A Method for Designing Marker-Based Tracking Probes

Larry Davis[1], Felix G. Hamza-Lup[2], and Jannick P. Rolland[1,2,3]
[1]Department of Electrical and Computer Engineering
[2]School of Computer Science
[3]College of Optics and Photonics: CREOL & FPCE
University of Central Florida
{davis@odalab.ucf.edu}

Abstract

Many tracking systems utilize collections of fiducial markers arranged in rigid configurations, called tracking probes, to determine the pose of objects within an environment. In this paper, we present a technique for designing tracking probes called the Viewpoints Algorithm. The algorithm is generally applicable to tracking systems that use at least three fiduciary marks to determine the pose of an object. The algorithm is used to create a integrated, head-mounted display tracking probe. The predicted accuracy of this probe was 0.032 $\pm$ 0.02 degrees in orientation and 0.09 $\pm$ 0.07 mm in position. The measured accuracy of the probe was 0.028 $\pm$ 0.01 degrees in orientation and 0.11 $\pm$ 0.01 mm in position. These results translate to a predicted, static positional overlay error of a virtual object presented at 1m of less than 0.5 mm. The algorithm is part of a larger framework for designing tracking probes based upon performance goals and environmental constraints.

1 Introduction

A current technique for tracking in virtual environments involves using fiduciary marks, or markers, to determine the pose of the objects of interest [18] [14]. These markers can be intrinsic to the object or placed upon the tracked object. Marker-based tracking has traditionally implied the use of optical tracking methods, as they typically provide the high accuracy required for Augmented Reality applications [2]. We expand the definition of a marker-based tracking system to include any tracking system that uses at least three distinct markers to determine the position and orientation of the object(s) tracked.

An issue when implementing a marker-based tracking approach (with external markers) is determining the configuration of the markers upon the object to be tracked. We refer to the collection of markers placed on the object to be tracked as a tracking probe. For tracking the simple motion of a rigid object with constant curvature (i.e. sphere), a minimum number of markers placed in a non-collinear, yet arbitrary fashion is sufficient. But, tracking applications involving anything more complex than the previously mentioned qualifications may require a larger number of markers in specific locations. For example, accurately tracking a translating, rigid cube requires only a minimum number of markers on a single side, while tracking the motion of fingers on a hand will require a larger number of carefully placed markers.

Another consideration when determining the configuration of a tracking probe is the range of motion the probe can undergo and still remain detected by the tracking system. This property is of increased importance if the tracking system suffers from line-of-sight issues. We refer to the amount of rotation a tracking probe can experience yet remain tracked as its field of regard. The field of regard for a tracking probe will depend upon the arrangement of the markers, as well as the angular extent through which individual markers can be rotated and be detected by the tracker. For active markers, the rotational extent is quantified by the cone of emission. For passive markers, the amount of rotation possible will depend upon a variety of factors, but may still be thought of as a cone of emission.

The concept of field of regard is illustrated in Figure 1. The large circle represents the object to be tracked and the black circles represent the markers. The lines extending from the black dots represent the extent of the cone of emission. The tracking probe is entirely contained within the field of

Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)
0-7695-2191-6/04 $20.00 © 2004 **IEEE**

view of the tracker, represented by a triangle enclosing the circles, with the apex at the structure representing the tracker. Assuming that the probe can rotate about an axis normal to the plane of the paper, the arrow in the figure represents the amount of rotation possible by the probe while the tracker is still able to detect three markers.

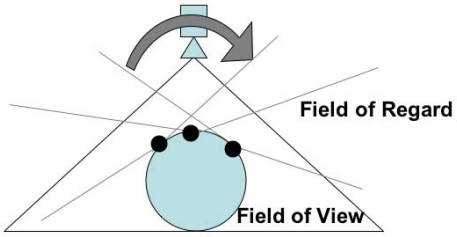

Figure 1. The Concept of Field of Regard

A tracking probe may have a larger field of regard if more markers are available for detection by the tracker. By adding enough markers, a tracking probe with a 360-degree field of regard in azimuth and elevation can be created. However, there is often a practical limitation on the number of markers that can be added to a tracking probe, whether it is the physical dimensions of the probe (not enough space) or the update rate with active markers (more markers means activation at a higher frequency) or tracking complexity (the system must distinguish more markers). In many cases, we wish to constrain the number of markers utilized.

In the end, we desire a tracking probe that conforms to the object being tracked, provides a large field of regard, uses a minimum number of markers, and provides an acceptable level of accuracy for the chosen application. We therefore introduce a method for configuring tracking probes that achieves these goals. This technique for determining the placement of markers when designing a tracking probe, or marker mapping, is a combination of ray tracing and an optimization method that minimizes the number of markers detected from different regions within the tracking volume. We refer to the technique as the Viewpoints Algorithm because optical tracking terms, markers must be "seen" from different "viewpoints."

