Real-time 3D hand tracking in a virtual environment

Kevin C. Smith[a], Dan Sandin[b], Thomas S. Huang[a], Joshua J. Eliason[b], and Geoffery A. Baum[b]

[a]University of Illinois at Urbana-Champaign, 405 N. Matthews, Urbana, IL, USA
[b]University of Illinois at Chicago, Chicago, IL, USA

ABSTRACT

The development of a reliable untethered interactive virtual environment has long been a goal of the VR community. Several nonmagnetic tracking systems have been developed in recent years based on optical, acoustic, and mechanical solutions. However, an inexpensive, effective, and unobtrusive tracking solution remains elusive. This paper presents a camera based three-dimensional hand tracking system implemented in the PARIS augmented reality environment and used to drive a demonstration application.

Keywords: Hand tracking, Virtual Reality, PARIS, CamShift, mean shift, tracking

1. INTRODUCTION

Since the development[1] of the CAVE ® (CAVE Automatic Virtual Environment) in 1992, a growing set of interactive virtual graphics applications has created a need for systems capable of reliably tracking body parts such as the head and the hand in a three-dimensional environment. Most CAVE-based virtual reality display device in use today has addressed this need with the use of a magnetic tracking system such as the Flock of Birds or pciBIRD (TM) from Ascension Technology Corporation. Yet this type of system has several drawbacks. Although some magnetic tracking systems are wireless, the pciBIRD magnetic tracking system necessitates that each sensor be tethered to the tracking system. This effectively limits the user's range of motion and makes the virtual reality (VR) experience less immersive. Since the system tracks based on changes in the magnetic field, the system is sensitive to conductive materials in the workspace. Another documented drawback associated with magnetic tracking systems is their range of operation. Although they often have high resolutions, the accuracy of many magnetic tracking systems falls dramatically when nearing the edge of their range of operation.[2]

Creating a untethered interactive virtual environment has long been a goal of the Electronic Visualization Lab (EVL) at the University of Illinois at Chicago, and of the VR community in general. Currently, users are required to manipulate a tethered wand to interact with the VR system. Several nonmagnetic tracking systems have been developed in recent years. They have been based on various tracking techniques including optical, acoustic, and mechanical solutions.[3] Other, more elaborate systems exist that combine several methods, such as the Constellation.[4] These systems have addressed the problem with varying degrees of success. However, a cheap, effective, unobtrusive tracking solution still remains elusive.

This paper presents an optical-based tracking system, dubbed CAMtrack3D, and a subsequent performance-analysis. Results will then be compared to the pciBIRD. CAMtrack3D operates by separating the 3D tracking problem into two major tasks: first, it tracks the objects in the two-dimensional video sequence captured from the camera, and second it uses the two-dimensional data along with knowledge of the environment to determine three-dimensional positions.

2. SYSTEM OVERVIEW

In this section we will give an overview of how the tracking system works and how it is affected by the characteristics of PARIS (Personal Augmented Reality Immersive System). First, we will explore the PARIS virtual reality display system and how its design and operation affect three-dimensional tracking. We will also become familiar with magnetic tracking systems and how they function, so they may be compared to CAMTrack3D's camera-based tracking system later.

Further author information: (Send correspondence to K.C.S.)
K.C.S.: E-mail: kevin.smith@idiap.ch, Telephone: +41 27 721 77 67

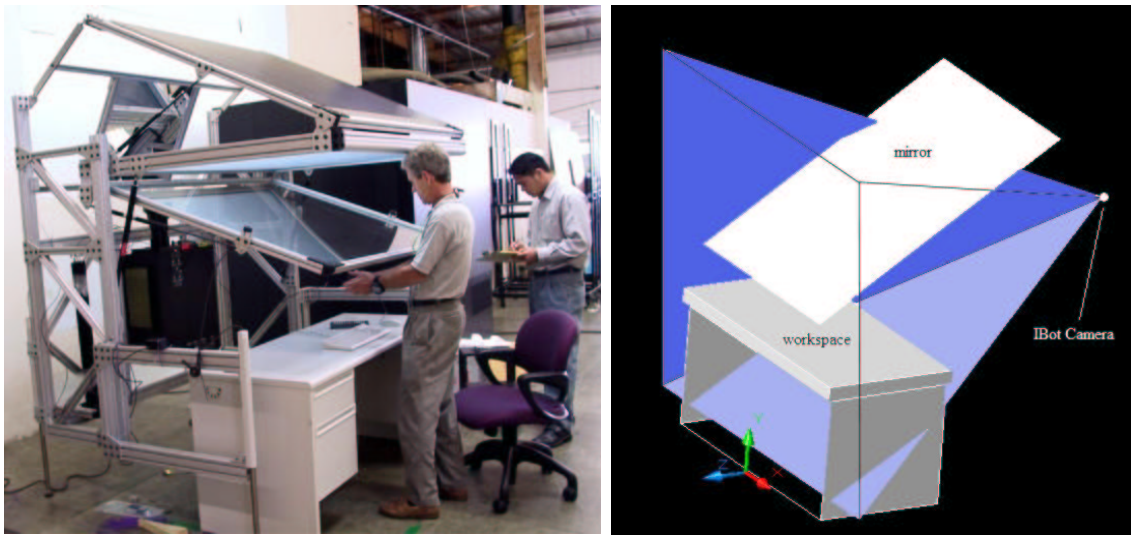

Figure 1. *(left)* Photograph of the first PARIS prototype at the University of Illinois at Chicago. *(right)* A computer model of the camera's field of view in the PARIS VR display.

2.1. PARIS

Located on the first floor of the SEL building at the University of Illinois Chicago campus is the prototype model of PARIS. CAMtrack3D was designed and calibrated to operate specifically for this prototype system.[5] Figure 1 depicts the arrangement of the final version of the PARIS prototype.

The stereo image is presented to the half-silvered mirror display when the projector reflects the image onto the mirror and then onto the screen. The screen is oriented horizontal to the floor above the user's head. The stereo image is then back-projected onto the screen from the mirror so that it appears correctly to the user when the reflection is viewed in the half-silvered mirror. The total viewing area of the half-silvered mirror is 48 inches deep and 66.5 inches wide. Below the half-silvered mirror is a desk, which serves as a workspace for the user.

