Eurographics Symposium on Virtual Environments (2004)
S. Coquillart, M. Göbel (Editors)

Optical Tracking using Line Pencil Fiducials

Arjen van Rhijn and Jurriaan D. Mulder

Center for Mathematics and Computer Science, CWI, Amsterdam, the Netherlands

Abstract

In this paper, a new pattern based optical tracking method is presented for the recognition and pose estimation of input devices for virtual or augmented reality environments. The method is based on pencils of line fiducials, which reduces occlusion problems and allows for single camera pattern recognition and orientation estimation. Pattern recognition is accomplished using a projective invariant property of line pencils: the cross ratio. Orientation is derived from single camera line-plane correspondences, and position estimation is achieved using multiple cameras. The method is evaluated against a related point based tracking approach. Results show our method has lower latency and comparable accuracy, and is less sensitive to occlusion.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Tracking I.5.5 [Pattern Recognition]: Interactive systems I.3.7 [Computer Graphics]: Virtual Reality

1. Introduction

In virtual environments, accurate, fast and robust tracking of a user's actions is essential for smooth interaction with the system. Many different tracking systems are available, based on for instance mechanical, electromagnetical, and acoustical technologies. However, most of these methods hamper the user in manipulating input devices, due to wires, heavy sensors on input devices, or movement constraints. Optical tracking is a promising technology for VR/AR systems, since due to its lightweight and wireless nature, it allows for almost unhampered operation.

Optical tracking can be divided into two categories: *vision based*, where a scene is analyzed for known features by advanced image processing techniques, and *feature based*, where the features are artificially added to the scene in advance to simplify image processing. Although vision based tracking is more generic, feature based tracking is often preferred, since it generally results in lower computational requirements and as a result has lower latency.

A common approach for feature based optical tracking for VR is to augment one or more input devices with fiducials. Image processing is simplified to searching the camera images for these specific fiducial features, resulting in a list of 2D features for each camera. The features of these fiducials are modeled in 3D for each input device, and stored in a model database. The model based tracking problem is defined as determining the geometric transformation, such that a given device model is mapped onto the 2D features in the camera images.

We divide model based tracking into two stages: *recognition* and *pose estimation*. Recognition involves determining the correspondence between the 2D image features and the 3D features of the device models. This correspondence can be derived using only 2D information, or by including 3D information. For recognition, the correspondence between groups of 2D image features and the associated model is sufficient (*pattern correspondence*). For pose estimation, the exact correspondence between the 2D image features and the features of the accociated model is needed (*feature correspondence*). A common approach is to apply point shaped fiducials to input devices and to use stereo geometry to calculate 3D feature positions, enabling recognition in 3D. A pose estimate (position and orientation) is obtained for each input device by using information from the recognition stage.

A major disadvantage and intrisic problem of optical tracking is that it requires line of sight to operate. When a user handles an input device, fiducials possibly get obscured, e.g. by the user's fingers, other input devices, or obstacles in the environment. In many optical tracking systems, this leads to failure to track the user's actions.

In this paper we present an optical tracking strategy to reduce problems arising from occlusion. Our approach consists of the recognition of patterns consisting of line fiducials, in contrast to the more commonly used point fiducials. Each pattern consists of a *pencil*, i.e. four lines intersecting in one point. Input devices are augmented by one or more of these pencils. Figure 1 shows an example of a cubic-shaped input device with line pencil fiducials. Recognition

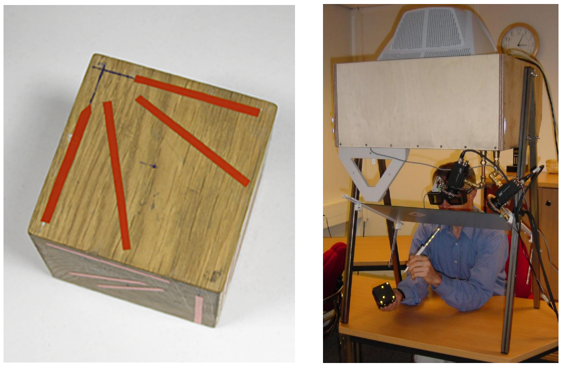

Figure 1: *(left) An example 7x7x7cm input device with line pencil fiducials. (right) A prototype Personal Space Station.*

is accomplished using a projective invariant property of line pencils, the cross ratio, and operates completely in 2D. Orientation is derived from single camera line-to-plane correspondences, and position is estimated from multiple camera images.

We have implemented and evaluated our optical tracking method using the Personal Space Station (PSS), a near-field desktop VR/AR environment [MvL02] (see Figure 1). The PSS enables a user to interact directly with the environment, using tangible 3D input devices. The s etup consists of cameras equipped with infrared filters and a ring of infrared LEDs illuminating the scene. Input devices are equipped with retroreflective fiducials, reflecting IR light back into the cameras.

