

Tracker Calibration using Tetrahedral Mesh and Tricubic Spline Models of Warp

Christoph W. Borst
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330
cborst@cacs.louisiana.edu

Abstract

This paper presents a three-level tracker calibration system that greatly reduces errors in tracked position and orientation. The first level computes an error-minimizing rigid body transform that eliminates the need for precise alignment of a tracker base frame. The second corrects for field warp by interpolating correction values stored with vertices in a tetrahedrization of warped space. The third performs an alternative field warp calibration by interpolating corrections in the parameter space of a tricubic spline model of field warp. The system is evaluated for field warp calibration near a passive-haptic panel in both low-warp and high-warp environments. The spline method produces the most accurate results, reducing median position error by over 90% and median orientation error by over 80% when compared to the use of only a rigid body transform.

1. Introduction

Electromagnetic position and orientation trackers are widely used to track motion for virtual environments and other applications. Among their strengths are small sensor size, lack of line-of-sight problems, and relatively low cost. Unfortunately, magnetic fields used by these trackers become distorted in the presence of metal objects due to eddy currents and ferromagnetism [1], and warp is also found in metal-poor environments [2]. This results in error in tracked position and orientation.

To compensate for field warp, various calibration techniques have been proposed that compute correction values based on a set of measured calibration points for which both tracked and correct positions are known. Typically, the calibration points are measured along a regular grid during a data acquisition phase, although unstructured grids have also been used. Even when calibration points are measured on a grid that is regular in

real space, the corresponding grid in tracked (warped) space is not regular.

Global methods for calibration use global functions that compute correction value as a function of the reported value being calibrated. The function forms can be chosen to be smooth and continuous across the tracked space, but they typically cannot handle local behavior gracefully. For example, high-order polynomials have been found to produce undesirable oscillations and can introduce additional error [3], but low-order polynomials clearly cannot capture much detail.

Local methods compute a correction based on nearby calibration values in tracked space. Approaches such as that in [2] resample calibration points into a regular grid to create a lookup table for trilinear interpolation of correction values. To avoid the error introduced by resampling, correction values can instead be computed directly from the original calibration data by finding surrounding calibration points and computing a weighted sum using one of various weighting function [4-6]. Compared to global methods, existing local methods can better handle local detail but do not model field warp as smoothly. For example, linear interpolation from a lookup table or polyhedron vertices results in a discontinuous gradient at cell boundaries that does not correspond to real field behavior.

Some approaches calibrate orientation in addition to position, treating orientation error as a function of position. The techniques are analogous to those for calibrating position. For example, Kindratenko and Bennett [4] describe a procedure for computing a correction using a weighted sum of error quaternions stored with nearby calibration points, and Ikits et. al. [7] use a polynomial fit to compute error quaternions.

Section 2 presents a new position and orientation calibration system that uses both local and global methods with calibration points gathered on a regular grid in real space. The local method searches a tetrahedrization of the warped space and computes a correction from vertices of a surrounding tetrahedron. The global method uses interpolating cubic splines with C^2 continuity to model

warp as a piecewise tricubic deformation of real space. Although the spline form does not have local control, local details are handled reasonably because their influence on distant segments is slight. In contrast to the use of global functions in other systems, e.g., [3, 5, 7], this global approach does not use a fitting routine to compute curve coefficients that map from tracked coordinates to calibrated coordinates. Instead, the curve coefficients map from correct space to warped space, and an inverse mapping is found for each point at the time it is calibrated. The system also includes procedures for removing the dependence on human judgment in alignment of the tracker base (Section 2.1) and sensor coordinate system (Section 2.4).

In Section 3, the system is evaluated for calibration of a region of space in front of a passive-haptic panel that must be accurately registered with its visually virtual counterpart. A low-cost calibration apparatus using LEGO blocks was developed for this purpose, and its accuracy is also evaluated.

2. Calibration System

2.1 Error-Minimizing Rigid Body Transform

The first level of correction provided by the tracking system is an error-minimizing rigid body transform. This approach is sufficient if shape distortions are minimal or if a user wishes to calibrate only for unknown position or orientation of the tracking system base. The author has also used it to find transforms between coordinate systems of multiple tracking systems and in combination with the other calibration methods as described in Section 3.1.

