Correction of Location and Orientation Errors in Electromagnetic Motion Tracking

John G. Hagedorn
Steven G. Satterfield
John T. Kelso
Whitney Austin
Judith E. Terrill
Adele Peskin

National Institute of Standards and Technology
100 Bureau Drive, Stop 8951
Gaithersburg, MD 20899-8951, USA

Abstract

We describe a method for calibrating an electromagnetic motion tracking device. Algorithms for correcting both location and orientation data are presented. In particular we use a method for interpolating rotation corrections that has not previously been used in this context. This method, unlike previous methods, is rooted in the geometry of the space of rotations. This interpolation method is used in conjunction with Delaunay tetrahedralization to enable correction based on scattered data samples. We present measurements that support the assumption that neither location nor orientation errors are dependent on sensor orientation. We give results showing large improvements in both location and orientation errors. The methods are shown to impose a minimal computational burden.

1 Background

The use of motion tracking devices is essential in immersive visualization systems. The location and orientation of the user's eyes must be continuously tracked so that the system can render images of a spatially stable virtual world, in stereo, in real-time. In addition, motion tracking is also commonly used to track hand-held devices that operate as tools in the virtual world created within the immersive display.

At the National Institute of Standards and Technology (NIST), we have an immersive visualization system that is pictured in Figure 1. This figure indicates several important components of the system: the three screens that provide the visual display and the motion tracker transmitter and sensors. The three screens are three large video displays that are placed edge-to-edge in a corner configuration. These three screens are used to display a single three-dimensional scene as shown in the figure.

We use a Flock of Birds magnetic motion tracking system manufactured by Ascension Technology Corporation. There are two components of that system that are important for our discussion: the transmitter unit and the sensor units as shown in Figure 1. The transmitter transmits a pulsed DC magnetic field; each sensor measures that field (Ascension, 2002). From these measurements the location and orientation of each sensor are calculated. The transmitter unit is a box that is approximately 30 cm on a side and the sensor is approximately 2.5 cm in its largest dimension. We run the system with two sensors in simultaneous operation. One sensor is attached to the user's stereo glasses and one sensor is attached to a hand-held pointing device.

The 3D scene displayed across the three screens is rendered based on the location and orientation of the user's eyes as determined by the tracking system's determination of the location and orientation of the sensor attached to the user's stereo glasses. The user can move around the virtual objects, viewing them from different directions and interactions with the objects can be accomplished through the use of the motion-tracked hand-held device.

The underlying software on which our immersive system is built is DIVERSE (Kelso et al., 2003). It provides a portable, modular, open source software platform that manages all aspects of the virtual environment. This includes handling the interfaces to devices (such as motion trackers and user interface devices), stereo parallax, asynchronous viewing frusta, and so on. DIVERSE provides easy methods for adding new functionality, such as new user interaction techniques, which has been done by NIST and others.

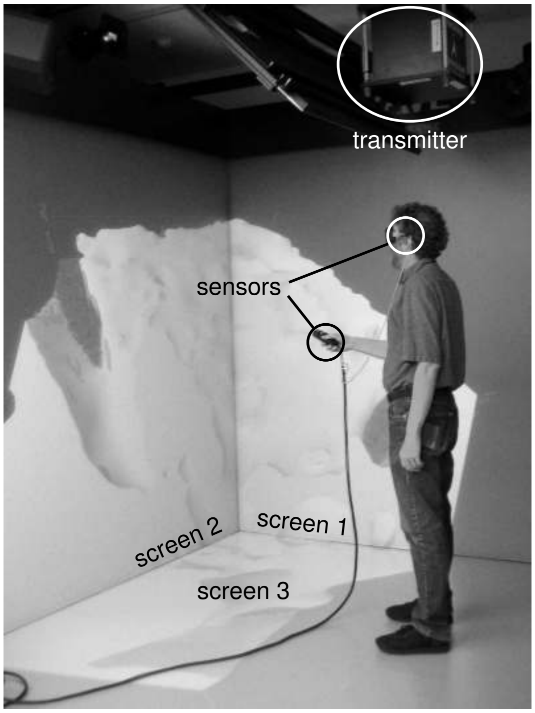

Figure 1: The NIST immersive visualization system. A single 3D scene is being displayed across the three screens. Note the tracker transmitter suspended from the ceiling. There are two tracker sensors; one is attached to the user's stereo glasses and the other is attached to a hand-held pointing device.

Electromagnetic motion trackers like the one we use at NIST are very commonly used in immersive visualization systems. In our installation, it became obvious that there were substantial errors in the data reported by the motion tracking system. These inaccuracies resulted in various effects, such as:

- virtual objects that should appear stationary appear to move as the user moves,
- straight lines appear bent when they cross the boundaries between screens, and

- virtual objects tied to the tracked hand-held device appear to be incorrectly located or oriented.