This paper is organized as follows. In Section 2 we review related work. Section 3 describes the concepts on which our method is based: the cross ratio of line pencils and line-to-plane correspondences. Section 4 describes the method we use for recognition and pose estimation. Section 5 presents the results of an evaluation of our optical tracking method, comparing accuracy and latency with a related point based optical tracking method. Section 6 provides a discussion of the results and an analysis of the advantages and disadvantages of our method compared to previous approaches. Section 7 provides conclusions.

2. Related Work

In computer vision literature, various methods for model based object recognition and pose estimation have been proposed. In this section, we focus on approaches used for optical tracking in VR.

Dorfmüller [Dor99] suggests a distance fitting method for device recognition and pose estimation, which operates in 3D. He uses retroreflective point shaped fiducials on each input device, which are easily detected as 2D blobs in the camera images. Next, all possible 3D positions are calculated using stereo geometry. A model of a triangle is then fitted to these 3D blob positions using the 3D distances. Ribo et al. [RPF01] have followed the same approach.

Recently, van Liere et al. [vLM03] have applied the use of projective invariant properties to optical tracking. They use the cross ratio of point patterns for recognition. The cross ratio remains constant under perspective transformations, and thus recognition is performed completely in 2D. Patterns consist of either 4 colinear or 5 coplanar points. Each device is augmented with one or more patterns, which are modeled and stored in a database. After recognition, pattern correspondence is known. To determine feature correspondence, stereo geometry is used to calculate the 3D positions of the pattern points. Using a distance fit, the feature correspondence is established, and the final pose can be estimated.

There are several commercial optical tracking systems available, such as Optotrak [OPT], AR-track [ART], and Dynasight [DYN].

A problem with the aforementioned optical tracking methods is that they are very sensitive to occlusion. If a point is not visible, the input device often cannot be recognized. Algorithms as Iterative Closest Point (ICP) [BM92] and Geometric hashing [LW88] are able to handle missing points and are better suited to handle occlusion. However, geometric hashing generally has large memory requirements and can be too computationally expensive to satisfy the requirements of a practical optical tracking system, whereas ICP relies on a good initial estimate for the device pose.

In this work, we present a practical optical tracking system, based on projective invariant properties of line pencil patterns. The recognition stage determines the feature correspondence, and operates completely in 2D. Since the feature correspondence is known after recognition, it is possible to use a single camera orientation estimation approach by using line-to-plane correspondences. The final pose estimate is derived by combining the results from multiple cameras.

3. Concepts

In this section, we introduce two concepts on which our optical tracking method is based: The cross ratio of line pencils, which is used for the recognition stage, and line-to-plane correspondences, which are used for single camera orientation estimation.

3.1. Cross ratio of line pencils

Projective geometry preserves neither distances nor ratios of distances. However, the cross ratio [FP02], which is a ratio of ratios of distances, remains constant under projective transformations, and can therefore be used to solve the recognition problem in 2D. The cross ratio of a pencil of four lines is dual to the cross ratio of four colinear points. The cross ratio of colinear points A,B,C and D (see Figure 2) is defined as

$$\frac{|AB|/|BD|}{|AC|/|CD|} \tag{1}$$

where $|AB|$ denotes the Euclidian distance between points A and B.

The cross ratio depends on the ordering of the points. Four points can be chosen in $4! = 24$ ways, but in fact the cross ratio function produces only 6 distinct values. Meer et al. [MLR98] introduced a projective and permutation p^2-invariant, which obtains a representation of point sets that is insensitive to both projective transformations and permutations. This allows us to calculate the same cross ratio of 4 colinear points for any permutation.

From Figure 2 it can be derived that the cross ratio of four points translates directly to a pencil of four lines. It can be

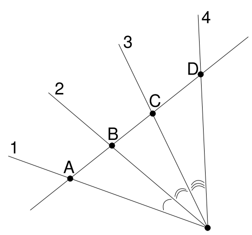

Figure 2: *The cross ratio of four colinear points is dual to the cross ratio of four intersecting lines.*

shown that Equation 1 can be manipulated into the following equation for the cross ratio of a pencil

$$\frac{\sin\theta_{12}/\sin\theta_{24}}{\sin\theta_{13}/\sin\theta_{34}} \tag{2}$$

where θ_{ij} is the angle between lines i and j.

The cross ratio can be used to identify pencils of four lines in a 2D image, allowing for single-camera pattern recognition.