Input to the procedure consists of a set of n calibration points described by tracked coordinates $\{\mathbf{p}_{i,tracked}\}$ and corresponding correct coordinates $\{\mathbf{p}_{i,correct}\}$. The coordinate system in which the $\mathbf{p}_{i,correct}$ are expressed need not be coincident with the tracker coordinate system, although the scale should be similar. The procedure finds a rigid body transform \mathbf{T}_{RB} that minimizes the sum of squared distances:

$$\sum_{i=0}^{n-1} \left\| \mathbf{p}_{i,correct} - \mathbf{T}_{RB} \mathbf{p}_{i,tracked} \right\|^2. \quad (1)$$

For a given orientation, the translation that minimizes the sum of squared distances is that which moves the centroid of $\{\mathbf{p}_{i,tracked}\}$ to the centroid of $\{\mathbf{p}_{i,correct}\}$. So, a solution that considers both orientation and translation in minimizing (1) has the form $\mathbf{T}_{RB} = \mathbf{T}_2 \mathbf{R} \mathbf{T}_1$, where \mathbf{T}_1 is a translation from the centroid of $\{\mathbf{p}_{i,tracked}\}$ to the origin, \mathbf{R} is a rotation about the origin, and \mathbf{T}_2 is a translation from the origin to the centroid of $\{\mathbf{p}_{i,correct}\}$. Calculation of the translations is straightforward, and the system finds \mathbf{R} by

searching the space of euler angles using a downhill simplex method [8] to minimize (1). Tracker readings can then be transformed by using \mathbf{T}_{RB} to transform tracked position and \mathbf{R} to transform tracked orientation.

2.2 Warp Calibration Using Tetrahedral Mesh

The second level of calibration compensates for field warp using a local method to correct for spatial distortions of tracked space. It is not a direct implementation of any previously existing approach, but its tetrahedral mesh construction and interpolation method resemble those described in [9].

The method uses calibration points acquired on a grid that is regular in real space. Hexahedral cells defined by the regular grid correspond to deformed cells in tracked space. A search structure is built that divides the deformed cells into tetrahedra. To calibrate a tracking system reading, a search procedure identifies the tetrahedron containing the tracked position. The barycentric coordinates of the tracked point within the tetrahedron are then found and used as weights for interpolation of error vectors and error quaternions stored with tetrahedron vertices.

The procedure is given a regular grid of $l \times m \times n$ calibration points with tracked coordinates $\{\mathbf{p}_{ijk,tracked}\}$ and corresponding correct coordinates $\{\mathbf{p}_{ijk,correct}\}$. If calibration of tracked orientation is desired, the procedure is also given corresponding tracked orientations $\{\mathbf{q}_{ijk,tracked}\}$ and a correct orientation $\mathbf{q}_{correct}$ that is constant for all points (see notes in Section 2.4). Orientation is represented using quaternions [10]. When the procedure is combined with that described in Section 2.1, the $\mathbf{p}_{ijk,tracked}$ and $\mathbf{q}_{ijk,tracked}$ have been pretransformed by the \mathbf{T}_{RB} and \mathbf{R} described there.

The system computes a position error vector for each calibration point as:

$$\mathbf{e}_{ijk} = \mathbf{p}_{ijk,correct} - \mathbf{p}_{ijk,tracked}. \quad (2)$$

If calibration of orientation is desired, the system also computes error quaternions using quaternion composition:

$$\mathbf{q}_{ijk,error} = \mathbf{q}_{correct} \mathbf{q}_{ijk,tracked}^*, \quad (3)$$

where \mathbf{q}^* is the quaternion conjugate of \mathbf{q} .

These equations follow from the following model of tracked values as a function of correct values and errors:

$$\mathbf{p}_{tracked} = \mathbf{p}_{correct} - \mathbf{e}, \text{ and } \mathbf{q}_{tracked} = \mathbf{q}_{error}^* \mathbf{q}_{correct}. \quad (4)$$

The grid contains $(l-1) \times (m-1) \times (n-1)$ hexahedral cells. These hexahedral cells are further divided into tetrahedra. Figure 1 illustrates two ways of dividing a hexahedral cell into five tetrahedra. In a tetrahedrization of the entire grid,

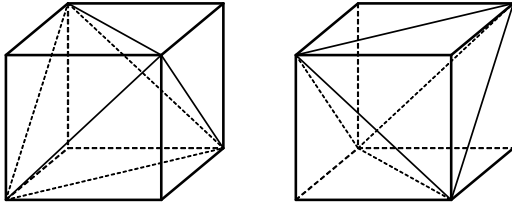


Figure 1: Two cell tetrahedrizations

these two cell tetrahedrizations are alternated for adjacent cells to provide a consistent division of shared hexahedral cell sides. This procedure results in two possible tetrahedrizations of the entire grid, resulting in different choices of calibration points used in computing a correction. The system allows a user to select which tetrahedrization is used, and it also provides the option of computing corrections for both choices and then using an average of the results. The latter approach is used for the evaluation described in Section 3.