In this paper, we describe the Viewpoints Algorithm for mapping markers on a tracking probe and its implementation in making an integrated, head-mounted display (HMD). We then present the results of numerical simulations of the performance of the tracking probe, followed by the experimental validation of the simulation results. Lastly, we discuss the results obtained and directions for future work.

2 Related Work

Within the biomechanics literature, there have been numerous efforts on how to place markers for optimal pose estimation. Techniques have been devised that address some of the problems associated with accurately tracking anatomical motion in a marker-based tracking system [15] [3] [1]. The work in this realm has extended to techniques for motion capture that use optimization [10]. However, these efforts are mostly focused on where to place markers to limit or correct the amount of relative motion between markers.

For rigid tracking probes, Morris and Donath quantified the cumulative effects of multiple error sources, including the effects of dynamic target array deformation errors [13]. Based upon the work presented in [19], a modified maximum error statistic was presented for determining the pose error for a given tracking probe configuration.

Davis, et al examined the use of global optimization techniques for placing markers on a tracking probe to obtain a 360-degree field of regard tracking probe [6]. The method used simulated annealing to minimize a cost function based upon the distance between neighboring markers on the probe [12] [7]. This was one of the first attempts at providing a formalized method for designing tracking probes to meet the requirements for a specific application.

Similarly, Vogt et al implemented a method for designing tracking probes using a Monte Carlo simulation technique [17]. The design methodology minimized the jitter error associated with the tracking probe, using the probe radius, marker heights, and number of markers as input variables. Because Tsai's calibration technique [16] was utilized to determine the pose of the tracking probe, the method required probe topologies with at least seven, simultaneously detected markers.

More recently, Davis et al proposed a method for predicting static errors in pose for any given tracking probe configuration [5]. The method used a first-order error propagation technique to apply the errors from the position estimates of individual markers to the overall pose estimation. In this work, the authors were able to predict the performance of a semi-spherical tracking probe and hinted at general trends for probe design.

The work presented in this paper is a significant extension of the research presented in [9] and [5].

Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)
0-7695-2191-6/04 $20.00 © 2004 **IEEE**

Specifically, we provide a technique for marker placement that can be customized to user-specific requirements. Indeed, the ability to predict the accuracy of a given probe design and to adjust the design in a regimented manner are key components for a formalized framework for designing and assessing tracking probes.

3 The Viewpoints Algorithm

We now present the details of the Viewpoints Algorithm. The Viewpoints Algorithm is based upon ray tracing fundamentals with additional constraints for marker detection and field of regard [8]. The algorithm assumes that the markers utilized are external to the object to be tracked. We also assume that the cone of emission (as defined in Section One) for either passive or active markers is known and that the probe is entirely within the working volume of the tracker. In addition, we assume that a 3D triangular model of the object to be tracked exists.

In general, marker-based trackers must detect at least three markers to determine the pose of a tracked object. Therefore, we attempt to place markers upon an object to meet this requirement. From the perspective of the tracker, the markers are detected or "in view". Knowing the point of origin and the extent of the tracking volume, i.e., the tracker "view", we can reverse the problem; instead of placing many markers on the object to conform to one "view", we could place many "views" around the object and determine where to place a marker. We can achieve this reversal by using the 3D model of the object and placing it at the center of an intermediary sphere. The sphere would then have the "views", which we call virtual viewpoints (or just viewpoints), represented by points optimally distributed on its surface using a global optimization technique [6] or distributed according the desired field of regard. For example, if a 180-degree field of regard was desired, viewpoints could be equally spaced on the surface of the intermediary sphere across a solid angle of 180 degrees.

By counting the number of times a polygon is "seen" by one of the virtual viewpoints, we can determine the optimal placement of markers. The centers of the polygons that are "seen" most often are chosen to become the marker locations.

To create a tracking probe using the viewpoints algorithm, we start by distributing the viewpoints on the intermediary sphere surrounding the 3D model of the object. The intermediary sphere has a radius that is twice the maximum distance between two vertices in the triangular mesh of the 3D model to ensure enclosure. For each viewpoint, we then determine whether a marker placed on any of the triangles in the 3D model could be "seen" by the viewpoint. The determination is based upon the angle between each triangle normal and the vector from the center of the triangle to the viewpoint in question. If the angle is less than half of the field of emission of a marker, then the triangle can be "seen" and the number of viewpoints that can "see"