3.2. Line-to-plane correspondences

Given a 2D line pencil in a camera image and its corresponding 3D pencil model, the geometric transformation that maps the model onto the 2D image features can be determined. The situation is illustrated in Figure 3. A pencil of 3D lines L_i is projected onto a pencil of 2D lines l_i in the image plane I. A line L_i is represented by its parametric equation

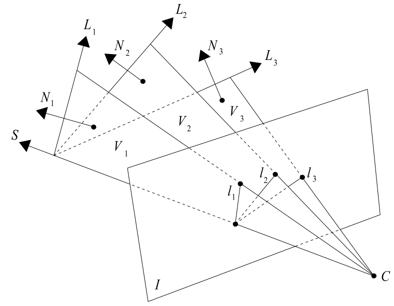

Figure 3: *3D lines L_i and their projections l_i on the image plane I, and the resulting 3D planes with normals vector N_i.*

$L_i = O_i + tD_i$, where O_i represents a point on the line, and D_i its direction. The camera C and the line projections l_i define a sheaf of planes V_i in 3D, with normal vectors N_i. Given only the camera C, the image plane I, and the pencil projection l_i, the problem is to determine a transformation such that the model lines are transformed into the corresponding planes V_i. This is generally known as the line-to-plane correspondence problem, and has been addressed by various researchers, e.g. [CH99, Che91].

The line-to-plane correspondence problem consists of two subproblems: determining position and determining orientation. The position of the 3D lines must lie on the intersection line S of the sheaf of planes. The position cannot be determined more accurately without using more cameras or extra features.

To estimate orientation, the problem is to find a rotation $\mathbf{R}$ applied to the 3D model lines, such that the directions D_i are transformed into the corresponding planes with normals N_i, i.e.

$$N_i^T \mathbf{R} D_i = 0 \tag{3}$$

Chen [Che91] has addressed the line-to-plane correspondence problem in the general case. He identified degenerate configurations of lines and planes, for which no solution can be determined. For valid configurations, he found a closed form solution in the case of three line-to-plane correspondences, resulting in an eight degree polynomial in one unknown. We will briefly review Chen's method, and show the simplifications we can make for the case of a configuration with a coplanar pencil of lines and a sheaf of planes.

Chen's method works by first rotating the model lines such that the first line D_1 lies inside the plane V_1, i.e. such that D_1 is perpendicular to N_1. This rotation is performed around axis $E = N_1 \times D_1$. The resulting configuration has only two degrees of freedom in rotation left. Axis E forms a coordinate system with N_1 and $\hat{D}_1 = E \times N_1$, which can be defined to be the x, y, and z axes of a coordinate frame in which the normals and rotated model lines are expressed. After rotating the lines to this canonical configuration, the remaining rotation can be written as

$$\mathbf{R}(N_1,\theta)\mathbf{R}(\hat{D}_1,\phi) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Substituting Equation 4 into 3 gives a system of equations in $\cos\theta$, $\sin\theta$, $\cos\phi$, and $\sin\phi$. This system can be solved using various algebraic manipulations and the fact that $\cos^2\theta + \sin^2\theta = 1$. The result is the polynomial in unknown $\cos\phi$

$$P(\phi) = \sum_{i=0}^{8} \sigma_i \cos^i \phi = 0 \quad (5)$$

where the coefficients σ_i are functions of the components of D_2, N_2, D_3, and N_3.

In our case, the lines have a common intersection point and are coplanar, and thus the directions $D_i = (x_i, y_i, z_i)$ and the normals $N_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ of the corresponding planes are both in a coplanar configuration. A canonical configuration is then easily obtained by applying a rotation $\mathbf{R}_D$ to orient the lines D_i in the YZ plane, such that $D_1 = (0,0,1)$, and a rotation $\mathbf{R}_N$ to orient the normals N_i in the YZ plane, such that $N_1 = (0,1,0)$. In the canonical configuration, all x-components of D_i and N_i are zero and cancel out the σ_i coefficients in Equation 5. It can be shown that the resulting polynomial can be written as

$$\begin{aligned} P(\phi) = {} & \alpha_3^4 - 2\alpha_3^2(\alpha_1^2+\alpha_2^2+\alpha_3^2)\cos^2\phi \\ & + (2\alpha_1^2\alpha_3^2 + (\alpha_1^2+\alpha_2^2+\alpha_3^2)^2)\cos^4\phi \\ & - 2\alpha_1^2(\alpha_1^2+\alpha_2^2+\alpha_3^2)\cos^6\phi + \alpha_1^4\cos^8\phi \end{aligned} \quad (6)$$

where

$$\begin{aligned} \alpha_1 &= -y_2\bar{y}_2\bar{z}_3 y_3 + y_2\bar{z}_2\bar{y}_3 y_3 \\ \alpha_2 &= y_2\bar{y}_2\bar{z}_3 z_3 - z_2\bar{z}_2\bar{y}_3 y_3 \\ \alpha_3 &= -y_2\bar{z}_2\bar{z}_3 z_3 + z_2\bar{z}_2\bar{z}_3 y_3 \end{aligned} \quad (7)$$

Equation 6 can be written as a square of fourth-degree polynomials with only a second and fourth order term. The roots are given by

$$\cos\phi = \pm\sqrt{\frac{\mathbf{B} \pm \sqrt{B^2 - 4\alpha_1^2\alpha_3^2}}{2\alpha_1^2}} \quad (8)$$

where $\mathbf{B} = \alpha_1^2 + \alpha_2^2 + \alpha_3^2$.