To calibrate a reading of tracked position and orientation (\mathbf{p}, \mathbf{q}) , the warped tetrahedral mesh must be searched to identify the tetrahedron surrounding \mathbf{p} . For small grids, a linear search is sufficient, but a spatial partitioning approach is used to efficiently search large grids. The range of tracked space occupied by the mesh is divided into voxels with a list of intersecting tetrahedra stored with each voxel. The coordinates of \mathbf{p} are used to index the voxel containing it, and the voxel's tetrahedron list is then searched for the tetrahedron containing \mathbf{p} . To test whether or not \mathbf{p} is inside a particular tetrahedron, the procedure determines to which side of each of the four bounding planes \mathbf{p} lies. The point \mathbf{p} is inside the tetrahedron if and only if it is on the inner side of all bounding planes. A small tolerance is added to avoid problems from limited precision of calculations.

To interpolate error from the values stored with tetrahedron vertices, the system computes the barycentric coordinates of \mathbf{p} within the surrounding tetrahedron and uses them as weights in a weighted average. This is done using Crout's algorithm [8] to solve for the barycentric coordinates (w_0, w_1, w_2, w_3) in the following system of equations:

$$\begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \quad (5)$$

where (p_x, p_y, p_z) are the coordinates of \mathbf{p} and (x_i, y_i, z_i) are the coordinates of tetrahedron vertex i , $(i = 0, 1, 2, 3)$.

A correction amount $(\mathbf{d}_p, \mathbf{d}_q)$ is then computed using:

$$(\mathbf{d}_p, \mathbf{d}_q) = \left(\sum_{i=0}^3 w_i \mathbf{e}_i, \text{Normalize} \left(\sum_{i=0}^3 w_i \mathbf{q}_{e,i} \right) \right), \quad (6)$$

where \mathbf{e}_i is the position error stored for vertex i and $\mathbf{q}_{e,i}$ is the orientation error stored for vertex i .

In (6), a weighted average of error quaternions is computed linearly and a renormalization step ensures the result is a unit quaternion. If the difference in the error quaternions is large, an averaging based on spherical linear interpolation may produce better results. There are two quaternions that represent an orientation, each one having its terms negated relative to the other. All system procedures that average or interpolate quaternions ensure that any two of the quaternions have a positive dot product. This is not possible for arbitrary sets of quaternions, but it is possible here since the quaternions represent similar orientations.

Finally, a calibrated tracker reading $(\mathbf{p}', \mathbf{q}')$ is computed as:

$$(\mathbf{p}', \mathbf{q}') = (\mathbf{d}_p + \mathbf{p}, \mathbf{d}_q \mathbf{q}). \quad (7)$$

The system also includes a mechanism for transitioning between calibrated and uncalibrated regions of space. The grid is grown outward in all directions with the addition of an outer shell of tetrahedra. The new vertices for the shell are found by projecting outer gridpoints of the original grid outward along edge lines and diagonals of the original grid. Tracked points in the outer shell are handled by the same correction system just described for the inner grid except that correction values for the new vertices are set to represent zero error in position and orientation. Beyond the shell, the calibration is not applied. The result is that the shell produces a transitional region from calibrated regions to uncalibrated regions.

2.3 Warp Calibration Using Spline Model

The third calibration level uses a global spline-based approach that has not been investigated by earlier tracker calibration research. The most well-established global method is the use of a polynomial fit to compute curve coefficients to map from tracked coordinates to calibrated coordinates, as in [3, 5, 7]. High-order polynomials produce undesirable oscillations that can introduce additional error [3], but low-order polynomials clearly cannot capture much detail. The spline model presented here better handles local detail. Although the chosen spline form lacks local control, the influence of local details on distant spline segments is slight. Furthermore, other spline forms can be substituted to trade off a degree of parametric continuity for local control.

In contrast to global mappings directly from tracked coordinates to calibrated coordinates, the spline model

coefficients map from unwarped to warped space, and an inverse mapping is found for each point when calibrated. Interpolation of correction values is then carried out trilinearly in the spline model's parameter space.

The procedure uses the same calibration data as the tetrahedral mesh method: $l \times m \times n$ calibration points with tracked coordinates $\{\mathbf{p}_{ijk,tracked}\}$, correct coordinates $\{\mathbf{p}_{ijk,correct}\}$, tracked orientations $\{\mathbf{q}_{ijk,tracked}\}$, and correct orientation $\mathbf{q}_{correct}$. The position errors $\{\mathbf{e}_{ijk}\}$ and orientation errors $\{\mathbf{q}_{ijk,error}\}$ are also the same.

The spline model has parameters u, v, w and is denoted $\mathbf{s}(u, v, w)$. Its knot intervals are uniform and it interpolates the calibration points, with $\mathbf{s}(i, j, k) = \mathbf{p}_{ijk,tracked}$. A description of the regular calibration grid is used to compute a transform \mathbf{T} from real space to parameter space using basic operations such as translation, rotation, and scale so that a point \mathbf{p}_r in real space can be mapped to the modeled tracked space using $\mathbf{s}(\mathbf{T}\mathbf{p}_r)$, e.g., $\mathbf{s}(\mathbf{T}\mathbf{p}_{ijk,correct}) = \mathbf{p}_{ijk,tracked}$. This relationship is illustrated in Figure 2.