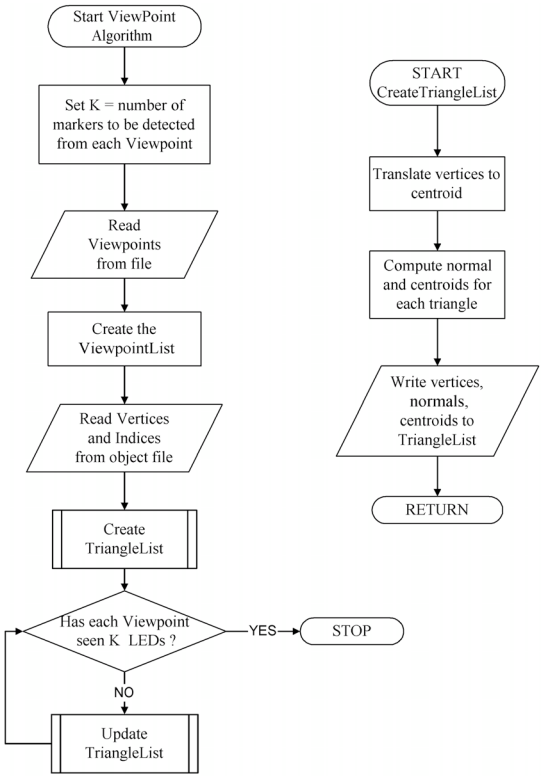

Figure 2. The Viewpoints Algorithm

the triangle is incremented by one. This test is applied to all the triangles in the model with respect to the same viewpoint. The process of testing for triangles which can be "seen" is repeated for all viewpoints on the intermediary sphere. Once all the viewpoints have been tested, the triangle which has the highest viewpoint count is selected as a final marker location and is removed from further consideration. The marker is mapped at the center of gravity of the triangle, although any point within the triangle will satisfy the "visibility" constraint. Each viewpoint that can "see" this triangle also has its marker count incremented. Once a viewpoint can "see" K triangles (its marker count equals K), it is removed from consideration also. The process for determining how many triangles are seen

by the viewpoints is continued until all viewpoints "see" K triangles. The final marker mapping is then saved and used for further analysis.

A potential problem with the Viewpoints Algorithm exists when dealing with complex objects, that is, objects with cavities or objects that cannot be defined by simple mathematical functions or simple parametric equations. For example, if we were trying to map markers on a 3D "U," there would be triangles that satisfy the "seen" requirement due to their normals, but would be "hidden" because they were located on a part of the "U" that was obscured by another portion of the letter. This is the situation that occurs if one imagines holding the "U" upright in one's hand, rotating it 90 degrees to the right, and trying to view the inner portion of the "U." The solution to this problem lies in using the parametric equation of the line passing from the viewpoint through the center of the triangle being considered. Once we have determined that a triangle can be "seen" by the viewpoint, we check to see if the line intersects any other triangles. If intersections occur beyond the first triangle, we check the angle criteria for each additional triangle intersected by the line. If we determine that the additional triangles pass the "visibility" test, we determine the point of intersection of the line for these triangles. We then substitute the intersection points for each of the "visible" triangles that intersect the line (including the triangle we started with) into the parametric equation and solve for the linear parameter. The intersection point that gives the smallest linear parameter is inside the triangle that is closest to the viewpoint, and, therefore, the triangle that is actually "seen." The other triangles that gave false "visibility" tests are removed from further consideration for the given viewpoint. Flowcharts illustrating the virtual viewpoint algorithm and the method for updating the number of times a triangle is "seen" are given in Figures 2 and 3, respectively.

The benefits of the Viewpoints Algorithm are threefold. First, this algorithm minimizes the number of markers used for the tracking probe. Second, it guarantees that at least K markers are visible from each viewpoint, where K is the minimum number of markers that are necessary for different tracking systems. Finally, a tracking probe with a custom, large field of regard can be created by increasing the number of viewpoints and arranging the viewpoints to emphasize a particular direction (e.g. putting the viewpoints at the front of an HMD).

The cost function of the Viewpoints Algorithm maximizes the number of times a triangle is "seen." In this sense, the algorithm will minimize the number of markers and achieve the desired field of regard. A drawback of the Viewpoints Algorithm, however, is its sensitivity to the tessellation of the 3D model. For example, a flat surface can be represented with relatively few polygons, while a complex, curved surface requires many polygons. If a complex feature on the 3D model is prominent, then many markers may get mapped in a very small area when one marker may be sufficient. Thus, a different mapping algorithm may provide better results in this case.