Because of the unique geometry of the PARIS prototype, a viewer stationed at a location behind the half-silvered mirror will observe the user's hands along with the reflection of the user's hands in the half-silvered mirror. It has been shown that in order to reconstruct the three-dimensional position of an object, two views of the scene must be known.[6] Normally, acquiring two views of a scene requires that two cameras be placed in some sort of stereo-rig, but in this case the two views of the same scene are observable to a single camera because of the reflection of the scene in the half- silvered mirror. If a single camera is placed so that it observes both the user's hands and the reflections of the user's hands, it will contain all the information required to reconstruct the three-dimensional position of the hands within a single video image.

Since CAMTrack3D relies on the video data from only one camera, a virtual camera must be created to model the second view. The virtual camera has two unusual properties. First, it must be modeled at the exact reflection point of the physical camera about the half-silvered mirror where the image is observed. Secondly, the handedness of the reflected (virtual) image will be switched since the physical camera is observing a mirror reflection. This must be accounted for when CAMtrack3D estimates the 3D position of the hands, as the horizontal coordinates of the virtual camera will be reversed.

PARIS was designed with a half-silvered mirror so that the user could see his/her hand in the workspace on the other side of the mirror. However, the projector in the prototype PARIS is extremely dim and so PARIS must be operated with all other lights in the room shut off. Since the construction of the prototype PARIS system, brighter projectors have entered the market that will eliminate this problem. However, until work is done to upgrade the projector in the PARIS prototype, the lighting conditions had to be dealt with otherwise.

A solution was found that allows the user to see his/her hand through the half-silvered mirror without having to illuminate the workspace with disrupting light. This solution called for the use of a blacklight and fluorescent colored gloves. A blacklight was mounted at the top of the half-silvered mirror so that the workspace was covered with ultraviolet light. Only the fluorescent gloves are illuminated under the blacklight, and so the PARIS application is not disrupted by extra light and the hands become visible to the camera and the user. An added benefit gained from using the gloves and blacklight is that CAMtrack3D could now be modified to track two hands instead of just one if the gloves are colored differently.

2.2. Magnetic Tracking

The PciBIRD is a stand-alone magnetic tracking system developed by Ascension Technologies that has the following components: a transmitting antenna, which is mounted below the desk and transmits electromagnetic signals; two receivers, one attached to a receiveing antenna on the stereo glasses and one attached to a receiveing antenna on the wand; a PC running proprietary software and collecting position data.

2.3. Hardware

One important advantage of the CAMTrack3D system is the low equipment cost and the ease of setup. The CAMTrack3D system is intended to run on a typical off-the-shelf PC. The only equipment required to run CAMTrack3D other than a standard PC is a PCI firewire card and a webcam. CAMTrack3D's setup uses an ADS PYRO IEEE 1394 Firewire card and an OrangeMicro IBot camera. The OrangeMicro Ibot is an affordable ($90 USD) IEEE 1394 Webcam that is capable of transmitting 400 Mb/sec non-compressed, non-interlaced full-motion digital video. This translates to about 30 fps at a resolution of 640x480.

3. COLOR TRACKING

The first of the two major tasks of CAMTrack3D is to perform 2D tracking of the user's hands. For color tracking should meet the following goals to be considered inexpensive and robust: (1) be computationally inexpensive; (2) tolerate image noise; (3) ignore distracters; (4) handle lighting variation; (5) tolerate partial occlusion; (6) recover from errors; (7) handle irregular object motion; (8) have comparable resolution to magnetic tracking system; (9) have comparable accuracy to magnetic tracking system; (10) have a comparable range of operation to magnetic tracking system; (11) have comparable update rate to magnetic tracking system; (12) exhibit low latency; (13) show low level of positional noise.

3.1. CamShift Algorithm

The CamShift algorithm was chosen to perform the color-based 2D tracking because it is simple and computationally inexpensive. It is also a very robust algorithm as it deals with common tracking problems such as irregular object motion, image noise, and distracters "for free." This is because the CamShift algorithm draws on ideas from robust statistical analysis. Robust statistics tend to ignore outliers, or data points far away from the region of interest. This makes them useful in dealing with noise and distracters in video sequences. For this reason, Gary Bradski[8] chose to use a mean shift technique when developing the CamShift algorithm. The mean shift algorithm is a robust nonparametric technique for finding the element that maximizes the density of a probability distribution. However, mean shift alone will not reliably track video in real time because it does not change its window size. Further steps were added to the algorithm to continuously adapt the window size for each video frame. The algorithm was thus appropriately dubbed the Continuously Adaptive Mean Shift (CamShift) algorithm. CAMTrack3D sends the video data from the IBot through four CamShift filters, each filter designated to track a particular hand or reflection of a hand. Each filter calculates a window that bounds the area of the hand being tracked. It then draws a bounding box in the video frame and sends the data off to the next filter. Results from are later used for 3D calculations. The last CamShift filter sends the video data to the video renderer to be displayed. Figure 2 is a block diagram of the operation of the CamShift algorithm. The steps in the dashed area are only completed when training the histogram at the start of operation.

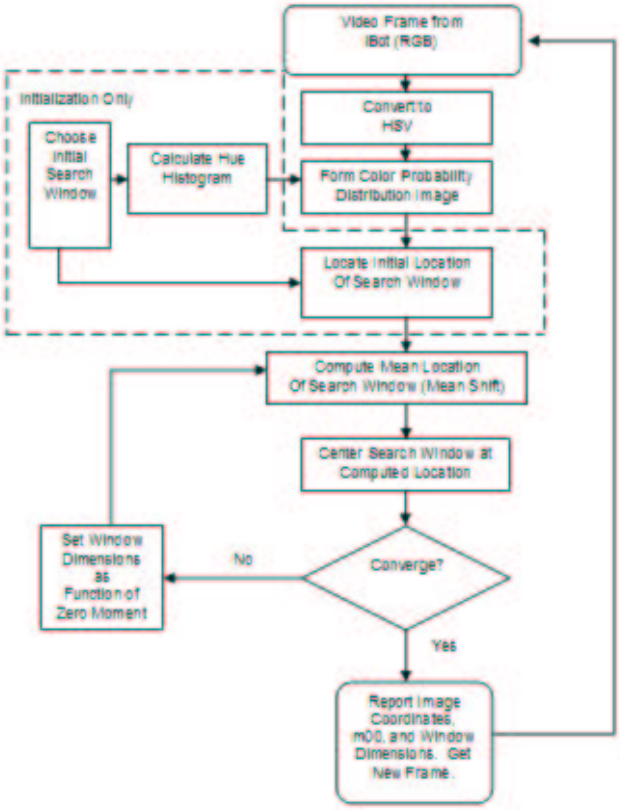

Figure 2. Block diagram of CamShift algorithm as implemented in CAMTrack3D.