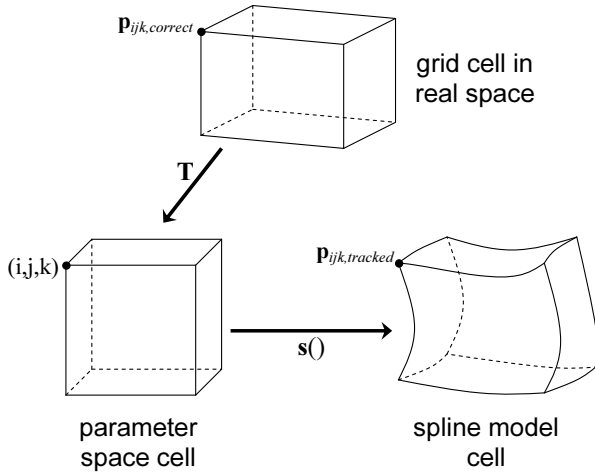


Figure 2: Forward mapping

The value $\mathbf{s}(u, v, w)$ is computed in a standard manner by a sequence of multiple one-dimensional spline operations. One-dimensional splines are constructed along the first dimension of the calibration grid (using tracked coordinates) and are evaluated at parameter value u to generate a set of $m \times n$ interpolated points. A second set of one-dimensional spline constructions and evaluation at v further interpolates these points along the second dimension to compute n points. A final spline construction from these points and an evaluation at w computes the point $\mathbf{s}(u, v, w)$. The spline constructions along the first dimension are stored and reused as needed, but the other spline constructions are not stored. Spline construction uses de Boor's CUBSPL procedure [11]. The system constructs "not-a-knot" splines by default, which are C^2 continuous interpolating splines with end conditions of C^3

continuity at the first and last join points. The system optionally constructs natural cubic splines.

To calibrate a tracker reading (\mathbf{p}, \mathbf{q}) , the system first inverts the above mapping to find $(u_p, v_p, w_p) = \mathbf{s}^{-1}(\mathbf{p})$. This can be done using an iterative technique or an optimization procedure to find parameters u_p, v_p, w_p that minimize:

$$\|\mathbf{p} - \mathbf{s}(u_p, v_p, w_p)\|^2. \quad (8)$$

The downhill simplex method is currently used. This choice of methods is sufficient to demonstrate and evaluate the spline model approach, but a faster technique requiring fewer function evaluations is preferable in practice. However, the number of required steps is small due to the availability of a good initial estimate $\mathbf{T}\mathbf{p}'$, where \mathbf{T} is the transform from real space to parameter space and \mathbf{p}' is the calibrated value computed using the tetrahedral mesh method described in Section 2.2.

The knot interval containing (u_p, v_p, w_p) is a cube in parameter space of the spline model that corresponds to a calibration grid cell. A correction value is therefore computed by trilinear interpolation (in the spline model's parameter space) of the error vectors and error quaternions of calibration points defining the cell. Interpolated quaternions are renormalized, and the final application of correction values to tracker readings is similar to equation (7). For calibration of position only, a more straightforward approach is to apply \mathbf{T}^{-1} to (u_p, v_p, w_p) .

2.4 Notes on Calibration of Orientation

The tetrahedral mesh and spline methods assume orientation error is a function of tracked position and does not depend on tracked orientation. Livingston and State [2] tested this assumption for an Ascension tracker and found it to be false. However, Ikits et. al. [7] showed that this result was obtained with an inappropriate choice of an orientation-dependent reference frame for defining orientation error. Equation (3) defines orientation error with respect to a fixed base frame as done in [7].

Calibration grid measures are assumed to be gathered using a constant sensor orientation $\mathbf{q}_{correct}$, although the equations are easily generalized to relax this constraint. The system includes a method for measuring $\mathbf{q}_{correct}$ in cases where sensor orientation is not known precisely, provided the sensor can be rotated in certain ways. This is useful if the mounting of the sensor to a calibration apparatus makes it difficult to accurately determine sensor orientation or if the sensor coordinate system is not assumed to be accurately aligned with the sensor housing.