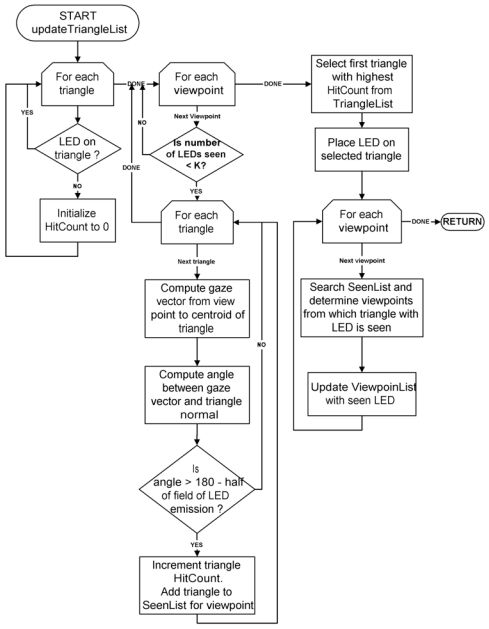

Figure 3. Detail of the UpdateTriangles Routine

The geometry of an integrated head tracking probe was designed using a 3D model obtained from CAD data. Using the Viewpoints Algorithm, we placed 11 markers on the front surface of the HMD. The specific viewpoints were spread at 45 degree increments on a 90 degree solid-angle wedge of an intermediary sphere. The final mapping of the markers is shown in Figure 4. The marker locations are the small dots on the shell of the HMD, while the viewpoints are the white squares surrounding the shell. The lines from the squares to the dots indicate markers that are "seen" from a particular viewpoint.

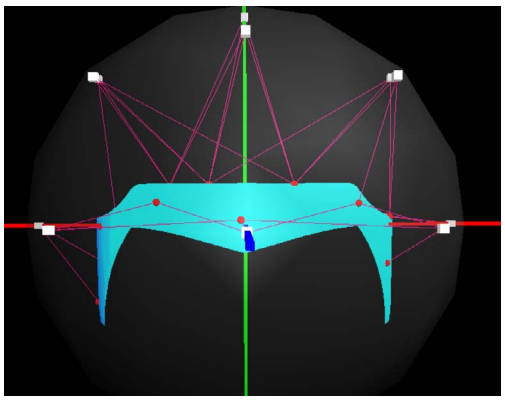

Figure 4. VRML Visualization of a Conformally Mapped Tracking Probe

4 Simulation of the Tracking Probe

After configuring the markers for a given tracking probe, we then seek to simulate the performance of the probe. In this manner, we can predict whether the probe will meet the specification for accuracy *a priori* and make adjustments in marker locations as needed while maintaining constraints on field of regard and the number of markers.

The simulation of tracking probes models the pose estimation process. The first step in this process is to determine the location of the markers within the local frame of the tracking probe, $\mathbf{x}_k$. The local marker positions are given by the marker mapping procedure, the schematics of the proposed tracking probe, or from user-created data files that specify a tracking probe. However, these locations do not have any errors associated with them when produced by the mapping algorithms. For example, the Viewpoints Algorithm uses a 3D model of an object that is composed of triangles to place the markers. When a particular triangle has been selected for a marker location, the marker is placed at the center of gravity of the triangle.

In practice, however, the marker location can only be approximated, usually to the size of the corresponding triangle within the 3D model. In addition, once the markers are placed on the object, the local coordinate frame may be changed, that is, recomputed based upon convenience. Therefore, we must add errors to these local marker locations to properly simulate the construction of a tracking probe. The magnitude of the errors can be determined based upon manufacturing data (if the probe is already physically constructed) or based upon the relative size of the triangles used in the 3D model. The errors in determination of the local frame, $\Delta\mathbf{x}_k$, are simulated by applying zero-mean Gaussian noise to the coordinates using the following procedure:

1. Determine the amount of noise to apply to local coordinates.
2. Seed the random number generator.
3. Using three uniform random numbers, determine a normalized noise vector for each marker.
4. Scale the normalized noise vector magnitude using a Gaussian random number. The Gaussian distribution used has zero-mean and a standard deviation equal to the noise amount chosen in Step 1.
5. Add the noise vectors to local marker locations.

Once we have simulated the local coordinates, we simulate the locations of the markers in the tracker frame of reference, $\mathbf{y}_k$. However, we must know where the probe is to be simulated within the tracking volume because the accuracy of the tracker may vary according to the probe location within the volume. Thus, we apply a rotation, $\mathbf{R}$ and translation, $\mathbf{t}$ to the local marker locations (with noise) to obtain the marker locations with respect to the tracker, also called the global marker locations. The rotation is specified in a matrix format, due to the formulation of the noise propagation procedure specified in [5]. Finally, based upon the accuracy of the tracker in that region, we add zero-mean Gaussian noise, $\Delta\mathbf{y}_k$ to the global marker locations using the procedure previously defined for the local marker coordinates.

We now have noisy and noise-free representations of the probe markers in the global and local coordinate frames. The next step is to estimate the amount of pose error. The detailed procedure for propagating the effects of marker noise to the pose estimation is described in [5], however we summarize the main results here. The errors on $\mathbf{y}_k$ and $\mathbf{x}_k$ are expressed as

$$\begin{aligned} \mathbf{x}_k &\rightarrow \mathbf{x}_k + \Delta\mathbf{x}_k \\ \mathbf{y}_k &\rightarrow \mathbf{y}_k + \Delta\mathbf{y}_k \ . \end{aligned} \tag{1}$$

These errors are then propagated to the pose estimation for the tracking probe. We can express the overall pose with error as

$$\begin{aligned} \mathbf{R}_{err} &\approx \Delta\mathbf{R}\mathbf{R} \\ \mathbf{t}_{err} &\approx \mathbf{t} + \Delta\mathbf{t} \ , \end{aligned} \tag{2}$$

where an equation of the form $a \approx b$ is taken to mean that a and b are approximately equal to a first order approximation.