3.1.1. Initialization

In order to initialize the system, the user must select an initial search window. The initial search window for each filter should bound the appropriate hand in the image. The image data inside the window will be used to seed the mean shift algorithm and to form a color histogram. If an incorrect window is chosen the histogram will be formed using incorrect color information and tracking will fail. To ease the process of initialization, the user's hands must be resting at the initialization positions. This is done to ease the process of initialization. The initialization positions are marked by handprints on the desk in the PARIS workspace. The user may alternatively initialize the system by dragging the mouse over an area in the image bounding the object to be tracked. The predetermined initialization locations can be fine- tuned at run time through the user interface. The system may be reinitialized at any time by pressing the *Track/Set Histogram* button if, for instance, the first initialization was unsuccessful.

3.1.2. Hue histogram

After a successful initialization, the window data for each hand is used to create a color histogram which will serve as a weighting function for the mean shift algorithm. The color, or hue histogram is created using the Hue Saturation Value (HSV) color system[9, 10] instead of the RGB values reported from the camera. HSV space separates out the hue (color) from saturation and brightness.

In CAMTrack3D, the hue histogram can have from 10 to 120 bins. Each bin represents a discrete range of hue values. The default number of bins is 80. The histogram is created by sampling the H value for each pixel in the window and binning it into the 1D histogram and normalizing.

3.2. Color distribution

In order for CAMTrack3D to track colored objects in a video sequence, a probability distribution image of the desired color in the video scene must be created for every video frame. A probability distribution function gives the probability of all possible outcomes accumulated from the reference outcome (starting point) up to the current outcome. A one-dimensional probability distribution is given by

$$F(x_j) = \sum_{k \leq j} f(x_k)$$

where $f(x_k)$ is the probability of a possible match. The color distribution image is a graphical representation of a two-dimensional probability distribution. During operation, the normalized hue histogram stored in memory from the initialization is used as a lookup table to create the color probability distribution image. RGB image data from the region of interest around the window is converted to HSV values. This data is then referenced with the hue histogram of the object to be tracked. The probability of a color match is determined by the size of the bin that the pixel has been referenced with. Using this method, probabilities range in discrete steps from 0.0 to 1.0. Since we are using 8-bit hue values, this corresponds to a range from 0 (lowest probability) to 255 (certainty). These probabilities are written as pixel intensity values in the image array, so that a bright pixel corresponds to a close match and a dark pixel corresponds to a background pixel.

Some problems naturally arise when using HSV space to model the color system. In practice, when the brightness level (V) or the saturation level (S) is low, the hue level (H) often becomes quite noisy and is often unreliable. In order to alleviate this problem, threshold levels have been introduced for brightness level and saturation level. Pixels with brightness or saturation levels below the experimentally determined threshold are ignored when creating the color distribution image. This results in much less noisy color probability distribution images. However, this means that when presented with dimly lit scenes, CAMTrack3D simply may not be able to track.

3.3. Mean shift

The mean shift algorithm is a nonparametric technique that climbs the gradient of a probability distribution to find the nearest dominant mode or peak. It has been proven that the mean shift algorithm converges to the modes of probability distributions.[11] This is akin to color matching, since intensity peaks in the probability distribution image represent the most probable matching points to the color model acquired in the initialization phase. The initial location of the search window used in the mean shift algorithm is determined in the initialization procedure. The mean shift algorithm then operates on the region of interest in the color distribution image. The region of interest is equal to the search window plus ten pixels in any direction. After the initial window size and location are determined, the mean shift algorithm computes the mean location or centroid within the search window. The search window is then centered at that location and the algorithm iterates to convergence.

For discrete two-dimensional probability distributions (such as in the color probability distribution image), the mean location can be computed by calculating the zeroth moment and the first moment in x and the first moment in y. The zeroth moment M_{00}, the first moment in x M_{10}, and the first moment in y M_{01}, are given by

$$M_{00} = \sum_x \sum_y I(x,y), \quad M_{10} = \sum_x \sum_y xI(x,y), \quad M_{01} = \sum_x \sum_y yI(x,y)$$

where I(x,y) is the intensity value of the pixel located at (x,y). From these values, the x and y coordinates of the mean location or centroid can be determined by

$$x_c = \frac{M_{10}}{M_{00}}$$

and

$$y_c = \frac{M_{01}}{M_{00}}.$$

In order to check for convergence, CAMTrack3D checks to see if the mean location has moved less than an empirically determined threshold. When this criterion is met, the CamShift filter reports the image coordinates and the window size to the three-dimensional calculation routine and waits for a new video frame. The mean shift algorithm is not meant to track dynamic probability distributions such as a video scene, since the window size of the mean shift algorithm does not change. In a video sequence, with each new frame the object to be tracked may grow or shrink. Statically adjusting the size of the window will not alleviate this problem. A small search window will get lost within a large object and data misrepresentative of the centroid location. A large fixed-size window would include distracters and also report erroneous data. Thus, it becomes necessary to use an adaptive window to track objects in a dynamic setting.

3.4. Adaptive window

In order to properly track objects in the video sequence, CAMTrack3D must adaptively adjust its search window size. When a new video frame becomes available, the current window size is enlarged in each direction by ten pixels to form the region of interest. The color distribution image is then calculated from the region of interest. The mean shift algorithm is then applied to the region of interest to find the centroid, and the search window location is moved to the location of the centroid. Next, the system checks the mean shift algorithm for convergence by comparing the location change to an empirically determined limit value eps: $dx^2 + dy^2 \leq eps^2$, where dx is the incremental movement in the x direction and dy is the incremental movement in the y direction. If the system converges, the image coordinates, zeroth moment, and window dimensions are reported and the system awaits a new video frame. Otherwise, the window dimensions are adaptively resized and the mean shift algorithm is applied again. Window size is adjusted by treating the color blob as an ellipse and calculating the second-order central moments. Then the linear size of the object is calculated assuming its shape is elliptical. The major and minor axes of the ellipse can be used to construct a rotated bounding box around the object. From there, an upright bounding rectangle is calculated and used as an initial window to for the next mean shift function call. The second-order moments are defined as

$$M_{11} = \sum_x \sum_y xyI(x,y), \quad M_{20} = \sum_x \sum_y xxI(x,y), \quad M_{02} = \sum_x \sum_y yyI(x,y).$$