It is worth noting that the discriminant of Equation 6 is always positive, since $B^2 - 4\alpha_1^2\alpha_3^2 = 2\alpha_2^2(\alpha_1^2+\alpha_3^2) + (\alpha_1^2 - \alpha_3^2)^2 + \alpha_2^2 > 0$, and $\sqrt{B^2 - C} > B$ since $B^2 > C > 0$. This means that Equation 6 always has four real solutions.

After determining $\cos\phi$ and $\sin\phi = \pm\sqrt{1-\cos^2\phi}$, $\cos\theta$ and $\sin\theta$ can be found by substituting the solutions back into the system of equations defined by 4 and 3, resulting in

$$\cos\theta = \frac{\alpha_1 \cos\phi \sin\phi}{\alpha_3 \sin\phi} \quad (9)$$

$$\sin\theta = \frac{\alpha_2 \cos\phi}{\alpha_3 \sin\phi} \quad (10)$$

The final rotation $\mathbf{R}_{tot}$ is then given by

$$\mathbf{R}_{tot} = \mathbf{R}_N \mathbf{R}(N_1,\theta)\mathbf{R}(\hat{D}_1,\phi)\mathbf{R}_D^T \quad (11)$$

4. Method

Our optical tracking method is based on line pencil fiducials. Each pencil of 4 lines represents a pattern. Patterns are applied to the surface of each input device, as illustrated in Figure 1. The method comprises two stages: recognition and pose estimation. The first stage determines the feature correspondence, using the cross ratio of line pencils. Orientation estimation is accomplished by using single camera line-to-plane correspondences. Translation is estimated from multiple cameras. A fitting procedure is used to optimize the pose and obtain the final estimate.

We detect line shaped blobs in the camera images, and record a point on the line and its direction. Lines are detected by using a dynamic threshold to determine possible line pixels, after which a spread is performed to detect connected pixels belonging to a line. Next, lines are fit through the connected pixels. A point on each line is needed for a clustering step and to determine the intersection points of the line pencils. The line directions are used during recognition and pose estimation. Only part of a line needs to be visible in order to derive its parameters.

4.1. Recognition

The recognition stage involves determining the correspondence between the lines detected in the images and the lines stored in a model database. The model database consists of a list of pattern models for each input device. Each pattern model consists of four line directions and the location of the intersection point of the pencil. The recognition method works completely in 2D, and relies on the cross ratio, a common projective invariant property.

We will describe each step of the recognition stage.

Clustering. As lines of one device possibly form a pencil with lines from a second device, we apply a simple clustering method. Lines of different input devices will form separate clusters, unless the devices are very close together with respect to the viewing direction of the camera.

Pencil Detection. For every cluster of lines, a list of pencils is generated by calculating all line-line intersections, and finding the intersection points through which at least four lines pass. Intersection points of more than four lines generate $\binom{N}{4}$ possible pencil combinations. All these combinations are considered valid pencil candidates until they are invalidated at a later stage.

Cross ratio calculation. We calculate the cross ratio of line pencils for every pencil detected in the camera images. The computation of the cross ratio depends on the order of the four lines. Since we know the directions of the lines with respect to their intersection point, we order the lines in a clockwise fashion to prevent ambiguities in line order. This results in only one possible cross ratio for every pencil. After recognition, it also gives us feature correspondence instead of just pattern correspondence. This information is later used in the pose estimation stage.

Cross ratio check. Changing light conditions, varying illumination of the retroreflective markers, and miscalibrations all add to image noise, resulting in small variations of the line directions. As the cross ratio function is very sensitive to noise (see [sM95, May95] for probabilistic analyses of the cross ratio), we include a training session determining the intervals of the cross ratio of each pattern. The device designer moves the pattern around in the workspace, while the system determines a mean cross ratio and its range of deviations. The obtained cross ratio of each pattern and its associated range are stored in the model database. During recognition, all pencils outside the range are not considered candidates for the given pattern.

4.2. Pose estimation

The pose estimation stage involves calculating the position and orientation of all input devices that have been found in the images.

Orientation estimation. After recognition, each identified pattern is used to obtain an orientation estimate of the associated input device. Line-to-plane correspondences are used to calculate possible rotations of the pattern model, generally resulting in four solutions (see Section 3.2). Two of these solutions can be filtered out, since they fall outside the planes defined by the 2D line pencils l_i and the intersection line S, see Figure 4. In the figure, a valid solution resulting in lines L_i is illustrated along with the corresponding invalid solution, resulting in the mirrored lines L_i'.