From [5], we know that $\mathbf{R} = \mathbf{V}\mathbf{D}\mathbf{U}^T$, with $\mathbf{U}$ and $\mathbf{V}$ resulting from the SVD of $\mathbf{H}$, the spatial covariance matrix between $\mathbf{y}_k$ and $\mathbf{x}_k$, and $\mathbf{D}$ given by

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} . \quad (3)$$

Therefore, to determine $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$, we must determine how the marker errors affect $\mathbf{H}$, which in turn is affected by $\mathbf{U}$ and $\mathbf{V}$. We propagate the error to $\mathbf{R}$ through the matrices $\mathbf{V}$ and $\mathbf{U}$ by

$$\begin{aligned} \mathbf{V} &\rightarrow \mathbf{V} + \Delta\mathbf{V} \\ \mathbf{U} &\rightarrow \mathbf{U} + \Delta\mathbf{U} . \end{aligned} \quad (4)$$

The errors applied to $\mathbf{U}$ and $\mathbf{V}$ can be defined as a transformed version of the original $\mathbf{U}$ and $\mathbf{V}$ matrices. To represent the transformations, we define two matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$\begin{aligned} \Delta\mathbf{V} &\approx \mathbf{A}\mathbf{V} \\ \Delta\mathbf{U} &\approx \mathbf{B}\mathbf{U} . \end{aligned} \quad (5)$$

The procedure for determining A and B is given in [5], where it can be noted that matrices $\mathbf{A}$ and $\mathbf{B}$ are anti-symmetric ($\mathbf{A} = -\mathbf{A}^T$). Given $\mathbf{V}$, $\mathbf{U}$, and $\mathbf{D}$, it can be shown that

$$\begin{aligned} \mathbf{R}_{err} &\approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}} \\ &\approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}I \\ &\approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{R}^T\mathbf{R} \\ &\approx \underbrace{e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{R}^T}_{\Delta\mathbf{R}}\mathbf{R} . \end{aligned} \quad (6)$$

Thus, we can compute the amount of rotational error, $\Delta\mathbf{R}$, introduced into a pose calculation by noise in the marker data as

$$\Delta\mathbf{R} = e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{R}^T . \quad (7)$$

If we define the centroids of the "ideal" local and global coordinates as $\mathbf{x}$ and $\mathbf{y}$, and the centroids of the local and global coordinates with error as $\mathbf{x}_c$ and $\mathbf{y}_c$, we can compute the translational error, $\Delta\mathbf{t}$, introduced into a pose calculation as

$$\begin{aligned} \Delta\mathbf{t} &= \mathbf{t}_{err} - \mathbf{t} \\ &= \mathbf{y}_c - \Delta\mathbf{R}\mathbf{R}\mathbf{x}_c - \mathbf{y} + \mathbf{R}\mathbf{x} \end{aligned} \quad (8)$$

An algorithm has been created and implemented to simulate this procedure. The inputs to the algorithm are the local and global marker coordinates with noise, the standard deviation of the Gaussian noise applied to the local and global marker coordinates, and the initial, error-free rotation matrix that defines the orientation of the tracking probe. The output of the algorithm is the incremental rotation matrix, $\Delta\mathbf{R}$ that transforms the tracking probe from its initial, error-free orientation to the orientation obtained when noisy marker data are used. From the $\Delta\mathbf{R}$, we compute the average pose error using a sum of squared errors formulation

$$\Delta P^2 = \frac{1}{K}\sum_{i=1}^{K} ||(\mathbf{R}_{err}\mathbf{x}_k + \mathbf{t}_{err}) - (\mathbf{R}\mathbf{x}_k + \mathbf{t})||^2 . \quad (9)$$

In this expression, ΔP is the pose error in millimeters, $\mathbf{R}_{err}$ is the concatenation of the initial, error-free orientation and the incremental rotation matrix ($\Delta\mathbf{R}\mathbf{R}$), the $\mathbf{x}_k$ are the noise-free local marker locations, and $\mathbf{t}_{err}$ is the sum of the initial position of the tracking probe and the translation between the noise-free local maker centroid and the noisy global marker centroid.

We can also determine the rotational and translational components of the pose error. The amount of error in rotation, ΔP_{rot}, can be determined from a concatenation of two quaternions, expressed as

$$q_{result} = q{q_{err}}^{-1} , \quad (10)$$

where q is the unit quaternion representing the error-free orientation obtained from $\mathbf{R}$, and q_{err} is the noisy orientation obtained from $\mathbf{R}_{err}$. The amount of error in rotation is equal to the angle extracted from the scalar portion of q_{result}. The error in position, ΔP_{pos}, is expressed as

$$\Delta P_{pos} = ||\mathbf{t}_{err} - \mathbf{t}|| . \quad (11)$$

In the course of the simulation process, we determined the value of ΔP, ΔP_{rot}, and ΔP_{pos}.