The orientation of the major axis of the blob can then be calculated:

$$\theta = \frac{\arctan\left(\frac{2(\frac{M_{11}}{M_{00}} - x_c y_c)}{(\frac{M_{20}}{M_{00}} - x_c^2) - (\frac{M_{02}}{M_{00}} - y_c^2)}\right)}{2}.$$

The major and minor axes of the ellipse (l and w) are given by:

$$l = \sqrt{\frac{(\frac{M_{20}}{M_{00}} - x_c^2 + \frac{M_{02}}{M_{00}} - y_c^2) - \sqrt{2(\frac{M_{11}}{M_{00}} - x_c y_c)^2 + (\frac{M_{20}}{M_{00}} - x_c^2 - \frac{M_{02}}{M_{00}} - y_c^2)^2}}{2}},$$

$$w = \sqrt{\frac{(\frac{M_{20}}{M_{00}} - x_c^2 + \frac{M_{02}}{M_{00}} - y_c^2) + \sqrt{2(\frac{M_{11}}{M_{00}} - x_c y_c)^2 + (\frac{M_{20}}{M_{00}} - x_c^2 - \frac{M_{02}}{M_{00}} - y_c^2)^2}}{2}}.$$

The CamShift algorithm iterates through a sequence of steps to shift location and adapt its window size until it converges. It continues to adjust its size and move toward the center after each iteration until it converges near the center. This entire process is executed out on a single video frame. When the next video frame becomes available, the search window size and location from the previous frame will be used to start the same process on the next video frame.

The CamShift algorithm does have some limitations. The search window can only increase in size by 10 pixels for any given video frame. If an object moves so that it is 10 pixels or more away from its image in the last frame, the CamShift filter is guaranteed to lose tracking. Because of the high frame rate and large distance between the camera and the user, this is not a typical behavior, although some close fast-moving objects may cause the system to fail under this condition. The minimum size of the search window is 3 pixels, and the window size must always be odd so there is an integer value for the center of the window.

3.5. Reflection Boundary

A systematic problem occurs if the user moves his hand near the half-silvered mirror when interacting with an application in PARIS. Because CamShift is a color-based blob tracking algorithm, an individual tracker cannot distinguish between two blobs of similar color that come within close proximity of each other. The algorithm treats the two blobs as if they were one large blob and proceeds to grow the search window until it encompasses both blobs. When the objects once again separate, the CamShift filter will track the larger and/or more closely color-matched blob. This becomes a problem when tracking multiple objects of the same or similar color, as is the case with a hand and its reflection.

Because of the geometry of the system and the placement of the camera, the image of the reflection of the hand will always appear above the image of the hand itself. Once each video frame CAMTrack3D calculates the midpoint between the upper bound of the filter tracking the image of the hand and the lower bound of the filter tracking its reflection once for each video frame. An imaginary boundary is created at the horizontal line corresponding to the midpoint. The boundary information is updated in each of the four filters once for every video frame. CamShift search windows are checked every frame to see if they cross the boundary line from the past frame. In essence, the trackers are not allowed to cross the reflection line. If they do, they are reset above or below the reflection boundary line accordingly. This guarantees that a hand and its mirror reflection will not confuse the tracker when they come within close proximity of one another.

3.6. Error Recovery

One of the shortcomings of the CamShift algorithm is that it is not guaranteed to resume tracking when tracking has been lost. In some cases, the tracker will not resume tracking unless the system is reinitialized. Another common problem occurs when the CamShift filter begins tracking a distracter after tracking on the original object has been lost. Although they may sometimes report erroneous data, the magnetic trackers do not suffer from the problem of permanently losing two-dimensional tracking. In general, tracking can be lost when the user moves his/her hand too fast or removes the hand from the field of view of the camera.

Here we will introduce the concept of the two different failure modes: single failure and double failure. Each of the different methods of failure studied in the previous paragraph can be assumed to correspond with either one or both of the trackers losing tracking. Now it is left to determine what constitutes single failure mode and what constitutes double failure mode. Zeroth-moment data and the image x coordinate from each hand, along with the three- dimensional error distance are used to decide if the system is operating correctly or in a mode of failure. Double failure mode can only occur when the zeroth-moment data for both hands is below a certain threshold value (experimentally determined to be 5000). Successful tracking requires (1) that each hand's zeroth moment be within 10% of the other, (2) image locations of the tracked objects be within 20 pixels of each other, and (3) the three-dimensional calculation error be lower than 1.5 inches. When all these criteria are met for both hands, the system is considered to be in tracking mode. Any failure of the above the criteria that does not result in a double failure mode will result in single failure mode. The system must then decide which filter is failing and flag that filter as in a failure mode.

Once the system decides that a tracker is in a failure mode (double or single failure), it uses this information to reacquire tracking. When in single failure mode, one of the tracking filters for the hand is still delivering reliable data. Because of the symmetry of the system, this information is used to predict the location of the failed filter. It was experimentally observed that the center of a hand and its reflection are always within twenty pixels of each other while in the workspace. That knowledge, along with the reflection border data, can be used to create a localized estimate of the position of the failed tracker. The search window can then be reset to the localized estimate and tracking should resume.

When the double failure mode occurs, there is no information available to make such an estimate of tracker position. The search window must then be set to a very large area to ensure that it will contain the image of the hand/reflection. During setup, one of the parameters the user must input to CAMTrack3D is the location of the edge of the mirror in the image. This information is then used to separate the video frame into two searchable areas for each hand.

4. TRIANGULATION

The second of the two major tasks of CAMTrack3D is to triangulate the three-dimensional position of the user's hands using the coordinates of the centroids of the hands and their reflections in the image. This problem is trivial when no noise is present in the system, but for the PARIS application CAMTrack3D must be able to estimate the position of the user's hands in the presence of noise.

4.1. Overview

It has been pointed out that the midpoint method is neither affine nor projective invariant, and thus behaves poorly under affine and projective transformations.[12] However, since the calibration and pose of our camera are known, this is not a concern to us. Epipolar geometry tells us that given two views of an object and a known pose between the cameras, it is possible to know the position of the object in three- dimensional space. This is not practical in a real application because the image points are imperfect in practice and will contain noise. The result is that the rays projecting from the camera center points will not necessarily intersect, and thus the problem becomes the estimation of the point of intersection. The midpoint method attempts to find the midpoint of a common perpendicular line segment between two rays. In CAMTrack3D, this is done by computing the distance between points along the rays and finding the minimum.

