [29].

- Manipulation of virtual objects (rigid objects, i. e., primarily translation and rotation) is a core requirement of most VE applications. Obviously, a very efficient method is direct manipulation with the hand.

- To perform selection in reality, a user stretches out the hand in the direction of the target object, then grabs it for manipulation, which can be tracked as a deictic followed by a mimetic gesture.

- System control describes all access to abstract functions that have no obvious correspondence in the three-dimensional environment. To provide good feedback, visible command objects (3D icons etc.) are often used for system control. Such system icons can be manipulated with similar gestures like normal objects.

Taken together, all input modes relevant for a general virtual environment can be provided by control of a 6DOF cursor and a grab/select command. In this paper, we propose to track the user's index finger via retroreflective markers and use the tip as a cursor. The select command is triggered by bending one's finger to indicate a grab or grab-and-hold (i.e., drag) operation. The simplicity of this approach is also its power. While the potential of optical tracking as a superior tracking technique is generally recognized, its complexity has prevented widespread acceptance. In contrast, our simple approach is at a sweet spot in the space of possible optical tracking approaches, allowing to develop a finger tracker that is fast, reliable, robust against occlusion, cheap, and accurate, and that can be interfaced easily to any VE and provides all necessary means of interaction through gestures. It combines natural and unobtrusive interaction through gesturing with precise and general purpose interaction in a mostly unrestrained virtual environment. Surprisingly, to our knowledge this particular approach has not been tried yet.

In the following, we discuss related work in section 2, followed by an overview of our approach in section 3, and details on the used finger model in section 4 and computer vision algorithms in section 5. The presentation is complemented by results in section 6 and section 7 concludes the paper.

2 Related Work

In this section, we give a brief overview of gesture based interaction methods which consider the human hand. As mentioned before, gestures may be classified as *manipulative* or *communicative*. Our overview of the literature will concentrate on manipulative gestures, since we are interested in systems which allow to grab, translate, rotate and release virtual objects. The interested reader is referred to Pavlovic *et al.* [14, 24] for a general survey of hand tracking methods and algorithms for hand gesture analysis.

Considering the complexity of shapes of the human hand which may appear in video images, the segmentation of the human hand can be figured out as the most crucial and time-consuming part a vision based system has to solve. In case of manipulative gestures, the tracking of the hand should operate in real-time. This is why system developers apply constraints either for the environment or the appearance of the human hand. We will distinguish *background* and *foreground constraints* generally applied for simplifying the segmentation process. *Background constraint systems* are often using a uniform (uncluttered) background [28, 5, 21, 18, 6]. Other systems assume a static or temporarily static background so that background subtraction [16, 32, 35] or segmentation by motion [19] can be performed. Unfortunately, using a controlled or known background is problematic or impossible in dynamic virtual and augmented environments where the scene changes over time. *Foreground constraint systems* detect markers attached to the human hand [7, 23, 22] or classify the human skin color [17, 10, 36]. Such systems assume controlled and static lighting conditions and rely on the assumption that no objects with similar color (e.g., skin/wood) appears in the image. Projection-based virtual environments are typically used with dimmed lights, leading to a decrease in color dynamics, which results in difficulties in identifying the human skin.

Template matching approaches for special hand features like the finger tips restrict the hand in its flexibility of defor-

mation since the finger tips should always be visible in the camera images [28, 10]. A common approach is to restrict the appearance of the hand to known depth values and to disallow other objects to appear inside the interaction volume [33]. Finally, an infrared camera system can be adapted to acquire optical signals at a controlled temperature for the human hand [30].

After image segmentation, the hand model plays a fundamental role in the tracking process. We distinguish em 3-d hand models and *appearance based models*. 3-d hand models use articulated structures of the human hand to estimate the hand movements [28, 21], whereas appearance-based models directly link the appearance of the hand movements in visual images to specific gestures [2, 12, 32]. 3-d hand model-based systems often provide a higher flexibility, due to the estimation of joint angles and a higher precision. Finally, the form of output from the tracking process determines the scope of possible applications. We classify 2-d systems [2], e.g., for controlling 2-d user interfaces [30], systems working in 3-d by supporting relative 3-d positions [23, 12] and systems which are using stereoscopic vision for most accurate, absolute 3-d positions [28, 32, 33]. Obviously, only absolute 3-d position is useful for our application scenario.

Often not addressed is the necessity of tracking initialization which means that the user is forced to move the hand to a known position while performing a specific pose. Systems like [2, 12, 28] need this initialization whenever the object detection algorithm looses track. Such an approach is not acceptable for spontaneous and natural interaction in virtual environments. Recently, a new algorithm called *Condenstation* [15] has been developed which tries to overcome the increasing uncertainty of a Kalman filter process. The algorithm is based on random sampling in order to track objects with a best fit over time, mostly independent from discrete time slots where the probability is not optimal. In the case of *Active Contours* it is necessary to collect alternative states and to prevent the system from loosing track, because a re-initialization of the system is computational expensive and not soluble in real-time. For the purpose of object tracking, we have implemented a Kalman filter, able to estimate a residual between the observed and estimated measurements. This residual value is used as a deciding factor whether the filter looses track or not.