The independent variables within the simulation were the tracker noise, the probe size, and, the number of markers on the probe. The dependent variables we studied were the overall pose error in millimeters, the orientation error in degrees, and the position error in millimeters. The tracker noise and probe size were incremented on a logarithmic scale. The number of markers is incremented arithmetically when a generic tracking probe shape is used. Currently, the simulation supports generic spherical and planar tracking probes with up to 25 markers.

At each increment of the independent variable, the probe was simulated N times ($N \geq 20$) and the pose error results were averaged. The average at each increment was then plotted on a linear or log scale. When plotted on a linear scale, the standard

deviation of the N analyses was included at each data point as error bars. Finally, a curve fitting was performed with the data.

4.1 Simulation Specifics

The simulation of tracking probes and pose errors was implemented in MATLAB. The simulation was built with a graphical user interface to allow for easier simulation control and easier adjustment of simulation parameters. Aside from the dependent and independent variables, we were able to vary the local noise of the probe markers, the increment applied to the independent variables, the number of iterations used to compute the average errors, the display of the results, and the initial probe position within the tracking volume.

The POLYFIT function was used to determine the coefficients of a polynomial curve that fits the data in the least squares sense. The POLYVAL function was used to obtain error estimates on the predictions from the polynomial. The error bounds assume that the data were independent, normally distributed, and with constant variance. The graphs were then generated with the PLOT and ERRORBAR functions. The central line corresponds to the results of a 4th order polynomial least-squares curve fit. The lines above and below the central line represent the 95% confidence interval for the data points plotted on the graph. A 4th order polynomial was chosen as opposed to an exponential function because a better fit was obtained when including the data for very small or very large values of the independent variable. Higher orders of magnitude of noise were chosen for the purpose of displaying the conditions where the assumptions of the model were violated, i.e., the order of magnitude of the noise was equal to that of the probe size. The r-squared value for the 4th order curve fit ranges between 0.62 and 0.75 when a probe was simulated. If we were to limit the independent variable to the range log(0.01) to log(100), an exponential curve fit would yield r-squared values of greater than 0.7 in all instances.

Within the simulation we used the EXPM function to compute the matrix exponential. This function uses a Padé approximation rather than a Taylor Series approximation. We also used the RAND and RANDN functions to generate uniform and Gaussian random variables, respectively. The random number generators were seeded using the system time.

The local marker noise value for the probe was set at 0.1 mm to coincide with the stereolithography manufacturing data for the accuracy of marker placement. The simulation of this probe predicted

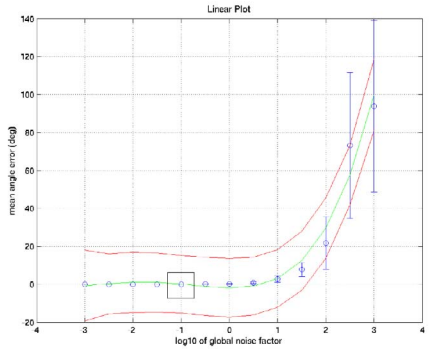

Figure 5. Simulation Results of the HMD Tracking Probe in Orientation

an error of 0.032 $\pm$ 0.02 degrees in orientation about the y-axis , shown in Figures 5 and 6, with an accuracy of 0.14 $\pm$ 0.08 mm in translation. The black rectangle shown in Figure 5 indicates the area displayed in Figure 6, which is a magnified view of the region within the rectangle . The plots shown were simulated using different global noise values for the HMD probe configuration. The simulation also predicted an error of 0.095 $\pm$ 0.05 degrees in orientation about the x-axis, which is not shown.

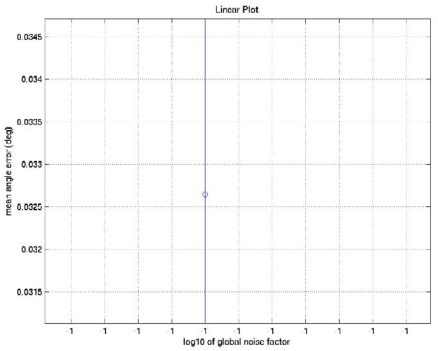

Figure 6. A Magnified View of the Simulation Results in Orientation

5 Experimental Validation

5.1 Placement of Markers on a HMD

One of the outputs of the Viewpoints Algorithm is a list of the marker locations expressed in the lo-

cal frame of the 3D model. Normally, these marker locations would then be integrated into the process of constructing the tracking probe. However, in this particular case of mapping the markers on an HMD, the object was pre-existing. So, instead of the marker being added as part of the mechanical design, it was necessary to approximate the marker locations and place them on the HMD by hand.