Suppose the measurement apparatus allows the sensor to be rotated about two axes \mathbf{a} and \mathbf{b} that are known with respect to coordinate system $\{F\}$. For example, let $\{F\}$ be the user-defined frame in which the $\mathbf{p}_{i,correct}$ of Section 2.1

were expressed, and let tracker readings be pre-transformed to $\{F\}$ with the error-minimizing rigid body transform. With respect to $\{F\}$, let \mathbf{q}_0 be the tracked orientation of the unrotated sensor, \mathbf{q}_1 be the tracked orientation of the sensor when rotated by some amount about \mathbf{a} , and \mathbf{q}_2 be the tracked orientation of the sensor instead rotated by some amount about \mathbf{b} . An intermediate coordinate system $\{G\}$ is constructed with \mathbf{a} defining its X axis, $\mathbf{a} \times \mathbf{b}$ defining its Y axis, and $\mathbf{a} \times (\mathbf{a} \times \mathbf{b})$ defining its Z axis. A rotation matrix \mathbf{R} with these axes descriptions as columns (normalized) maps orientations from $\{F\}$ to $\{G\}$ and is converted to quaternion \mathbf{q} . The tracked versions of \mathbf{a} and \mathbf{b} , named \mathbf{a}' and \mathbf{b}' , are found by extracting the axes from rotations $\mathbf{q}_{a'}$ and $\mathbf{q}_{b'}$, where $\mathbf{q}_{a'} = (\mathbf{q}_1 \mathbf{q}_0^*)$ and $\mathbf{q}_{b'} = (\mathbf{q}_2 \mathbf{q}_0^*)$. A quaternion \mathbf{q}' is then computed from \mathbf{a}' and \mathbf{b}' using a construction similar to that used to construct \mathbf{q} from \mathbf{a} and \mathbf{b} . The difference in orientations \mathbf{q} and \mathbf{q}' is due to tracking error, so the orientation of the unrotated sensor with respect to $\{F\}$ is found as $\mathbf{q}_{correct} = (\mathbf{q} \mathbf{q}'^* \mathbf{q}_0)$.

3. Evaluation

3.1 Overview

The system has been used to calibrate a region of space near the surface of a passive-haptic panel that provides haptic stimulation to users of a visually virtual control panel. This application requires higher tracking accuracy than is needed for conventional virtual environments, since a real object needs to be registered with its virtual counterpart. Further from the panel surface, less accuracy is required and field warp calibration is not performed. The transition mechanism described in Section 2.2 is used to smoothly transition between the regions. Additionally, all tracker readings are pretransformed by the rigid body transform described in Section 2.1 to reduce the difference between tracked and correct coordinates of the grid, further improving the transition.

To evaluate the calibration system, calibration points were gathered on a grid with $7 \times 5 \times 3$ points near the panel surface. 100 readings at random locations within the grid were used to find average position and orientation error after each level of calibration. The experiment was performed once in a low-warp environment and once in heavily distorted environment. The tracker was an Ascension MiniBird 500 with its transmitter placed behind the right side of the panel structure.

3.2 Data Collection

Calibration points were gathered by snapping a LEGO-mounted sensor in place at regular intervals along a large LEGO plate placed on the panel as seen in Figure 3. The panel rests in a grooved base that allows it to slide back

and forth to provide readings in a third dimension. LEGO strips are located along the grooves so the panel can be positioned with anchor blocks. A slight tilt in the panel results in calibration mesh cells being slightly sheared. Each edge of a single calibration cell measures 13 LEGO spaces (about 4.1 inches) in real space, so calibration grid size is about $24.6 \times 16.4 \times 8.2$ inches (neglecting shear).

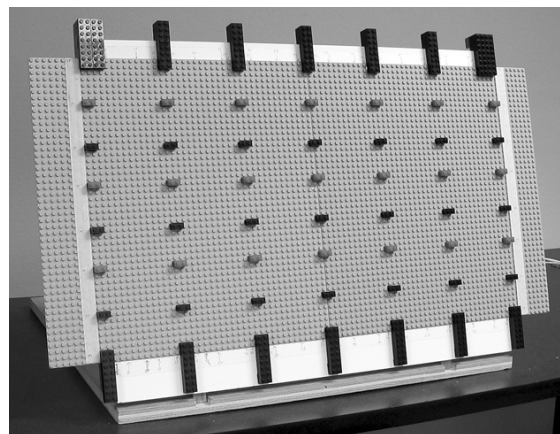


Figure 3: Calibration grid on panel

At each gridpoint, 1000 successive readings were taken and averaged to determine tracked position and orientation. Quaternions were used to represent orientation and were averaged using their barycenter followed by renormalization.

To experimentally measure correct sensor orientation as described in section 2.4, the LEGO structure pictured in Figure 4 was used. Rotations about three axes are shown. Only two rotations are needed by the procedure.