4.2. Virtual Camera

The first step in the triangulation is to model each camera's position in three-dimensional space. CAMTrack3D accomplishes this by creating a virtual camera, or a model of the reflection of the real camera. This virtual camera is modeled just as a real camera would be, but the coordinate system of the virtual camera must become left-handed, and it is positioned where the reflection of the camera would appear to itself. This transformation of coordinates is called the *parity transformation*. It is equivalent to a mirror reflection followed by a 180 degree rotation parallel to the mirror. The reflection, rotation, and parity transformation together are known as the *virtual camera transformation*. The position of the real camera must undergo a translation (-z direction) towards the mirror along the line of its optical center. The distance to the mirror from the camera is denoted d_{mirror}. It must then be rotated about the x axis by twice the angle $-\theta$, which is the angle between the mirror and the optical center of the camera. Then a final translation is performed in the +z direction by a distance d-mirror. After this, the coordinates of the virtual camera are known. Mathematically, this transformation can be written

$$\mathbf{C_v} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{mirror} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(-2\theta) & -\sin(-2\theta) & 0 & 0 \\ \sin(-2\theta) & \cos(-2\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_{mirror} \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{C_1}$$

where and $\mathbf{C_1}$ are the coordinates of the virtual camera.

The virtual camera transformation depends on the fact that the optical center of the camera is perpendicular to the mirror in the plane perpendicular to the y axis. The image points along with the camera centers must be modeled in order to draw the intersecting rays and determine the position of the object being tracked. So it is also necessary to determine how to model the image planes in three dimensions given the image coordinates from the CamShift filters and the locations of the camera centers.

The image planes are modeled by constructing planes perpendicular to the z axis located at the point z = -1 for both the IBot camera and the virtual camera. The x and y coordinates are then adjusted according to the focal lengths. Finally, the 3D image coordinates are converted into a unit vector originating at the camera center and directed towards the image coordinate. Since we will later require both sets of 3D image coordinates in the same coordinate system, it is necessary to transform the image coordinates of the virtual camera into the real camera coordinate frame.

This transformation, the *image point transformation*, is performed on the unit vector pointing from the virtual camera's center to the image coordinate on the projected image plane. The virtual camera's image coordinate is calculated from the origin just as the real cameras. It is then converted to a unit vector in a similar manner and transformed to account for the different orientations of the camera coordinate frames. This transformation

involves first rotating by -2θ about the x axis, and then 180 degrees about the z axis. Finally, it is placed at the coordinates of the virtual camera center C_v. The x coordinate must be multiplied by -1 in order to account for the parity transformation due to the mirror. The entire image point transformation can be written mathematically as

$$\mathbf{x_2'} = \begin{pmatrix} \cos(-2\theta) & -\sin(-2\theta) & 0 & 0 \\ \sin(-2\theta) & \cos(-2\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\pi) & -\sin(-\pi) & 0 \\ 0 & \sin(-\pi) & \cos(-\pi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \mathbf{x_2} \\ 0 & 1 & 0 & \vdots \\ 0 & 0 & 1 & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{C_v}$$

where $\mathbf{x_2}$ is the image coordinate unit vector for the virtual camera as calculated in the virtual camera's reference frame and $\mathbf{x_2'}$ is the transformed vector in the real camera frame.

4.3. Camera Calibration

In order to correctly model the camera and the virtual camera correctly, we must be able to model the image planes in three-dimensional space. We have already determined the camera centers and the image points within the image planes, but the rays cannot be constructed until the geometry of the projected image planes is known. Solving this problem requires knowledge of the camera's intrinsic parameters. The intrinsic parameters were determined using CalibFilter, an OpenCV filter based on a technique developed by Zhang at Microsoft Research.[13]

CamTrack3D uses the focal length data from CalibFitler's output files to determine the positioning of the image frame with respect to the camera center. After the image planes are modeled, the image points can be placed within the image plane by the following transformation. For corresponding image points (x1,y1) and (x2, y2) in the real camera and the virtual camera, the 3D position of the image points within the real camera's reference frame is

$$x = \frac{x_1 - 160}{f_x} \quad y = \frac{y_1 - 120}{f_y} \quad z = -1$$

and the 3D position of the image points within the virtual camera's reference frame can be given by

$$x = \frac{-(x_2 - 160)}{f_x} \quad y = \frac{y_2 - 120}{f_y} \quad z = -1$$

The rays can then be constructed by drawing a line from the camera center to the image points. The only remaining problem is then determining the best intersection point. Figure 3 depicts the scene constructed thus far, including the camera centers, image planes, mirror, image points, and projected rays. The rays converge at a point in the PARIS workspace (x,y,z) that should approximate the 3D position of the tracked hand.

4.4. Iteration to Triangulation

As was discussed in Section 4.1, the rays will generally not intersect, so an estimate of the intersection point must be performed. CAMTrack3D estimates the intersection point using the midpoint method, which finds the midpoint of a common perpendicular line segment between two rays. This is done through an iterative process that estimates the intersection point by computing the distance between points along the rays and finding the minimum. CAMTrack3D computes the distance between each of the points and finds the smallest distance. It then uses the closest points to center five new points along the ray, spaced in smaller intervals. The process repeats itself iteratively until the points are within 0.01 inch of each other. After the final iteration, the two closest points are used to construct a line segment. The midpoint of the line segment is computed and stored as the estimation of the (x,y,z) position of the hand.

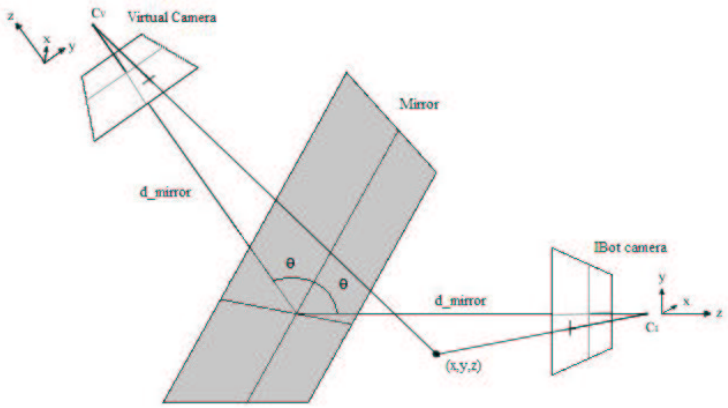

Figure 3. Triangulation. Rays projected from the camera center of the real and the virtual camera pass through the image points on each image plane and meet at a point in space (x,y,z) that approximates the position of the tracked hand.