3 System Overview

We did not find a human hand tracking system which fulfills all of our requirements. Specifically, all purely natural-feature-based tracking systems are either not accurate for the purpose of augmented reality or not independent from the environment or application. To overcome these problems our optical tracking consists of retroreflective markers operating with infrared light. The tracking system poses minimal constraints to the environment and can be easily adapted for other virtual reality applications. The proposed design is intended for a fixed working area of reasonable size (1-3m squared) where dextrous interaction can occur. A suitable workspace is defined by the volume above a table - this is both useful in combination with back-projection tables [20] and augmented reality scenarios [27, 13]. In the following we focus on an outside-in tracking system, since occlusion problems may not be such a big problem as for inside-out tracking and in addition a stereo rig with a higher baseline should be more precise for the purpose of 3-d interaction.

We want to require minimal effort in the setup and maintenance of the system. Thus it is a requirement that the system can be used without any special lighting or background. Moreover, a simple calibration procedure is necessary to allow quick installation of the system after the location or environment has changed. To allow for a relatively unrestrained environment, a marked glove is used for real-time separation of the finger from the background. The glove is fitted with retroreflective markers, which are illuminated by an infrared light source. A stereo camera pair with infrared lenses filters out most of the background. The infrared light source is co-located with the camera, so that light emitted in the direction of the retroreflective markers is directly reflected towards the camera in a fashion similar to [9]. After segmentation of the 2-d marker locations, they are passed on to the marker matching module, where markers are correlated using a method based on epipolar constraints and a kinematic model of the finger. A motion estimator has been added in order to smooth and predict the motion of the user's finger. Therefore, the synthesized 3-d position values are used as periodic measurements during a Kalman filter process. The filter itself takes parameters of a linearized kinematic model such as velocity, acceleration and angular velocities. These parameter values may be used in order to predict a future pose of the user's finger.

We will examine the used marker and finger model in more detail, and then discuss the relevant steps in computer vision processing required to transform images from the camera into 3-d model parameters.

4 Markers and finger model

The intention of the finger tracker is to determine enough information to robustly track the position, orientation and pose of the user's index finger. For real-time determination of these parameters without the need to constrain environmental conditions, we resort to using a marked (but untethered) glove. In the following, we describe our considerations regarding shape and placement of these markers.

Possible marker shapes are shown in figure 3. Round or

square reflector blips are features of the surface, which is fine as long as the markers face the camera. However, while interacting in virtual reality, hand and fingers are constantly rotated in space, and markers will often turn away from the camera. In this case, the blip would not indicate the real position of the joint any more.

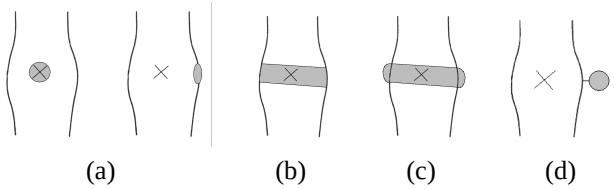


Figure 3. Shape of markers - in contrast to round blips on the surface (a) that do not always represent the joint position (cross) well if rotated away from the camera, flat rings (b) are always centered at the joint, while convex rings (c) improve upon flat rings in that they have better retroreflective properties. Our final choice are displaced balls (d) that suffer the least from self-occlusion of the fingers.

As an alternative solution, we tried ring-shaped markers composed from small stripes of reflector material that are wrapped around the finger joints. A section of the rings will always face the camera independent of the rotation of the joint. After some experimentation, the ring markers were modified to have a convex rather than a flat surface (Figure 4(a)). In that way, a portion of the retroreflective surface of the marker will always be oriented towards the camera, allowing for a higher amount of light to be reflected to the camera, thereby making segmentation easier. Unfortunately, our experiments showed that both blip and ring makers suffer from the fact that the joint center cannot easily be determined from the position of the markers due to self-occlusion of the fingers.

We therefore finally settled on using displaced balls on the back of the finger (Figure 4(b)), that have good retroreflective properties and are not often significantly occluded by the 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. This makes the glove independent of the user's finger lengths. Indeed, there is an offset between the markers and the real joint positions. The real kinematics can be estimated using the user dependent finger's segment lengths and thickness, however, our work is focused on a user independent and finger calibration free solution when estimating the pose of the chain of markers. 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. Our experiences confirmed that it does not affect finger movement or interaction in any noticeable way.

For reconstruction, we employ a 3-d finger model based on a kinematic chain of the finger joints 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 users finger tip to the last marker in the chain which is used to determine the 6DOF "hot spot". To enable a user to interact with his or her finger tip, this offset must be determined. However, we found that most users are willing to accept that the actual hot spot is offset by a small amount from their finger tip, and interaction is not affected.

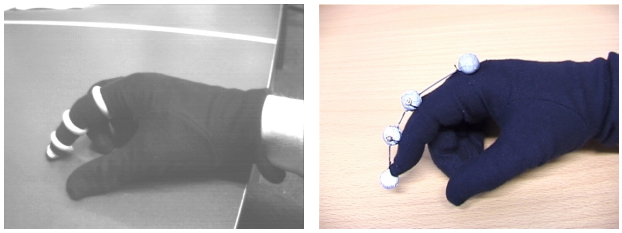


Figure 4. Gloves fitted with retroreflective markers

5 Computer vision processing

For performing the whole work cycle shown in figure 5, four tasks can be figured out which are the important operations of the tracking procedure. These are the calibration, the segmentation, the marker matching and the motion estimation which includes the prediction of the model. These operations will be described in the following sections.