The markers were placed physically on the HMD with the assistance of a software package called GeoMagic from Raindrop Software. In this package, the ability is present to determine the distance between specified points on a 3D model. During the Viewpoints mapping technique, we chose the local coordinate system to be coincident with that of the 3D model. Therefore, when viewing the model in GeoMagic we were then able to determine the location of prominent features relative to the origin of the local coordinate system, then determined the distance from a given feature to a marker location. The procedure was done by hand using a digital caliper, so additional errors may be present. In future designs, these errors can be minimized by creating the HMD prototype with colored regions (whose locations are determined from the Viewpoints Algorithm *a priori*), indicating where the markers should be placed. The conformally mapped HMD tracking probe is shown in Figure 7. The +z-axis is in the gaze direction of the HMD (out of the page). The +y-axis is in the direction toward the top of the page. Along the x-direction, the HMD is approximately 33 cm in length.

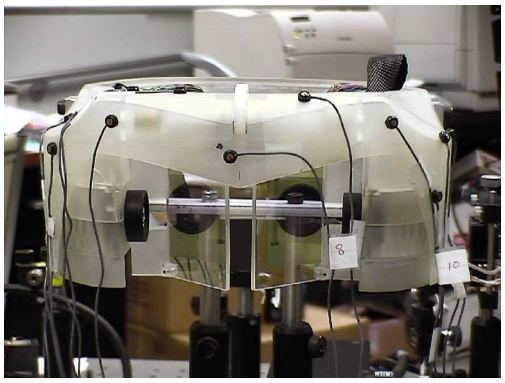

Figure 7. Realization of a Conformally Mapped Tracking Probe

5.2 Measurements

The tracking system used for the experiments was an OPTOTRAK 3020 from Northern Digital, Inc. The tracker has an accuracy of 0.1 mm at a distance of 2.25 m. The noise pattern of a single maker was measured and closely approximated a Gaussian distribution. For the orientation and position measurements, the HMD was mounted on motion stages that were fixed to an optical breadboard. A motorized rotation stage with an accuracy of $\pm$ 0.001 degrees was used to measure orientation about the y-axis . A motorized translation stage accurate to $\pm$ 0.01 mm was used to measure position. While the simulation predicted an accuracy of 0.09 degrees about the x-axis, the corresponding measurements were not attempted at this time because of the lack of a high-precision tilting platform and the difficulty of mounting the HMD in an orientation perpendicular to the existing motorized rotation stage.

In the experiment, the initial orientation of the HMD was recorded from the tracker. It was then rotated by 10 degrees about its y-axis and the final orientation was recorded. We then computed the quaternion corresponding to the rotation between the initial and final orientations of the HMD and extracted the amount of rotation. This procedure was repeated 100 times and the results were averaged. A similar procedure was performed using a 10 mm translation with the HMD facing the tracker. We measured an accuracy of 0.028 $\pm$ 0.001 degrees in orientation for rotation about the y-axis and 0.2 $\pm$ 0.005 degrees about the x-axis. The measured accuracy in position was 0.11 $\pm$ 0.01 mm.

6 Discussion and Conclusions

In this work, we have introduced a novel marker mapping technique for the design of conformal tracking probes, which are probes designed according to the requirements and constraints of the application environment. The tracking pose error results achieved point to the promise of this approach for use in AR environments. To achieve a registration error of 1 mm, the angular error for a tracking probe must be less than 0.057 deg [11]. With the design presented, we have obtained a static accuracy of approximately 0.03 degrees, which is on the order two times better than required. The prediction of 0.09 degrees in accuracy about the x-axis would yield a registration error of approximately 1.5 mm, which is still quite good.

We expect the tracking probe to be more accurate in rotation about the y-axis due to the arrange-

ment of the markers. If we consider an ellipsoid surrounding the markers on the HMD, we would see that the major axis would roughly correspond to the x-axis. One of the minor axes would roughly correspond to the y-axis. Because the spatial extent of the markers is greater along the major axis, the tracking probe will be more sensitive to rotations about an orthogonal axis. This phenomenon as applied to marker-based tracking is discussed in [5] and is mathematically derived in [4]. This will be experimentally verified with the HMD tracking probe in future work.

The Viewpoints Algorithm not only finds optimal positions for the markers, but it also minimizes the number of markers needed to fulfill a given field of regard constraint. The algorithm is not foolproof, though, because of inconsistencies with the function which tests for whether a triangle is seen or not. In this test, the triangle normal is compared to the vector from the viewpoint being tested. If the resulting dot product between the two vectors is negative, then the triangle can be "seen." Unfortunately, if one imagines a viewpoint on one side of the letter "M," there may be up to three triangles along a ray cast from the viewpoint through the "M" that may pass the test. We shall expand the algorithm to make use of the Z-buffer to resolve the ambiguity encountered in this scenario.