5. ANALYSIS

There are several important measurable quantities that can be used to evaluate any tracking system's performance. We will address six within this section. The first performance measure we will look at is resolution. Resolution refers to the smallest reliable unit of measure according to which the system can distinguish the 3D coordinate. The second performance measure, accuracy, is a measure of how well the reported coordinates match the ground truth. The range of operation gauges the space in which the system can be relied on to operate without a significant increase in error. CAMTrack3D is a unique system in that its resolution depends on position, so range of operation takes on additional meaning. The update rate is the speed at which the 3D coordinates are calculated and reported. High update rates are essential in a real-time system such as CAMTrack3D. The fifth performance measure, latency, refers to the time delay between when an event occurs and when the system reports the event. Large latencies in hand tracking cause the system to feel sluggish to the user. The sixth and final performance measure is positional noise. Positional Noise measures the amount of fluctuation due to noise that is inherent in any tracking system. Obviously, a large amount of positional noise is undesirable. CAMTrack3D will be evaluated using each of these performance measures throughout this section. In order to give a point of reference for these evaluations, similar evaluations will be performed on the pciBIRDTM magnetic tracking system currently in use in PARIS and compared to those of CAMTrack3D.

5.1. RESOLUTION

The worst-case scenario points are the points within the workspace which correspond to the greatest combined distance from the camera centers. At the worst-case scenario point, the single-pixel field of view for the virtual camera is larger than that of the real camera. The effective resolution was determined by projecting pixels into three-space. The worst-case resolution was determined to be 0.4027 inches. The best possible resolution within the workspace occurs at a point on the mirror near the top of the workspace, and was calculated to be 0.2516 inches in a similar manner. Ultimately, the resolution of CAMTrack3D is limited by the resolution of the image it is given, which is governed by its sensing device, the IBot IEEE 1394 camera. Currently CAMTrack3D is run at a relatively low resolution of 320 x 240, which is below the capability of the IBot (640 x 480). Increasing to this image resolution would effectively increase the resolution of CAMTrack3D by a factor of two. For comparison, the published specifications for the pciBIRD magnetic tracking system list the static resolution as 0.0196 in. This represents a much higher resolution by a factor of approximately 20.

5.2. ACCURACY

In order to test the accuracy of CAMTrack3D and the magnetic tracking system, tracking data was taken from 85 data points spread throughout the PARIS workspace. At each data point at least five trials were performed resulting in anywhere from 300 to 500 samples per data point for each tracking system. The data points were carefully laid out to cover the entirety of the PARIS workspace. In order easily to compare the reported position of a tracked object with the true position, the shape of the object must be well known. Using objects with symmetric properties helps ensure that the centroids of the reflection and the actual image of the object occur at the same point in space. This is why fluorescent paint bottles were chosen as test subjects for the accuracy test. The fluorescent paint bottles are 3 inches tall and have a diameter of 0.75 inches. The magnetic tracking system records the three-dimensional position of a small electromagnetic receiver device. This device was placed at all 85 data points in a similar manner to CAMTrack3D's test subject and data was collected.

Data was collected by running five trials per data point. This resulted in a collection of anywhere between 300 and 500 three-dimensional coordinate samples per data point. CAMTrack3D was reinitialized before each trial was taken in order to ensure reliable data was taken. The data was then analyzed and compared to the true location for each data point. Because the CamShift trackers steadily reported the same values, the three-dimensional position did not vary often. This reliable behavior was typical for most data points taken using CAMTrack3D. However, some of the most remote data points that appeared smallest in the image frame and were the least poorly lit had more varying three-dimensional sample coordinates.

The next set of figures presents a level by level analysis of the tracking errors recorded for CAMTrack3D and the magnetic tracking system. Figure 4(a) is a bar chart showing the errors reported between CAMTrack3D's mean tracker location and the true location of the test specimen for every point on the tabletop plane. The largest error reported at this level, a value of approximately 2.3 inches, occurs at the location of data point 20, one of the extreme boundaries of the PARIS workspace. CAMTrack3D's average error level at the tabletop plane is 1.04 inches. The magnetic tracking system's average error level, as seen in Figure 4(f) is 2.20 inches, over twice the value of CAMTrack3D. In general, CAMTrack3D exhibits more accuracy more consistently. At some points, the magnetic tracking system has errors over eight times larger than the corresponding CAMTrack3D error value. In fact, the magnetic tracking system only performs slightly better than CAMTrack3D at data points 3, 4, 8, and 9 within 6 in. from the tabletop. The magnetic tracker is less accurate in all other cases. This is significant because while the user may not necessarily make use of the full volume of the PARIS workspace, he/she will surely use an area larger than the 14 x 6 x 9 inch cube that the magnetic tracking system performs on a comparable level to CAMTrack3D.

An analysis of the error distance categorized by data points illustrates this point even further. The magnetic tracker only outperforms CAMTrack3D in 3 out of the 20 data points by a slight margin when averaged over height. At some data points, the average magnetic tracker error is over four times larger than that of CAMTrack3D.

5.3. RANGE OF OPERATION

The range of operation gauges the space in which the system can be relied on to operate without a significant increase in error. Considering the accuracy data from the previous section, it is safe to say that CAMTrack3D operates reliably within the limits of the PARIS workspace. The PARIS workspace occupies an approximate volume of 56 x 27 x 18 inches. The published technical specifications of the magnetic tracking system state that the effective range of operation is a sphere of 30 in. radius around the transmitter. This roughly corresponds with the findings from the accuracy test.