5.1 Camera Calibration

The calibration must be very easy to perform, because virtual reality users typically are not computer vision experts. Therefore, our system can adaptively calibrate a stereo rig by tracking a single moving point acquired from each of the two cameras. As a result, the calibration data may be entered by just waving a passive reflective marker around. The parameters which are estimated by the calibration procedure are the focal lengths of both cameras, the translation and rotation from one to the other camera and the

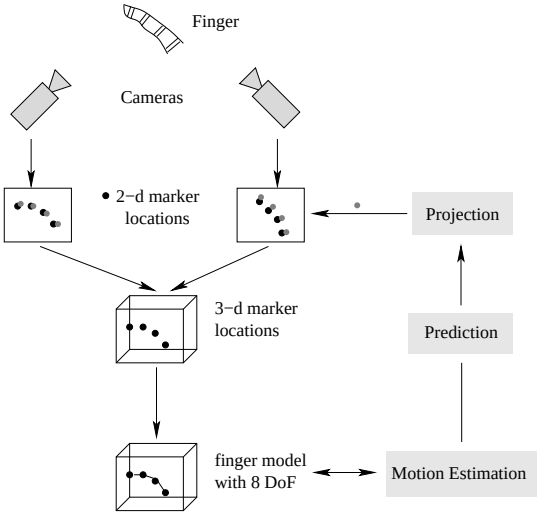


Figure 5. Processing pipeline

structure depth with regard to one camera coordinate frame. Given a rough estimate of these parameters, the system is able to estimate a global optimum by compairing a set of measurements of one camera with the transformed and projected measurements of the other camera on the first camera's image plane. Details about this easy-to-use calibration technique can be found in [8].

5.2 Segmentation

The principal task the segmentation process has to perform is the estimation of the center of gravity for each marker. The center of gravity is computed from the weighted contributions of the pixels covered by the markers in the greyscale image. We have implemented a threshold based segmentation, because it is simple and able to work in real-time. Pixel values which are above a given threshold are used to estimate the center of gravity of the marker image. This segmentation is not satisfying for all purposes as it is described above, but it works much faster than elliptic fitting algorithms. Later on, we try to compensate for these errors by using a Kalman filter for motion estimation.

Unlike ring markers, a spherical marker's center of gravity generally matches the joint center very well, which reduces uncertainty and improves the behavior of the Kalman filter described in section 5.4.

5.3 Matching of markers

In addition to the segmentation, we need a mechanism which correlates extracted features of both images. Due to reflections on specular surfaces, noise can be included in the list of segmented features and should be detected by the matching module.

Application of epipolar constraint does not solve the complete matching problem, which is problematic in cases where the corresponding feature for a marker in the first image is not the feature which has the closest distance to the epipolar line in the second image. This can lead to erratic matching that combines image features which are not correlated in reality. Since the epipolar constraint module can not detect such ambiguous cases based on the distance of a feature from the epipolar line, all matching features which lie within a small neighborhood of the epipolar line must be considered as candidates for 3-d points.

Detection of correct 3-d points and their assignment to finger joints is done by analysis of the 3-d position values we retrieve with the previously described uncertainty. By using knowledge about the distances between the markers on the user's finger and some further constraints, the system is able to estimate the finger's pose:

- The first constraint is based on the assumption that the marker positions are located approximately in one 3-d plane. While it is indeed possible to move a finger sideways to some degree, this constraint is sufficiently satisfied by the rigid marker skeleton.

- The second constraint is based on the non-ambiguous sequence of pre-known marker distances.

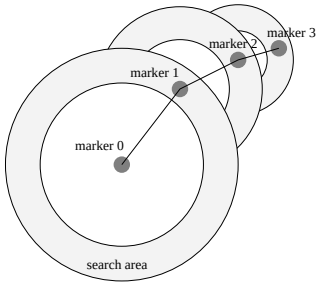


Figure 6. Marker matching

Figure 6 illustrates the procedure of marker matching. A random 3-d marker position is chosen. In the next step, the algorithm searches for a second marker position which has been located close to the surface of a sphere with a radius determined by the known marker distance. If no such marker position can be found, the algorithm starts with another arbitrarily chosen 3-d marker position. If a marker can be found close to the sphere's surface, a second sphere is used to find the third marker position and so on. The procedure is successful if a full path including four markers has been found, if the identified 3-d locations are located within a given threshold to a 3-d plane and if the shape of the polygon constructed from the joint positions is convex. One additional constraint which enhances the performance

of the system is based on knowledge retrieved from the motion prediction, which is described below in section 5.4. Consider the case when a complete path between the measurements could not be estimated in the presence of occlusion. Here, we derive the marker matching by the prediction of the Kalman filter process. The measurements close to the predicted values are taken as the new marker positions, while for occluded markers we may be able to use the predicted values. However, a better solution can be estimated using some biological constraints. If the finger tip marker or the marker furthermost to the fingertip is lost, we use a calculated marker position using the approximation that the first joint angle β_i close to the finger tip is two thirds of the second joint angle α_i. This constraint may also be used to estimate an inner marker position.