The Viewpoints Algorithm also relies upon having a 3D model of the object upon which the markers are to be mapped. However, within the process of mapping markers on an object, there are potential issues concerning the model itself. The problem lies in the triangle density of the 3D model used to represent the real object. If a 3D model has a large number of small triangles, then the triangle density is high. As such, a high triangle density is desirable because it would be a more accurate representation of the object that will have markers placed upon it. Still, the Viewpoints Algorithm may end up mapping many markers in quite a small area. A solution to this issue is to apply a minimum distance constraint to the algorithm during the mapping process. In addition, improvements in the computational efficiency of the algorithm can be implemented, although speed of execution is not currently an issue in the probe design process..

When simulating the pose estimation process for any new tracking probe, errors may occur from numerical inaccuracies. First, the pose error formulation uses matrices instead of quaternions to represent rotations. Matrices require more numbers to represent an equivalent quaternion rotation, therefore matrix operations are more subject to numerical drift when used for comparable transformations. Also, the marker noise for a given tracking system may not be well-approximated by a Gaussian distribution. Finally, because random distributions are generated each time the simulation is run, there will be variability in the results. At small noise levels, the difference is negligible, but larger noise levels induce greater variability. In the future, the tradeoff between static noise vectors and dynamically generated noise must be examined in comparing conditions.

Practitioners usually take a bottom-up approach to building applications or systems in virtual environments, assembling pieces to create a larger whole. The result is often a "Frankenstein" system that may accomplish the desired task, but is inelegant and causes users to conform to it. As humans, our systems and applications should conform to us, not vice versa. Moreover, an elegant design has advantages that may include improved ergonomics and the potential for enhanced performance. Thus, an added benefit of this framework is to promote a top-down design method for marker-based tracking, which allows exploration of the design space and ultimately designing to the specifications of the application.

Acknowledgments

The authors thank Eric Clarkson for discussions regarding the mathematics presented and Paulius Micikevicius for discussions regarding the implementation of the Viewpoints Algorithm. The work presented was funded by the U.S. Army Simulation, Training, and Instrumentation Command (STRICOM), the Office of Naval Research under Grant N00014-03-1067, and the Florida Education Fund.

References

[1] T. Andriacchi, E. Alexander, M. Toney, C. Dyrby, and J. Sum. A point cluster method for in vivo motion analysis: Applied to a study of knee kinematics. *Journal of Biomechanical Engineering*, 120(12):743–749, 1998.

[2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 20001.

[3] A. Cappello, A. Cappozzo, P. La Palombara, L. Lucchetti, and A. Leardini. Multiple anatomical landmark calibration for optimal bone pose estimation. *Human Movement Science*, 16:259–274, 1997.

[4] L. Davis. *Conformal Tracking for Virtual Environments*. PhD thesis, University of Central Florida, 2004.

[5] L. Davis, E. Clarkson, and J. Rolland. Predicting accuracy in pose estimation for maker-based tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '03)*, pages 28–35, October 2003.

[6] L. Davis, J. Rolland, R. Parsons, and E. Clarkson. Methods for designing head-tracking probes. In *Proceedings of the Joint Conference on Information Sciences (JCIS)*, pages 498–502, March 2002.

[7] K. Dowsland. Simulated annealing. In C. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*. Wiley & Sons, 1993.

[8] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1996.

[9] F. Hamza-Lup, L. Davis, C. Hughes, and J. Rolland. Marker mapping techniques for augmented reality. In *Proceedings of the International Symposium on Computer and Information Sciences (ISCIS 2002)*, pages 152–156, October 2002.

[10] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings of Computer Animation '2000*, pages 77–83, 2000.

[11] R. Holloway. *Registration Errors in Augmented Reality Systems*. PhD thesis, University of North Carolina - Chapel Hill, 1995.

[12] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[13] T. Morris and M. Donath. Using a maximum error statistic to evaluate measurement errors in 3d position and orientation tracking systems. *Presence: Teleoperators and Virtual Environments*, 2(4):314–343, 1993.

[14] J. Rolland, L. Davis, and Y. Baillot. A survey of tracking technology for virtual environments. In *Augmented Reality and Wearable Computers*, chapter 3. Erblaum, 2000.

[15] I. Söedrkvist and P. Wedin. Determining the movement of the skeleton using well-configured markers. *Journal of Biomechanics*, 26(12):1473–1477, 1993.

[16] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[17] S. Vogt, A. Khamene, F. Sauer, and H. Niemann. Single camera tracking of marker clusters: Multiparameter cluster optimization and experimental verification. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02)*, pages 127–136. IEEE and ACM, September 2002.

[18] G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications, special issue on Tracking*, 22(6):24–38, 2002.

[19] H. Woltring, R. Huiskes, A. De Lange, and F. Veldpaus. Finite centroid and helical axis estimation from noisy landmark measurements in the study of human joint kinematics. *Journal of Biomechanics*, 18(5):379–389, 1985.