Thus, the volume associated with the range of operation for CAMTrack3D is approximately 2720 in^3. Because of the placement of the magnetic tracking system, only roughly one quarter of the entire range of operation is used. This makes the volume associated with the range of operation of the magnetic tracker within the PARIS workspace to be approximately 2830 in^3. The result is a slightly larger range of operation for the magnetic tracking system (4% larger). It also should be noted that the magnetic tracking system is also used to track the position of the head, as well as the position of the wand. The user's head is constantly outside of the PARIS workspace, but still within the range of the magnetic tracking system. CAMTrack3D's range of operation is dependant on the placement of the camera, the size of the mirror, and the resolution of the image. Because of

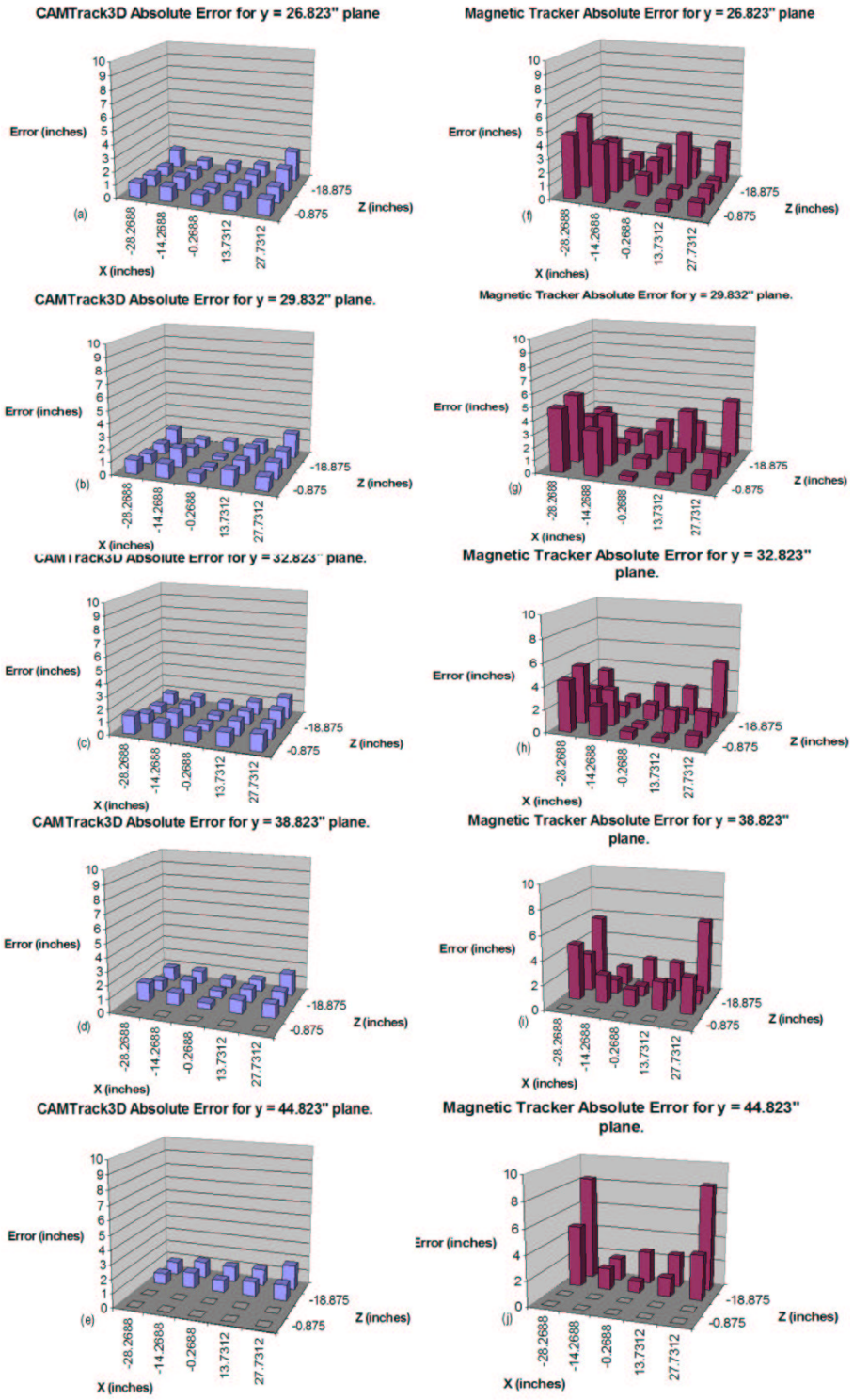

Figure 4. (a)-(e) CAMTrack3D accuracy test results for various constant-y planes. (f)-(j) Magnetic tracker accuracy test results for various constant-y planes.

this, increasing the size of the mirror and placing the camera further back from the mirror would significantly increase CAMTrack3D's range of operation.

5.4. UPDATE RATE

In order to measure CAMTrack3D's update rate a DirectX video rendering filter was used to measure the frame rate of the video stream. Several tests were run with the CamShift filters running, while the video rendering filter reported the frame rate and the number of dropped frames. The video stream with the CamShift filters averaged a frame rate of 29.99 Hz. There was only one dropped frame reported over the course of three tests. It should be noted that the test does not completely reflect the performance of CAMTrack3D, since the test was run on the CamShift filters without any three-dimensional calculations or error decisions. However, it should also be noted that the three-dimensional and error calculations are extremely inexpensive. For comparison, the magnetic tracking system is published to have an update rate of 105 Hz, over three times faster than CAMTrack3D. However, both systems can be considered real-time, as the general accepted minimum data rate for a real-time system is somewhere between 15 and 20 Hz.

5.5. LATENCY

The fifth performance measure, latency, refers to the time delay between when an event occurs and when the system reports the event. Large latencies in hand tracking systems cause the application to feel sluggish to the user. There are several possible sources for delays in a hand tracking system such as CAMTrack3D. There is a transport time for the data to be transmitted from the IBot camera to the IEEE 1394 card. The data must then be sent over the PCI bus into memory. The video data must then be moved in memory to the frame buffer, where it is operated on and finally displayed by CAMTrack3D.

A Direct Show Filter that creates an optical oscillator by video feeback was used to measure the overall latency of CAMTRrack3D. The camera was pointed at the display screen and the filter operates by looking at the image pixel data and determineing if the average RGB value is above or below 128. If it is above 128, the image is assumed to be white, and if it is below 128, black. Whenever the filter senses that the image is white, it writes all the pixels in the next frame buffer to black and vice versa. The camera is then focused at the monitor and what results is a feedback system, where the video image repeatedly changes from black and white. The time between changes is the latency.

A filter graph was created to determine the latency experimentally. The resulting latency inherent in CAM-Track3D was determined to be approximately 133.5 ms through several trials. The magnetic tracking system has a measured latency of 50 to 100 ms. The magnetic tracking system seems to suffer from a much higher latency effect when tracking sudden or quick movements.

5.6. POSITIONAL NOISE

Positional Noise measures the amount of fluctuation in the location data due to various noise sources. Large amounts of data fluctuation cause a positional noise effect in the virtual representation of the user's hand. This effect is disturbing to the user and makes interacting with the virtual environment difficult.