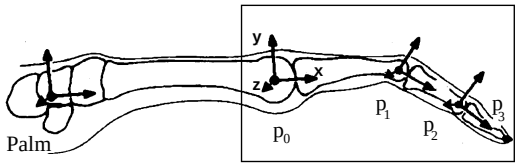


Figure 7. Finger coordinate system

For the following sections we assume a coordinate system defined by the finger pose (figure 7). The finger is located in the xy-plane and the origin is at point p_0, with the x-axis pointing in the direction of p_1. As common for kinematic chains, the coordinate system for p_1 is defined relative to the reference frame of p_0. Analogously, the reference system of p_2 is defined relative to p_1. p_3 is only necessary to define the direction of the x-axis in the reference frame of p_2. In case the user's finger is bent, the global rotation matrix of the finger at frame i can be calculated as follows

$$\boldsymbol{e}_{x,i} = \frac{\boldsymbol{p}_{1,i} - \boldsymbol{p}_{0,i}}{\|\boldsymbol{p}_{1,i} - \boldsymbol{p}_{0,i}\|} \quad (1)$$

$$\boldsymbol{e}_{z,i} = \frac{\left[\boldsymbol{p}_{3,i} - \boldsymbol{p}_{0,i}\right] \times \boldsymbol{e}_{x,i}}{\| \left[\boldsymbol{p}_{3,i} - \boldsymbol{p}_{0,i}\right] \times \boldsymbol{e}_{x,i}\|} \quad (2)$$

$$\boldsymbol{e}_{y,i} = \boldsymbol{e}_{z,i} \times \boldsymbol{e}_{x,i} \quad (3)$$

First, we calculate $e_{x,i}$ as the norm of the vector from $p_{0,i}$ to $p_{1,i}$. Since all markers should lie on a plane, we can use $p_{3,i}$ to define a second vector used to compute the y- and z-axis of this coordinate system by applying the cross product of vectors. The result gives the base vectors of the global finger reference frame which can be combined in a global rotation matrix at time frame i. The vectors $e_{x,i}$,$e_{y,i}$, and $e_{z,i}$ form the columns of the matrix.

$$R_i = \begin{pmatrix} \boldsymbol{e}_{x,i} & \boldsymbol{e}_{y,i} & \boldsymbol{e}_{z,i} \end{pmatrix} \quad (4)$$

5.4 Modeling and Estimating Motion Kinematics

For developing a robust finger tracker it is important to achieve good estimates of the finger pose, even though measurements are imprecise and include distortions. Measurements such as the marker positions are assumed to contain white noise. The Kalman filter used in our implementation is responsible for filtering the motion model parameters values. Whenever the system equations[1] do not fit the real motion process well, the residual between real motion and motion model will be interpreted as random system noise. The Kalman filter as used in our implementation is rather a filter which extracts a kinematic state from periodic noisy measurements than a predictor of future marker positions used for speeding up the segmentation. Our implementation is using the Kalman filter in order to enhance the finger pose matching for the current frame. The process of marker and finger pose matching consists of a minimal path search of estimated 3-d point distances and is known to be NP-complete. Searching only a small number of markers does not really suffer from this fact, but even a moderate number of falsely detected marker positions (reflections etc.) can quickly affect computational performance of the search. The Kalman filter is a good tool to overcome this problem by predicting new 3-d marker positions. Based on this prediction the algorithm can directly select markers in locations likely to contain valid 3-d points. As mentioned before and shown in figure 7, our measurement vector $\boldsymbol{x}_i$ at time frame i includes the location of four 3-d marker positions.

$$\boldsymbol{x}_i = \begin{pmatrix} \boldsymbol{p}_{0,i} & \boldsymbol{p}_{1,i} & \boldsymbol{p}_{2,i} & \boldsymbol{p}_{3,i} \end{pmatrix} \quad (5)$$

These measurements are not correct due to noise from calibration and segmentation errors. We assume this noise is white noise $\boldsymbol{\eta}_i$ added to the correct measurement $\boldsymbol{x'}_i$.

$$\boldsymbol{x}_i = \boldsymbol{x'}_i + \boldsymbol{\eta}_i \quad (6)$$

Consider figure 8 for the transformation of marker positions $\boldsymbol{p}_0$,$\boldsymbol{p}_1$,$\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ from time frame $i-1$ to time frame i. For each point of the marker model a translation T_{i-1} and rotation R_{i-1} is performed. This incremental and relative rotation is modeled using angular velocities $\boldsymbol{\omega} = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$. We are applying equation 7

$$q = \frac{\omega_x}{2}i + \frac{\omega_y}{2}j + \frac{\omega_z}{2}k + \sqrt{1 - \frac{\omega_x^2 + \omega_y^2 + \omega_z^2}{4}} \quad (7)$$

introduced by Azarbayejani and Pentland [1] to transform the angular velocities into a quaternion representation of the rotation R_{i-1}. For the translational as well as for the rotational components it is assumed that each point $\boldsymbol{p}_{j,i-1}$, $j := [1..4]$ undergoes a motion with constant angular velocity and with constant translational acceleration. In other

[1] In our case the motion kinematic equations.

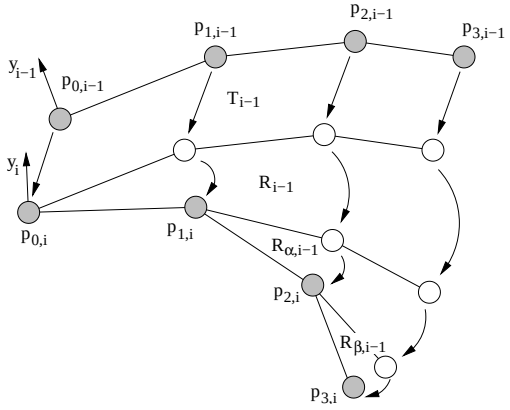


Figure 8. Marker transformation

words we use a linearized kinematic model for motion estimation, which is simple and less computationally intensive than using the accurate model. However, this linearization is only effective for a short period in time. Therefore, realtime motion capturing is neccessary and the precision decreases with the frame rate. Using this linearization, the translation T_{i-1} can be expressed as

$$T_{i-1} = \boldsymbol{v}_{i-1}\,\Delta t + \frac{1}{2}\boldsymbol{a}_{i-1}\,\Delta t^2 \tag{8}$$

where $\boldsymbol{v} = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$ is the translational velocity, $\boldsymbol{a} = \begin{pmatrix} a_x & a_y & a_z \end{pmatrix}^T$ is the constant translational acceleration and Δt is the time interval $t_i - t_{i-1}$. To estimate the finger's motion kinematics, the bending of joints has been modeled by applying a rotation $R_{\alpha,i-1}$ for the first joint and $R_{\beta,i-1}$ for the second joint. $R_{\alpha,i-1}$ is defined as a rotation around the z-axis using the angle α_{i-1}:

$$R_{\alpha,i-1} = \begin{pmatrix} \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{9}$$