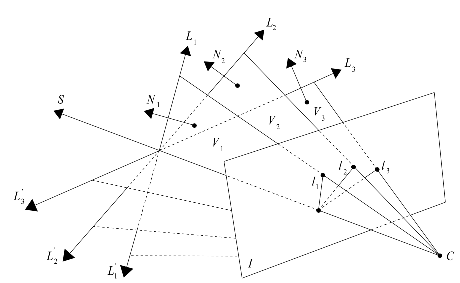

Figure 4: *A rotation solution resulting in lines L_i. For each rotation solution there is an invalid solution resulting in the mirrored lines L_i'.*

To disambiguate the two remaining rotation solutions of each pattern, we determine the mismatch between the rotations of each pattern with the patterns detected in other camera images. The orientation mismatch is defined as

$$\varepsilon = \cos^{-1}(q_1 \cdot q_2^{-1})[0] \tag{12}$$

where q_1, q_2 are quaternions representing solutions of a pair of pattern. As final device orientation, the solution with the smallest orientation mismatch is selected.

Determining device position. Device position is determined using the two patterns with the smallest orientation mismatch, i.e. the patterns used to derive the orientation estimate. We first determine the vector between the intersection points of these patterns in model coordinates, see Figure 5. This vector is transformed by the rotation matrix, resulting in the vector $W = P_1 - P_2$. Next, the camera location C_2 is translated to C_2' over vector W. The position P_1 is then given by the intersection point of lines (C_1, S_1) and (C_2', S_2).

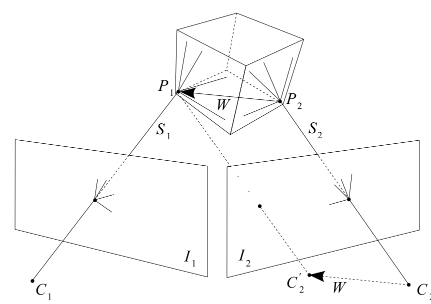

Figure 5: *Determining translation*

Invalid candidate pattern detection. After the complete pose of each device is determined, we detect all invalid pattern candidates that have been identified during recognition. All valid identified patterns will be used in a fitting step to further refine the pose estimate. To detect invalid pattern candidates, we examine the orientation mismatch between the rotations of each pattern candidate and the pose estimate of the associated device. If the difference in orientation is too large, the candidate is invalidated. Although it is possible that an invalid pattern candidate produces the same rotation, we have never experienced so in practice.

Fitting. Pose estimation based on line-to-plane correspondences can produce some jitter in the device pose while stationary, due to noise in the images and thus the detected 2D line directions. To reduce this problem, the estimated pose

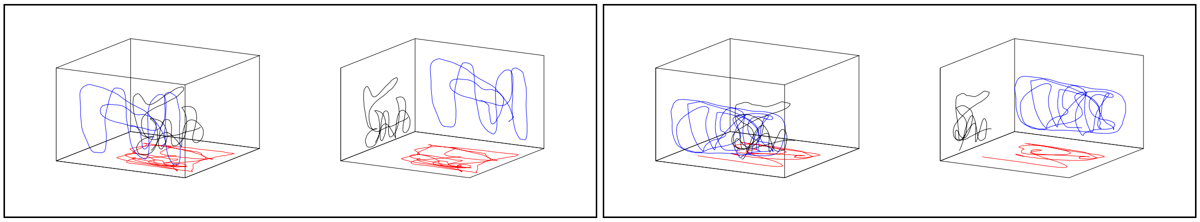

Figure 6: *The 3D data recordings in the XY, XZ, and YZ planes. Depicted are the recordings in the tracking volume and the 2D projections of each recording onto its corresponding plane. Left for the tracker based on line patterns, right for the tracker based on point patterns.*

is used as an initial value for a fitting method. We use the simplex algorithm, which is a well-known method for optimization introduced by George Dantzig in 1947 [Chv83]. The cost function to be minimized is defined by the angle and distance between the 3D planes of each identified pencil pattern and the corresponding transformed model lines. Since the initial pose is accurate, this procedure completes very fast.

5. Results

We have implemented and evaluated our optical tracking method on the PSS, our near-field desktop VR system. To examine the performance of our method, we compared the accuracy and latency to the tracking algorithm as described in [vLM03]. This method uses point patterns, using projection invariant properties for recognition and stereo geometry to transform recognized patterns to 3D in order to estimate a device pose. Each pattern consists of 5 coplanar points, applied to the sides of a cubic shaped input device. For both methods we used flat retroreflective material. For better reflectivity at glancing angles, spherical or cylindrical shapes could be used. The point markers have a diameter of 5 mm and need to be placed at least 4 mm apart. The line markers have a dimension of approximately 2x45 mm.

5.1. Accuracy

Method. An absolute accuracy study of an optical tracker is a time-consuming and tedious task. The tracking volume has to be divided into a grid of sufficient resolution. Next, the input device has to be positioned accurately at each grid position, after which the pose estimate from the tracker can be compared to the grid locations.