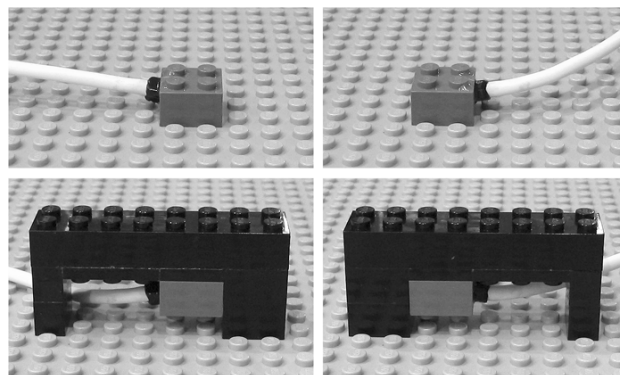


Figure 4: Rotation of a LEGO-mounted sensor

For evaluation of the calibration system, 100 random LEGO coordinates within the grid were generated. Tracker readings were gathered at each of these coordinates using an average of 1000 readings each. The tracker readings were passed through all levels of the calibration system for

analysis of remaining error. Position errors were computed using length of vector difference in positions, and orientation errors were computed using absolute value of the shortest angle of rotation between orientations.

After all other data collection, the calibration grid measurement was repeated so the accuracy of the LEGO apparatus could be estimated.

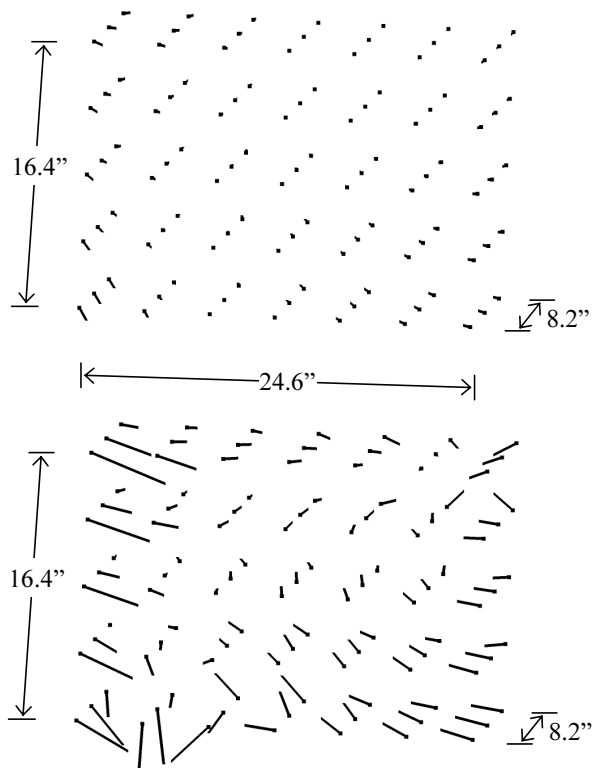


Figure 5: Calibration grid position error for low-warp (top) and high-warp (bottom) environments

3.3 Results

Table 1 summarizes the difference in two separate measurements of the 105 calibration grid points in the low-warp environment. This estimates the error that can be expected to result from limited accuracy of the LEGO apparatus and changes in the environment or in behavior of the tracking system over a short period of time. Approximately 3 hours passed between measurements.

Table 1: Difference in repeated grid measures

	mean	σ	median
position (inches)	0.014	0.006	0.013
angle (degrees)	0.207	0.173	0.153

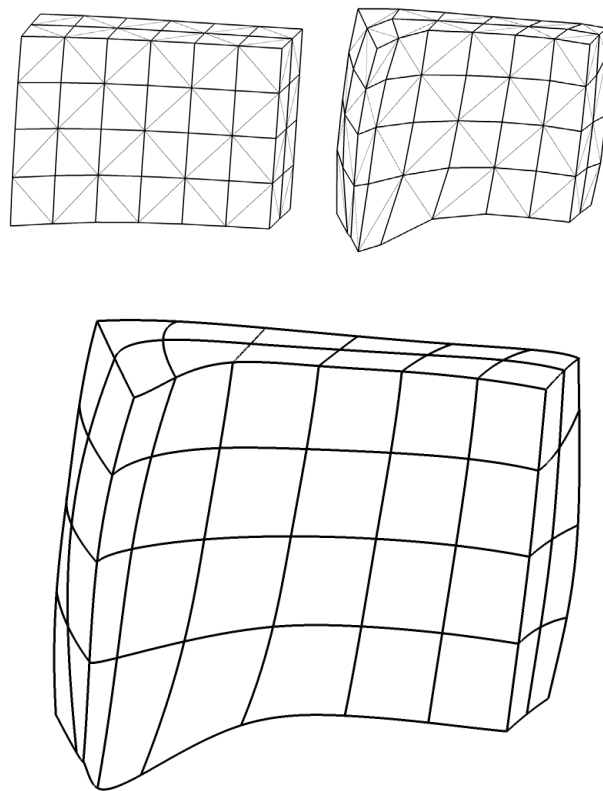


Figure 6: Warp models, showing the tetrahedral meshes for both low-warp and high-warp environments (top) and the spline model for the high-warp environment (bottom)

Figure 5 and Figure 6 illustrate the calibration grid data in the low-warp and high-warp environments. Figure 5 illustrates position error at gridpoints using line segments from correct positions to tracked positions after application of the rigid body transform described in Section 2.1. The transform is applied to show error in a way that does not depend on human judgment to align coordinate systems. Figure 6 shows tetrahedral mesh and spline models used for warp calibration. For clarity, inner structures are not shown.