The rotation $R_{\beta,i-1}$ is defined similar.

Consider once again the incremental transformation shown in figure 8, where the rotation depends on the angular velocity and the translation depends on the translational velocity and translational acceleration as described before. As we assume to have a linearized motion, we are able to calculate the new marker positions if we know the following parameters collected in the state vector

$$\boldsymbol{s}_i = \begin{pmatrix} \boldsymbol{v}_i & \boldsymbol{a}_i & \boldsymbol{\omega}_i & \alpha_i & \beta_i \end{pmatrix}^T \tag{10}$$

These parameters are estimated during a Kalman filter process. Thus, we are using the "hat"(ˆ)-notation for estimated parameters and are able to express the marker movements by the following equation:

$$\hat{\boldsymbol{p}}_{0,i} = \boldsymbol{p}_{0,i-1} + \hat{\boldsymbol{v}}\,\Delta t + \frac{1}{2}\hat{\boldsymbol{a}}\,\Delta t^2 \tag{11}$$

$$\hat{\boldsymbol{p}}_{1,i} = \hat{\boldsymbol{p}}_{0,i} + \hat{R}_i \boldsymbol{p}_{1,i-1} \tag{12}$$

$$\hat{\boldsymbol{p}}_{2,i} = \hat{\boldsymbol{p}}_{1,i} + \hat{R}_i \hat{R}_{\alpha,i} \left(\boldsymbol{p}_{2,i-1} - \boldsymbol{p}_{1,i-1}\right) \tag{13}$$

$$\hat{\boldsymbol{p}}_{3,i} = \hat{\boldsymbol{p}}_{2,i} + \hat{R}_i \hat{R}_{\alpha,i} \hat{R}_{\beta,i} \left(\boldsymbol{p}_{3,i-1} - \boldsymbol{p}_{2,i-1}\right) \tag{14}$$

These equations can be seen as an estimation process of future measurements at time frame i while previous measurements given at time frame $i-1$ are known:

$$\boldsymbol{x}_i = \begin{pmatrix} \boldsymbol{p}_{0,i} & \boldsymbol{p}_{1,i} & \boldsymbol{p}_{2,i} & \boldsymbol{p}_{3,i} \end{pmatrix}^T \tag{15}$$

Whenever a new measurement is available, the Kalman filter is performing a correction step (also called *measurement update*) to keep the residual between measurements and estimated measurements as low as possible by minimizing the error using a least square approach. The function $f\left(\boldsymbol{x'}_i, \hat{\boldsymbol{s}}_{i|i-1}\right)$ which is dependent on the current estimated state and the last measurement vector should be minimized and is given in equation 16.

$$f\left(\boldsymbol{x'}_i, \hat{\boldsymbol{s}}_{i|i-1}\right) = \begin{pmatrix} \boldsymbol{p'}_0 - \hat{\boldsymbol{p}}_{0,i} \\ \boldsymbol{p'}_1 - \hat{\boldsymbol{p}}_{1,i} \\ \boldsymbol{p'}_2 - \hat{\boldsymbol{p}}_{2,i} \\ \boldsymbol{p'}_3 - \hat{\boldsymbol{p}}_{3,i} \end{pmatrix} = \boldsymbol{0} \tag{16}$$

After *measurement update* a new prediction can be performed. This step is also called *time update*, because this procedure is projecting the current state forward in time. Considering our application context, a linear transformation of the state vector is applied which is given by:

$$\begin{aligned} \hat{\boldsymbol{v}}_{i|i-1} &= \hat{\boldsymbol{v}}_{i-1} + \hat{\boldsymbol{a}}_{i-1}\Delta t \\ \hat{\boldsymbol{a}}_{i|i-1} &= \hat{\boldsymbol{a}}_{i-1} \\ \hat{\boldsymbol{\omega}}_{i|i-1} &= \hat{\boldsymbol{\omega}}_{i-1} \\ \hat{\boldsymbol{\alpha}}_{i|i-1} &= \hat{\boldsymbol{\alpha}}_{i-1} \\ \hat{\boldsymbol{\beta}}_{i|i-1} &= \hat{\boldsymbol{\beta}}_{i-1} \end{aligned}$$

The strength of the Kalman filter is its feasibility to model noise, even allowing the system to filter state values in noisy environments. The existence of noise is assumed for two different processes.

- The measurement includes a white noise such that the expectation value is zero $E(\boldsymbol{\eta}_i) = 0$ and noise included in one measurement is independent from noise of another measurement.

$$E(\boldsymbol{\eta}_i \boldsymbol{\eta}_j^T) = \begin{cases} \Lambda_{\boldsymbol{\eta}_i} & i = j \\ 0 & i \neq j \end{cases} \tag{17}$$

$\Lambda_{\boldsymbol{\eta}_i}$ describes the covariance matrix of measurement noise at time frame i.