In order to measure the level of positional noise present in CAMTrack3D and the magnetic tracking system, an analysis of the data used for the accuracy test was performed. The standard deviation for CAMTrack3D's reported location position of each data point was determined and plotted in Figure 5. Clearly, the closer the test subject is to the mirror, the lower the level of positional noise. This is to be expected, as points nearer the mirror are closer to each cameras optical center, and thus have a higher resolution as discussed in Section 6.2. There is one large outlier in the data, at data point 6, 12 inches from the tabletop. This is due to the poor lighting conditions at a point near the edge of the workspace. The average positional noise level is highest at the tabletop level, with a value of 0.249 inches. It falls significantly at the 18-inch level to a value of 0.085 inches.

The magnetic tracking system performs much better by comparison. Figure 5 is a plot of the positional noise level present in the magnetic tracking system. It can easily be seen that the noise levels present in the magnetic system are indicative of the system's higher resolution. Just as the resolution of the magnetic tracking system is an order of magnitude greater than that of CAMTrack3D, the positional noise level is generally an order of

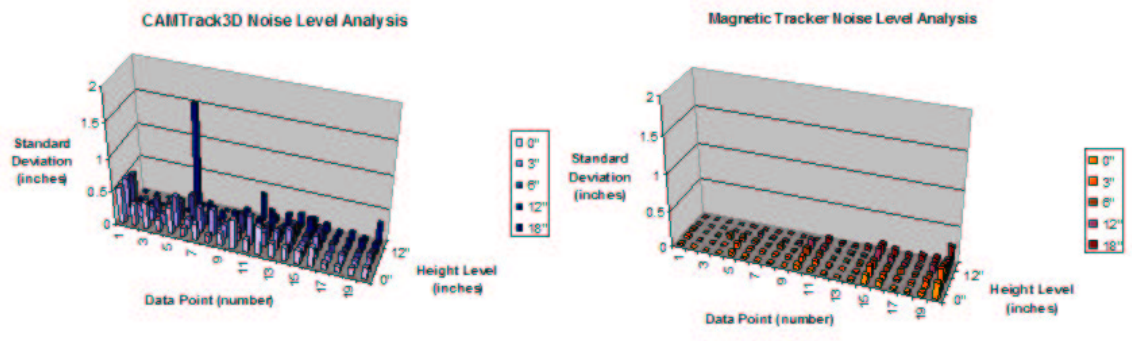

Figure 5. *(left)* Positional Noise level present in CAMTrack3D. *(right)* Positional Noise level present in the magnetic tracking system.

magnitude smaller. The average noise level for the tabletop plane is 0.042 inches. The highest average noise level was recorded at the 6-inch plane, a measure of 0.054 inches. Inspection of the figure can lead one to the conclusion that there is a possible source of interference somewhere between the x = 14 in. data points and the x = 18 in. data points. Data points 5, 10, 15, and 20 all have significantly larger noise levels than the rest of the points over every height level.

6. CONCLUSION

This paper has presented a real-time optical-based tracking system, CAMtrack3D, and has performed a subsequent analysis of the CAMtrack3D system and comparison to the current magnetic tracking system.

CAMTrack3D was determined to have superior accuracy to all but the closest data points to the origin. In some cases CAMTrack3D's accuracy was over eight times greater than that of the magnetic tracking system. However, the high latency of CAMTrack3D is an obstacle that still needs to be overcome. Additionally, the magnetic tracking system exhibited less noise.

The equipment cost of CAMTrack3D is an attractive feature. The equipment for the magnetic tracking system costs about three times as much as that required for CAMTrack3D. Another advantage of the CAMTrack3D system is that is untethered. The pciBIRD limits the user's range of motion and makes the VR experience less immersive. Another significant advantage of the system is that it does not suffer from interference from conductive materials. CAMTrack3D is also a more mobile system, as it can be easily set up or taken down in about 10 to 15 minutes.

CAMTrack3D also suffers from several disadvantages when compared to the magnetic tracking system. The largest and most important is that the magnetic tracking system provides six-dimensional tracking information. CAMTrack3D cannot provide orientation information in the current state, and VR application development must be adjusted to accommodate this deficiency. Also, CAMTrack3D tracking temporarily fails under occlusion, which is not a problem with the magnetic tracking system as it is not vision-based.

There is still much room for improvement, however. The most obvious and immediate improvement that can be made to CAMTrack3D is an increase in the resolution of the video stream. Such an increase would lower noise levels, increase the effective resolution, and most likely increase the overall accuracy. Also, correcting for lens distortion will yield more accurate tracking results.

REFERENCES

1. C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R.V. Kenyon, and J.C. Hart, "The CAVE: Audio visual experience automatic virtual environment, " *Communications of the ACM* vol. 35, no. 6, pp. 65-72, June 1992.

2. V. Kindratenko, "A Comparison of the accuracy of an electromagnetic and a hybrid ultrasound-inertia position tracking system," in *Teleoperators and Virtual Environments*, vol. 10, no. 6, pp. 657-663, 2001.
3. K. Meyer, H. L. Applewhite, and F. A. Biocca, "A survey of position trackers," *Presence*, vol. 1, no. 2, pp. 173-200, 1992.
4. E. Foxlin, M. Harrington, and G. Pfeiffer, "ConstellationTM: A wide-range wireless motion-tracking system for augmented reality and virtual set applications," in *Siggraph*, pp. 371-378, 1998.
5. A. Johnson, D. Sandi, G. Dawe, Z. Qiu, S. Thongrong, and D. Plepys, "Developing the PARIS: Using the CAVE to prototype a new VR display," in *CDROM Proceedings of IPT 2000: Immersive Projection Technology Workshop*, 2000, vol. 35, pp. 67-72.
6. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge University Press, 2000.
7. M. Nixon, B. McCallum, W. Fright, and N. Price, "The effects of metals and interfering fields on electromagnetic trackers," em Presence: Teleoperators and VEs, vol. 7, pp. 204-218, 1998.
8. G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," http://www.intel.com/technology/itj/q21998/articles/art_2.htm, May 1998.
9. A. Smith, "Color gamu transform pairs," *Siggraph 78*, pp. 12-19, 1978.
10. J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics Principles and Practice*. Reading, MA: Addison-Wesley, 1996.
11. Y Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790-799, June 1995.
12. R.I. Hartley and P. Sturm, "Triangulation," em Computer Vision and Image Understanding, vol. 68, no. 2, pp. 146-157, November 1997.
13. Zhengyou Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330-1334, June 2000.