We follow the approach of Mulder et al. [MJvR03] to obtain a fast indication of the accuracy. Their approach entails moving the input device over three planes, and collecting the position measurements from the tracker. Next, for each data set, the measurements are fit to a plane by minimizing the RMS distance to this plane. Measures as the average and maximum distance of the measurements to the fitted plane give an indication of the accuracy of the tracker.

	Line based tracker				
Plane	Avg	Max	90%	99%	Unit
XY	0.38	1.21	0.70	1.06	mm
	0.24	1.10	0.49	0.96	deg
XZ	0.35	1.89	0.72	1.44	mm
	0.34	2.08	0.76	1.37	deg
YZ	0.28	1.68	0.53	0.96	mm
	0.15	2.70	0.30	0.80	deg

Table 1: *Line based tracker measurement-to-plane distances in mm and angular deviations in degrees.*

We extended this approach by including rotation. When moving the input device over the planes, we collect both position and orientation measurements. Next, the mean angle between the input device and the fitted plane is determined. We can then calculate accuracy measures as average and maximum angular deviation of the orientation measurements to the mean angle.

Results. Figure 6 shows the position measurements of both tracking methods over three orthogonal planes, corresponding to movements of the input device in the XY, XZ, and YZ planes. The workspace of the PSS is approximately 40x40x40 cm. Tables 1 and 2 summarize the results. Depicted is the average distance of the measurements to the plane, the maximum distance, and the maximum distances of the 90% and 99% points closest to the plane. For orientation, the average angular deviation of the measurements with respect to the mean angle between the input device and the plane is depicted, along with the maximum deviation and the maximum deviations of the 90% and 99% measurements closest to the mean.

From these tables it can be derived that both tracking methods perform very well. The line based tracker performs slightly better in position in the XZ and YZ planes, while the point based tracker performs slightly better in position in the XY plane. The line based tracker performs better in orientation in all cases.

	Point based tracker				
Plane	Avg	Max	90%	99%	Unit
XY	0.19	0.82	0.40	0.62	mm
	0.31	1.16	0.63	0.89	deg
XZ	0.47	1.62	1.01	1.54	mm
	0.65	1.99	1.19	1.63	deg
YZ	0.56	1.99	1.14	1.85	mm
	1.17	1.75	1.56	1.70	deg

Table 2: *Point based tracker measurement-to-plane distances in mm and angular deviations in degrees.*

N	Point (fps)	Line (fps)
5	160	110
10	50	100
15	5	92
20	« 1	80
25	-	66

Table 3: *Framerate measurements of the line based tracker versus the point based tracker, as a function of the number of 2D image features N. Measured using two cameras.*

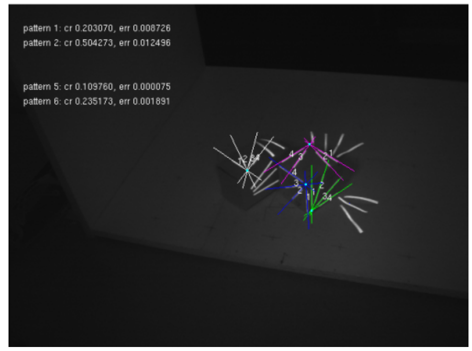

Figure 7: *A snapshot of a camera image for the line tracker. Recognized patterns are drawn in different colors.*

There are various sources of accuracy differences between both tracking approaches. First, both methods rely on a model description of each input device. Inaccuracies in these model descriptions translate directly to inaccuracies in the estimated pose. Second, the 2D features detected from the camera images are different. As the data used for recognition and pose estimation is different, both methods will yield different results in accuracy. Third, the line based method uses a fitting procedure on all identified patterns in the camera images, whereas the point based method uses only three points for pose estimation. Since in the XY plane only one pattern is visible to the cameras during the data recording, this fitting procedure does not increase accuracy much in this case.

5.2. Latency

We have recorded the framerate of both tracking methods as a function of the number of features present in two camera images. Framerates were measured on a system with a 2.2 GHz Pentium IV CPU and 1 Gb RAM. Table 3 summarizes the results when using two cameras. For both trackers, the features were placed relatively close together, in order to test the worst case situation where no clustering is performed. An example of a resulting camera image is shown in Figure 7. In the figure, 28 lines have been detected in the image. The recognition stage correctly identified 4 pencil patterns. The framerate for this situation was 67 fps. The framerate represents the total tracking time, including detection of 2D features in both camera images. Point and line detection in the camera images took about 7-9 ms of the total tracking time. The extra computational cost of line detection compared to point detection is about 1 ms.

From the table it can be derived that the point based method becomes infeasible for more than 15 features, a relatively low number. In contrast, the performance of the line based method almost decreases linearly with the number of features and maintains high framerates.