- The filter models system noise that results from imprecise system equations. For instance, the linearized kinematic motion model is not describing the real motion. Thus, there is a difference between the linearized and the real motion which can be modeled as a white system noise similar to equation 17. We denote the covariance matrix of system noise Q_i.

Figure 9 shows the complete extended Kalman filter process as applied for finger tracking purposes.

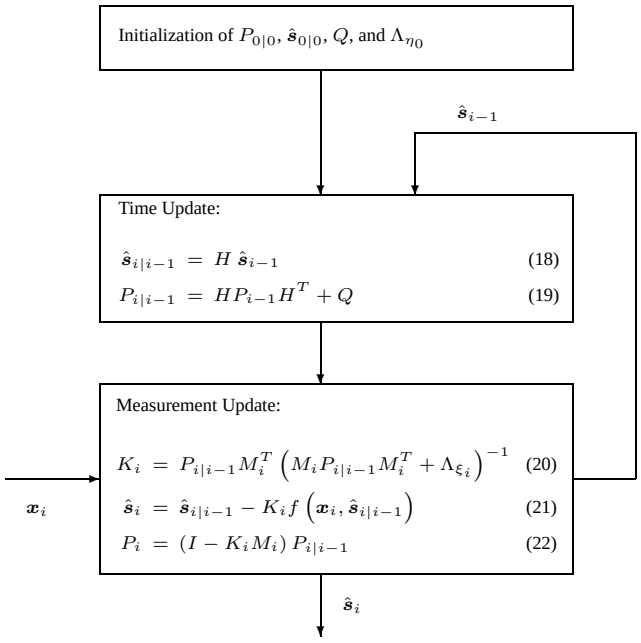


Figure 9. The extended Kalman filter

As a first step, an initialization of the filter is necessary. Therefore, the covariance matrix of the state vector $P_{0|0}$, the state vector itself, the system and measurement noise matrices need to be specified. Afterwards, the state vector and its covariance matrix can be projected forward in time using:

$$H = \begin{pmatrix} I_3 & \Delta t\, I_3 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & I_2 \end{pmatrix} \tag{23}$$

The next step is to correct the state and covariance matrix P_i whenever a new measurement is available. Therefore, the Kalman gain matrix K_i is calculated which is used as a relative weighting of the trust in real measurements vs. the estimated system state. Since equation 16 is non-linear, we have to apply the extended Kalman filter, which requires calculation of the Jacobian matrix M_i

$$M_i = \frac{\partial f\left(\boldsymbol{x}_i, \hat{\boldsymbol{s}}_{i|i-1}\right)}{\partial \boldsymbol{s}_i} \tag{24}$$

and the new measurement noise matrix Λ_{ξ_i} which is influenced by the derivative of the function $f\left(\boldsymbol{x}_i, \hat{\boldsymbol{s}}_{i|i-1}\right)$.

$$\Lambda_{\xi_i} = \frac{\partial f\left(\boldsymbol{x}_i, \hat{\boldsymbol{s}}_{i|i-1}\right)}{\partial \boldsymbol{x'}_i}\Lambda_{\eta_i}\frac{\partial f\left(\boldsymbol{x}_i, \hat{\boldsymbol{s}}_{i|i-1}\right)}{\partial \boldsymbol{x'}_i}^T \tag{25}$$

6 Experimental results

Experiments with real sequences of marker based finger motions were done on an Athlon 800 MHz processor using ELTEC's PcEye2 frame grabber board and two PULNiX TM-560 PAL cameras. The finger tracking operates in real-time with 25 frames per second[2] and an accuracy of 0.5 to 2 mm in the range of one square meter. The angular accuracy is difficult to analyse because it is dependent on the bending of the user's finger. Analysing the jittering in rotational values while having a bent finger, the angular error is below one degree in average.

We have connected the tracking system via sockets with the *Studierstube* [31] augmented reality system. The latency of the whole system is about 50 to 100 ms. We can compensate this latency while using predicted marker positions. However, the accuracy of the system is reduced to 5 mm precision while predicting 80 ms forward in time.

The application we have used for rendering is a virtual chess application where chess men are displayed as virtual objects and the chess board is real. In order to grab a virtual chess man the user has to move his finger to the middle of one square and intersect the marker located at the finger tip with the virtual chess man and bend the finger in order to grab the virtual object. While holding the finger bent, the user is able to drag (translate and rotate) the chess man and release it by stretching out the finger. This kind of interaction was found to be intuitive and easy to learn because it is similar to a real grab gesture the user performs. Compare figure 1(a) for an image of the collision of the user‘s finger with a chess man and figure 1(b) for an image while performing the grab gesture and dragging the virtual object. During fusion of real images and virtual images there is one thing that is not perfect in regard to a fully immersive illusion of the user, which are incorrect placements of the virtual objects in regard to the real objects like the human hand. Considering figure 10, the grabbed chess man should be located at the finger tip, but it appears on the palm. Future augmented reality systems should handle occlusions of virtual objects. This may be solved by estimating depth values of the real scene, however, this is a time-consuming

[2]The frame rate is limited by the update rate of the camera.