The low-warp grid shows shape distortions primarily on the left edge of the grid. This is the edge furthest from the electromagnetic transmitter located behind the right side of the panel. Care has been taken to remove metal objects from the environment, but some warp can be expected at the left edge since the MiniBird system uses a short-range transmitter.

Severe distortion is seen in the high-warp grid. This results from a PC monitor located left of the panel, a PC speaker located right of the panel, and a few smaller PC components on the table. This type of environment is

usually avoidable, but the two cases presented here were chosen to illustrate the range of warp that can be expected.

Figure 7 illustrates tracked error for the 100 test points. These plots reflect use of the same rigid body transform used for Figures 5 and 6.

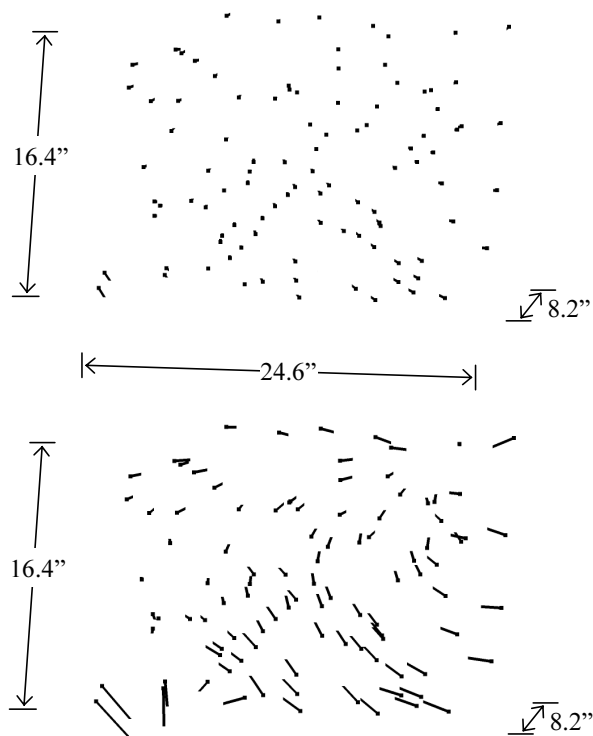


Figure 7: Position error for 100 random points in low-warp (top) and high-warp (bottom) cases

The accuracy of each calibration level for the 100 test points is summarized in Table 2, Table 3, Table 4, and Table 5. The first row of each table reflects use of the rigid body transform only. The other rows show accuracy of the tetrahedral mesh and spline methods. A *trimmed mean* is computed by eliminating the largest 5% and the smallest 5% of error values, resulting in 90 values instead of 100. In the high-warp case, this eliminates a few extreme values corresponding to cells that were warped too severely for a calibration system to reasonably model their warp at the chosen grid resolution. Conclusions drawn here are based on medians and trimmed means.

The tetrahedral and spline methods are both shown to greatly reduce position error compared to the use of only the rigid body transform. The reduction in position error is around 80-90% using the tetrahedral method. Compared to the tetrahedral method, the spline method further reduces position error by about 25% in the low-warp case and about 40-50% in the high-warp case. In the low-warp case,

the remaining position error is not much larger than the error expected from the measurement system as seen in Table 1.

Both tetrahedral and spline methods also greatly reduce orientation error. When the tetrahedral method is compared to the use of only the rigid body transform, it is found to reduce orientation error by about 75-85% in the low-warp case and by as much as 95% in the high-warp case. Use of the spline method further reduces orientation error by a small amount.

Table 2: Position error (inches) after each calibration level for low-warp environment

level	mean	σ	median	trimmed mean
1	0.218	0.100	0.209	0.212
2	0.026	0.015	0.024	0.024
3	0.019	0.009	0.018	0.018

Table 3: Orientation error (degrees) after each calibration level for low-warp environment

level	mean	σ	median	trimmed mean
1	1.968	0.482	2.000	1.952
2	0.515	0.514	0.353	0.446
3	0.445	0.414	0.338	0.390

Table 4: Position error (inches) after each calibration level for high-warp environment

level	mean	σ	median	trimmed mean
1	0.931	0.454	0.804	0.882
2	0.218	0.377	0.079	0.152
3	0.128	0.302	0.045	0.075

Table 5: Orientation error (degrees) after each calibration level for high-warp environment

level	mean	σ	median	trimmed mean
1	10.545	2.060	10.822	10.605
2	0.886	1.558	0.473	0.617
3	0.585	0.568	0.399	0.507

Note that the comments above compare tetrahedral and spline methods to the use of only the rigid body transform. This is to avoid any dependence of results on human judgement for base frame alignment, which would typically result in even larger improvements being found.