The performance issues of the point based tracker are due to the following. First, the line tracker can reject combinations of four lines that do not form a pencil at a very early stage. However, the point tracker has to test each combination of five points, and has to use stereo geometry to transform an identified pattern to 3D, before it can test whether the pattern is coplanar. Second, the point tracker only has pattern correspondence after recognition, while the line tracker has feature correspondence. Therefore, the point based tracker has to transform the identified pattern points to 3D using stereo geometry. Next, all 25 combinations of 5 points in both cameras have to be tested in order to determine feature correspondence.

5.3. Occlusion

One of the main advantages of our tracking method is its ability to handle considerable amounts of occlusion. Figure 8 shows an example of a user handling an input device, causing occlusion with his fingers. As there is still enough information in the camera images to derive four line directions, the tracking system is able to correctly identify the input device and estimate its pose. A similar amount of occlusion in case of an input device with point patterns will generally lead to tracking failure.

6. Discussion

In the previous sections we have described a method for the recognition of marker patterns based on projective invari-

Figure 8: *An example of occlusion: (left) A user occludes part of the pencil pattern. (right) A snapshot from the camera of the same view. The tracking system identified the correct pattern.*

ant properties of lines, and pose estimation based on line-to-plane correspondences. We now discuss some advantages and disadvantages of the recognition and pose estimation stages of the method.

6.1. Recognition

- *Accuracy*.
  The 2D features are subject to noise due to changing lighting conditions and camera properties. Calculating line directions from a 2D image can be done more accurately than calculating marker positions, unless those markers are very large. We have experienced smaller variations of the detected cross ratio's than for the point based tracker.
- *Latency*.
  The latency of our tracking approach increases linearly with the number of lines in the camera images. There are several points that make the tracker efficient. First, the ordering of pencil lines can be determined in 2D, so that there are no problems with cross ratio permutations. Second, since the line ordering is known, the exact feature correspondence is known after recognition. The point based tracker only establishes pattern correspondence during recognition, and has to test 25 combinations of 5 3D points in two cameras to determine feature correspondence. Third, all combinations of 4 lines that do not result in a pencil in 2D can be quickly identified. The point based tracker needs to consider each combination of 5 points during recognition, resulting in $\binom{N}{5}$ cross ratio calculations.
  The complexity of the recognition stage depends on the number of detected line features. The worst case performance is obtained when all lines in the camera image intersect at one point, resulting in $\binom{N}{4}$ pencils, with N the number of detected line features in the camera images. Therefore, the complexity of the recognition stage $O(N^4)$. However, in practise the number of pencils is low, and the recognition stage very efficient.
- *Occlusion*.
  The main motivation for using lines instead of points as pattern features is that it allows for significant amounts of occlusion. With points, depending on the method used, one missing point can be enough for the tracker to fail. For instance, the point based tracker used in the evaluation requires all points to be visible. Although in the line based tracker all lines in a pattern also need to be visible, it is no problem if part of the line is occluded, as long as its direction can be determined (see Figure 8). For the point based tracker, the occlusion problem could be reduced by adding more points to each surface, but computational cost would increase considerably.
- *Robustness*.
  During testing of the line based tracker, we found that the tracker cannot find a valid pattern in some cases, leading to tracking failure. In all cases we investigated, this problem was caused by the line detector used to extract line direction from the camera images. In some cases, the light reflected from the retroreflective fiducials back into the cameras is not enough in order to distinguish a line.
  However, since these tracking failures are caused by blob detection problems, the tracking method itself is very robust.
- *Pattern constraints*.
  Designing patterns for input devices is subject to some constraints. First, patterns have to consist of 4 lines intersecting in a common point, and have to be coplanar. Currently, if two patterns have one line completely occluded, information of both patterns cannot be used in the recognition stage. Future work will include determining if it is possible to use projective invariants of non-coplanar lines (see e.g. [Sug94]).
  Second, due to the sensitivity of the cross ratio to noise, only a limited number of patterns is possible. The cross ratio of each pattern has to be unique in its range. Moreover, the cross ratio function has many symmetries, resulting in duplicate pencil configurations. Currently we expect to be able to create about 20 distinguishable patterns.