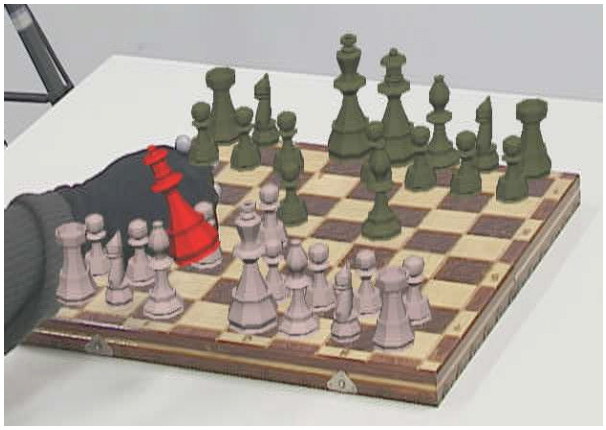
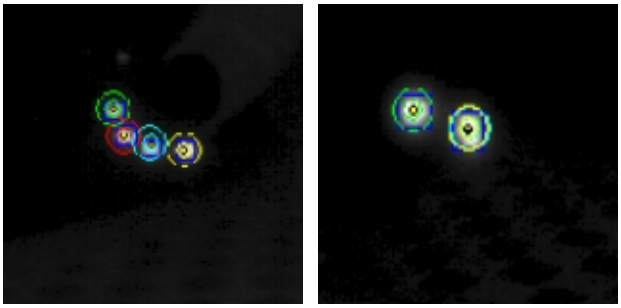


Figure 10. Fusion of the real and virtual world

reconstruction problem, which is currently not solvable in real-time. A video of the finger tracking in combination with a chess application can be downloaded from http://www.ims.tuwien.ac.at/pages/research/vr/fingertracker/.

In regard to the robustness of the tracking, a source of problems of vision-based tracking systems is occlusion. In our tracking environment the cameras are positioned more or less orthogonal to each other. Thus, there is no need for our tracking system to detect four markers in each camera image (see figure 11). We are able to estimate the finger



(left camera image) (right camera image)

Figure 11. Occlusions of markers

pose if one marker in each camera image plane is lost, or if two markers in one of the images are invisible. With our finger tracking system the interaction with virtual environments is performed wireless and more intuitive than with most other devices commercially available. One marker which is transiently lost causes no dramatic problems, but to allow two handed interactions more cameras are needed to be robust with regard to occlusions.

7 Conclusion and Future Work

We have presented an optical finger tracking system with high precision for three-dimensional input in virtual environments. It allows spontaneous and intuitive interaction through gestures and is simple, cheap, fast and robust against occlusions. The proposed tracking system does not need any initialization for tracking and there is no need to adapt the finger model to different users, since we are using a "exo-skeleton" model fixed to a glove. The system is operating in a relatively unrestrained environment.

Until know, we have not done any evaluation with regard to ergonomics and comfort. However, from a practical point of view the marked glove fits to different users as long as the cotton glove suits the user's hand. The "exo-skeleton" seems to disturb the user only in situations in which the sphere markers collide with other real objects. Therefore, the "exo-skeleton" should be designed using smaller sphere markers than in our current implementation. The tracking method is expandable for two hand user input, but since occlusion is a well-known computer vision problem, multiple cameras will be necessary to solve hand-hand occlusions. Future plans include investigating how to track the contours of the human hand without having restrictions about the complexity of the background.

8 Acknowledgements

This research is supported by the European Community under contract no. FMRX-CT-96-0036. We would like to thank the members of the interactive media systems group at the Vienna University of Technology and in particular Gerhard Reitmayr for his support in creating an augmented video using the *Studierstube* system.

References

[1] A. Azarbayejani and A. P. Pentland *Recursive Estimation of Motion, Structure, and Focal Length,* IEEE PAMI 17(6), June 1995

[2] A. Blake and M. Isard *3D position, attitude and shape input using video tracking of hands and lips* In Proc. of SIGGRAPH'94

[3] R. A. Bolt *"Put-That-There": Voice and gesture at the graphics interface* Computer Graphics (SIGGRAPH '80 Proceedings), Vol. 14, No. 3. July, 1980, pp. 262 - 270

[4] D. Bowman (Ed.) *3D User Interface Design: Fundamental Techniques, Theory, and Practice* SIGGRAPH 2000 course notes No. 36, New Orleans, ACM Press, August 2000

[5] T. Brown and R.C. Thomas *Finger tracking for the Digital Desk* First Australasian User Interface Conference, AUIC 2000, 1999, pp. 11 - 16

[6] R. Cipolla and N.J. Hollinghurst *A human-robot interface using pointing with uncalibrated stereo vision* In Computer Vision

for Human-Machine Interaction, R. Chipolla and A. Pentland (Eds.), Cambridge University Press, 1988

[7] R. Cipolla, Y. Okamoto, and Y. Kuno *Robust structure from motion using motion parallax* In Proc. of the Fourth International Conference on Computer Vision, 1999, April 1993, pp. 374 - 382

[8] K. Dorfmüller and H. Wirth *Real-Time Hand and Head Tracking for Virtual Environments Using Infrared Beacons* In: N. Magnenat-Thalmann, D. Thalmann (Eds.) "Modelling and Motion Capture Techniques for Virtual Environments", International Workshop, CAPTECH'98, Geneva, Switzerland, November 1998, Proceedings LNAI 1537, Heidelberg: Springer Verl., 1998

[9] K. Dorfmüller *An optical tracking system for VR/AR-Applications* In Proc. of the Eurographics Workshop on Virtual Environment'99, Springer-Verlag Wien NewYork, Vienna, Austria, 1999, pp. 33 - 42

[10] R. O'Hagan, A. Zelinsky *Visual gesture interfaces for virtual environments* First Australasian User Interface Conference, 2000. AUIC 2000, 1999, pp. 73 -80