Comparisons between existing evaluations of tracker calibration systems are complicated by differences in grid sizes, grid resolutions, hardware, environment, error metrics, and other differences in experimental conditions. Based on the experiment above, the spline method can be expected to outperform local methods that use linear interpolation. The results obtained for the spline model are also better than those reported for polynomial fit techniques by Kindratenko [3], who reported “four times

improved position and three times improved orientation.” Ikits et. al. [7] report an 88.6% improvement in position error mean and an 80.3% improvement in orientation error mean using fourth-order polynomials. However, polynomials can only be expected to perform well when there is minimal local detail to be handled. The spline approach can be expected to handle a wider range of environments. Based on a direct comparison by Kindratenko and Bennett [4], polynomial fit would also not be expected to perform as well as local interpolation methods for orientation calibration. The experiment above shows the spline model to be slightly better than local interpolation for orientation calibration.

4. Conclusion and Future Work

The tracker calibration methods presented here can greatly reduce both position and orientation errors. This was confirmed experimentally for tracker calibration near a passive-haptic panel in both low-warp and high-warp environments. Calibration based on inverse evaluation of a tricubic spline model was shown to be superior to calibration based on interpolation of correction values from vertices of a surrounding tetrahedron. In addition to these calibration methods, the system includes procedures that remove dependence on human judgment for alignment of both transmitter and sensor coordinate frames.

The system can easily be extended to incorporate and test various alternative weighting functions or spline forms to see if any of them produce improved results. For example, further evaluation could determine if a particular spline end condition improves results, or whether or not it is beneficial to sacrifice a degree of parametric continuity to gain local control.

The drawback of the spline method is the computational expense of repeated spline construction and forward evaluation to calibrate a single reading. The downhill simplex procedure should be replaced with a more efficient procedure and the actual speed of the calibration methods should be evaluated. To control execution time for demanding real-time applications, an upper bound can be placed on the number of steps taken by the inverse spline solver. Since a good initial estimate is available from the tetrahedral method, a small number of steps should suffice in a carefully tuned solver. In cases where the spline method remains too costly, the spline model may still be useful for generating high-resolution LUTs or polyhedral meshes as input to faster approaches.

Although the system was evaluated for a short-range tracking configuration, it is also applicable to long-range configurations as long as a suitable apparatus exists for acquiring grid measurements. As the number of cells in the calibration grid increases, the cost of global spline construction increases and the global spline method may become too costly. In this case, a spline form having local

control may be useful since fewer gridpoints would be needed for spline segment construction.

5. Acknowledgements

The author thanks Jack Sullins and Elaine Sullins for assistance with construction of the physical panel, and Arun P. Indugula for assistance with data collection.

6. References

- [1] M. A. Nixon, B. C. McCallum, W. R. Fright, and N. B. Price, "The Effects of Metals and Interfering Fields on Electromagnetic Trackers," *Presence: Teleoperators and Virtual Environments*, vol. 7, pp. 204-218, 1998.
- [2] M. A. Livingston and A. State, "Magnetic Tracking Calibration for Improved Augmented Reality System," *Presence: Teleoperators and Virtual Environments*, vol. 6, pp. 533-546, 1997.
- [3] V. Kindratenko, "Calibration of Electromagnetic Tracking Devices," *Virtual Reality: Research, Development, and Applications*, vol. 4, pp. 139-150, 1999.
- [4] V. Kindratenko and A. Bennett, "Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems," presented at Virtual Environments 2000 Eurographics Workshop, Berlin, 2000.
- [5] S. Bryson, "Measurement and Calibration of Static Distortion of Position Data from 3D Trackers," presented at SPIE Conference on Stereoscopic Displays and Applications III, San Jose, 1992.
- [6] W. Briggs, "Magnetic Calibration by Tetrahedral Interpolation," presented at NIST-ASME Industrial Virtual Reality Symposium, Chicago, 1999.
- [7] M. Ikits, J. D. Brederson, C. D. Hansen, and J. M. Hollerbach, "An Improved Calibration Framework for Electromagnetic Tracking Devices," presented at Virtual Reality, Yokohama, Japan, 2001.
- [8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, UK: Cambridge University Press, 1988.
- [9] D. N. Kenwright and D. A. Lane, "Interactive Time-Dependent Particle Tracing Using Tetrahedral Decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, pp. 120-129, 1996.
- [10] K. Shoemake, "Quaternions," University of Pennsylvania, Department of Computer and Information Sciences, Philadelphia May 1994.
- [11] C. deBoor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.