6.2. Pose estimation

- *Accuracy*.
  The pose estimation method is sensitive to noise in the 2D line directions. Small jitter in the 2D lines is amplified in the 3D plane normal vectors. This results in a small amount of jitter while holding the device stationary. It is possible to reduce this jitter by including a subsequent filtering step.
  The accuracy of the pose estimate depends highly on the accuracy of the model. We have observed that holding a pattern at certain angles to the cameras produces a larger mismatch between the pattern orientations for both cameras. We are currently still investigating this phenomenon. The magnitude of the problem depends on the accuracy of the pattern model, and the resulting angular deviation in the final pose estimate is removed during the fitting step.
- *Latency*.
  The experimental results show that the method is quite fast, maintaining framerates of over 60 Hz with 25 line features visible in two cameras. Since the exact feature correspondence is known from the recognition stage, the pose estimation step is efficient.
  The complexity of the pose estimation stage depends on the number of patterns detected. For each detected pattern, two possible rotations are calculated. Next, each pattern pair is checked and the best pair is selected. This makes the complexity $O(N^2)$, with N the number of detected patterns.
- *Camera placement*.
  The pose estimation method based on line-to-plane correspondences is more flexible than the more common pose estimation method based on stereo geometry. Stereo geometry requires the same pattern to be visible in two cameras. Therefore, cameras need to be placed relatively close together. However, the accuracy of the pose estimate depends on the spacing between the cameras. A small camera spacing results in a low depth resolution.
  In the case of line-to-plane correspondences, camera restrictions are relaxed and the cameras can be placed more optimally with respect to occlusion and accuracy.
- *Generality*.
  Line-to-plane correspondences are used to obtain an orientation estimate of the input device for each pattern. Although the line-to-plane correspondence method works with a single camera, it yields two valid orientation solutions. A second camera is thus needed to derive the correct solutions. We have considered using an extra line for each pattern to disambiguate the two orientation solutions, and to determine position from one camera. However, position estimates would be too inaccurate due to the low resolution in the viewing direction of the camera. Since we need an extra camera for position, and more lines would clutter the images more and produce more candidates during recognition, we chose to disambiguate the two pattern orientation solutions using this extra camera.

7. Conclusion

In this paper, we have described a new optical tracking algorithm based on line pencil fiducials. Patterns are recognized using the cross ratio of line pencils. The cross ratio is a projective invariant property and thus allows for single camera recognition. An orientation estimate is obtained by using single camera line-to-plane correspondences. Translation is derived from multiple cameras.

Results show the method has lower latency and a comparable accuracy compared to a related point based tracking method. This is due to several properties of line pencils, which allow us to reject feature combinations at an early stage in the algorithm, and allows for a single camera recognition and orientation estimation approach.

An important advantage of line fiducials is that only part of the fiducial needs to be visible in order to detect its direction. Therefore, our tracking method allows for significant amounts of occlusion.

Future work will include investigating other projective invariant properties of lines and their applicability to optical tracking.

References

[ART] http://www.ar-tracking.de/.

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 239–256.

[CH99] CHRISTY S., HORAUD R.: Iterative pose computation from line correspondences. *Comput. Vis. Image Underst. 73*, 1 (1999), 137–144.

[Che91] CHEN H. H.: Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 6 (1991), 530–541.

[Chv83] CHVATAL Q.: *Linear Programming*. W. H.Freeman and Co., 1983.

[Dor99] DORFMÜLLER K.: Robust tracking for augmented reality using retro-reflective markers. *Computers & Graphics 23*, 6 (1999), 795–800.

[DYN] http://orin.com/index.htm.

[FP02] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[LW88] LAMDAN Y., WOLFSON H.: Geometric hashing: A general and efficient model based recognition scheme. In *Proc. International Conference on Computer Vision* (1988), pp. 238–249.

[May95] MAYBANK S.: Probabilistic analysis of the application of the cross ratio to model based vision: Misclassification. *Intl. J. of Computer Vision 14* (1995), 199–210.

[MJvR03] MULDER J. D., JANSEN J., VAN RHIJN A.: An affordable optical head tracking system for desktop vr/ar systems. In *Proceedings of the Eurographics Symposium on Virtual Environments 2003* (2003), ACM Press, pp. 215–223.

[MLR98] MEER P., LENZ R., RAMAKRISHNA S.: Efficient invariant representations. *IJCV 26*, 2 (1998), 137–152.

[MvL02] MULDER J. D., VAN LIERE R.: The personal space station: Bringing interaction within reach. In *Proceedings of the Virtual Reality International Conference, VRIC 2002* (2002), pp. 73–81.

[OPT] http://www.ndigital.com/optotrak.html.

[RPF01] RIBO M., PINZ A., FUHRMANN A.: A new optical tracking system for virtual and augmented reality applications. In *Proc. of IEEE Instrumentation and Measurement Technology Conference, IMTC* (2001), vol. 3, pp. 1932–1936.

[sM95] ÅSTRÖM K., MORIN L.: Random cross ratios. In *Proc. 9th Scand. Conf. on Image Anal.* (1995), pp. 1053–1061.

[Sug94] SUGIMOTO A.: Geometric invariant of noncoplanar lines in a single view. *Proc. 12th IAPR Int. Conf. on Pattern Recognition (ICPR'94)*, 10 (1994), 190–195.

[vLM03] VAN LIERE R., MULDER J.: Optical tracking using projective invariant marker pattern properties. In *Proceedings of the IEEE VR2003 Conference* (2003), pp. 191–198.