[11] C. Hand *A Survey of* 3*D Interaction Techniques* Computer Graphics Forum, Vol. 16, No. 5, 1997, pp. 269–281

[12] T. Heap and D. Hogg *Towards 3-D Hand Tracking using a Deformable Model*, 2nd International Face and Gesture Recognition Conference, 1996

[13] N. Hedley, L. Postner, R. May, M. Billinghurst, and H. Kato *Collaborative AR for Geographic Visualization* Proc. Int'l Symposium on Mixed Reality, Yokohama, Japan, March 2001, to appear

[14] T.S. Huang and V.I. Pavlovic *Hand gesture modeling, analysis, and synthesis* In Proc. of IEEE International Workshop on Automatic Face and Gesture Recognition, Sept. 1995, pp. 73 - 79

[15] M. Isard and A. Blake *Condensation - conditional density propagation for visual tracking* Int. J. Computer Vision, 1998

[16] K. Ishibuchi, H. Takemutra, and F. Kishino *Real Time Hand Gesture Recognition using 3D Prediction Model* International Conference on Systems, Man, and Cybernetics, Vol. 5, Le Touquet, France, Oct. 1993, pp. 324 - 328

[17] C. Jennings *Robust finger tracking with multiple cameras* In Proc. of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999, pp. 152 - 160

[18] A. Katkere, E. Hunter, D. Kuramura, J. Schlenzig, S. Moezzi, and R. Jain *ROBOGEST: Telepresence using Hand Gestures* Technical report VCL-94-104, Visual Computing Laboratory, University of California, San Diego, Dec. 1994

[19] I.J. Ko and H.I. Choi *Extracting the hand region with the aid of a tracking facility* Electronic letters32(17), August 1996, pp. 1561 - 1563

[20] W. Krueger, C. A. Bohn, B. Froehlich, H. Schueth, W. Strauss, and G. Wesche *The Responsive Workbench: A Virtual Work Environment* IEEE Computer, 28(7), 1995, pp. 42-48

[21] J.J. Kuch, T.S. Huang *Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration* Proc. of the Fifth International Conference on Computer Vision, 1995, pp. 666 - 671

[22] F. Lathuiliere and J.- Y. Herve *Visual hand posture tracking in a gripper guiding application* In Proc. of the IEEE International Conference on Robotics and Automation, ICRA '00, Vol. 2 , 2000, pp. 1688 - 1694

[23] C. Maggioni *A novel gestural input device for virtual reality* Virtual Reality Annual International Symposium, IEEE, 1993, pp. 118 - 124

[24] V.I. Pavlovic, R. Sharma, and T.S. Huang *Visual interpretation of hand gestures for human-computer interaction: a review* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7 , July 1997, pp. 677 -695

[25] F.K.H. Quek *Toward a Vision-Based Hand Gesture Interface* Virtual Reality Software and Technology Conf., Aug. 1994, pp. 17 - 31

[26] F.K.H. Quek *Eyes in the Interface* Image and Vision Computing, vol. 13, Aug. 1995

[27] R. Raskar, G. Welch, and W.Chen *Tabletop Spatially Augmented Reality: Bringing Physical Models to Life using Projected Imagery* Second Int. Workshop on Augmented Reality (IWAR'99), San Francisco, October 1999

[28] J. Rehg and T. Kanade *Digiteyes: Vision-based Human Hand Tracking*, Technical Report CMU-CS-TR-93-220, Carnegie Mellon University, 1993

[29] M. Usoh, K. Arthur, M. Whitton, R. Bastos, A. Steed, M. Slater, F. Brooks Jr. *Walking ¿ Walking-in-Place ¿ Flying, in Virtual Environments* Siggraph 1999, Computer Graphics Proceedings, Annual Conference Series, Addison Wesley Longman, Los Angeles, 1999, pp. 359 - 364

[30] Y. Sato, Y. Kobayashi, and H. Koike *Fast tracking of hands and fingertips in infrared images for augmented desk interface* In Proc. of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 462 - 467

[31] D. Schmalstieg, A. Fuhrmann, and G. Hesina *Bridging Multiple User Interface Dimensions with Augmented Reality* Proceedings of the 3rd International Symposium on Augmented Reality (ISAR 2000), pp. 20-30, Munich, Germany, Oct. 5-6, 2000

[32] J. Segen and S. Kumar *Human-computer interaction using gesture recognition and 3D hand tracking* In Proc. of the International Conference on Image Processing, Vol. 3, ICIP 98, 1998, pp. 188 - 192

[33] A. Utsumi and J. Ohya *Multiple-hand-gesture tracking using multiple cameras* IEEE Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 1999

[34] C. Ware and S. Osborne *Exploration and virtual camera control in virtual three dimensional environments* Proceedings of the 1990 Symposium on Interactive 3D Graphics (Snowbird, Utah). In Computer Graphics Vol.24, No. 2, March 1990, pp.175-183

[35] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland. *Pfinder: Real-time Tracking of the Human Body*, Integration Issues in Large Commercial Media Delivery Systems. A. G. Tescher and V. M. Bove, 1996

[36] A. Wu, M. Shah, N. Da Vitoria Lobo *A virtual 3D blackboard: 3D finger tracking using a single camera* In Proc. of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 536 - 543

[37] Z. Zhang and O. Faugeras *3-D Dynamic Scene Analysis*, Springer Series in Information Sciences, Springer-Verlag, Berlin 1992