Figure 2 displays an example of how straight lines may appear bent. In this figure, the grid lines should all be straight. To the user in the immersive environment, the lines appear bent at the points where they cross the boundaries between screens. This is due to the fact that the images are being drawn based on an incorrect location for the eyes of the observer. This figure is based on actual errors observed in the motion tracking system at NIST. It is by no means the worst case that could have been provided. In informal observations that we made before initiating this project, we observed location errors in excess of 50 cm and orientation errors that appeared to be more than 15 degrees.

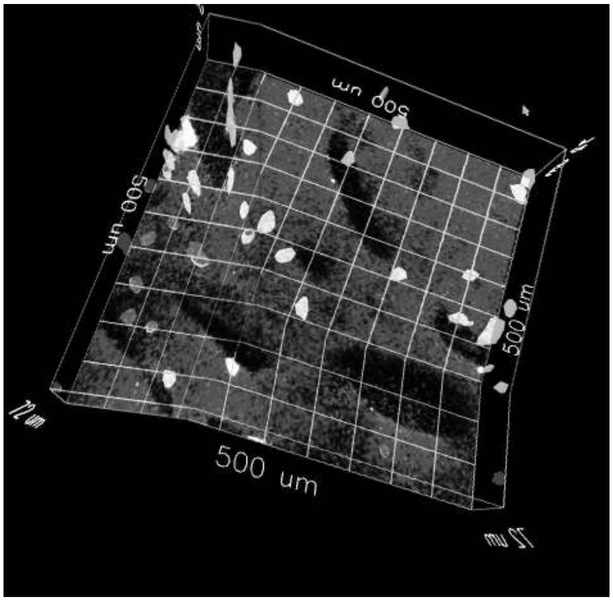

Figure 2: Distortions due to tracker miscalibration. The grid lines should be straight. They bend at the points where they cross the boundaries between screens in the immersive environment. (Note that the image displayed here depicts 3D data from a confocal microscope and the annotation within the image refers to that data set.)

Electromagnetic trackers operate by measuring electromagnetic fields generated by the device itself (Raab et al., 1979). These fields can be distorted by various environmental factors such as metal objects and electromagnetic fields generated by video monitors (Nixon et al., 1998). These environmental factors contribute substantially to errors in measurements reported by the tracking device. It is also important to note that these environmental factors differ for each installation, so any correction of these errors must be based on the particulars of the installation.

2 A Note on Terminology

Throughout this paper we discuss several methods for representing orientation. In all cases, orientation is described as a rotation that transforms an object from a nominal orientation to the desired orientation. Thus, there is an equivalence between rotational transformations and orientations. For this reason, we often use the term ”rotation” as a synonym for ”orientation”, particularly when we refer to an orientation in terms of its representation as a rotational transformation.

We use the term ”location” throughout this paper to refer to the point at which an object (such as a tracker sensor) is located. Many other authors use the term ”position” rather than ”location” with this meaning. However ”position” is sometimes used with other meanings. In particular, an important survey article on tracker calibration (Kindratenko, 2000) uses ”position” to encompass both location and orientation. To minimize confusion, we generally avoid the use of the term ”position” unless the context makes the meaning clear.

3 Previous Work

There has been substantial work in measuring and correcting motion tracking errors. An excellent survey article (Kindratenko, 2000) was published in 2000. Since then several additional efforts have been described, such as (Saleh et al., 2000; Ikits et al., 2001; Jayaram and Repp, 2002; Borst, 2004).

Essentially all of the published efforts follow the same basic approach: collect tracker data at known locations and orientations; then use these data (in some form) to perform real-time correction of reported data.

It should be noted that location and orientation corrections are typically derived only from the measured location without use of the measured orientation. So there is an implicit assumption that errors are dependent on sensor location but not on sensor orientation. Only (Livingston and State, 1997) has observed that this assumption does not hold; at least two other projects (Kindratenko and Bennett, 2000; Zachmann, 2000) have observed no substantial dependency of tracker error on sensor orientation. We were interested in investigating the validity of this assumption because it has substantial implications for data collection procedures as well as for the construction of the correction algorithms.

3.1 Data Collection

The collection of data has been done in a variety of ways, constrained by the need to use equipment that will not interfere with the tracker measurements. This usually involves the use of a measurement apparatus made of wood and/or plastic (for example, (Bryson, 1992; Zachmann, 1997; Saleh et al., 2000; Jayaram and Repp, 2002; Borst, 2004)). In at least one case an aluminum apparatus was used and tests were performed to ensure that this apparatus did not interfere with the measurements (Livingston and State, 1997). In most cases measurements were made on a regular three-dimensional grid in physical space (for example, (Bryson, 1992; Zachmann, 1997; Saleh et al., 2000; Ikits et al., 2001; Jayaram and Repp, 2002; Borst, 2004)) or in tracked space (for example, (Ghazisaedy et al., 1995)). In other cases measurements were made at points that were not necessarily on a grid (for example, (Livingston and State, 1997)).

3.2 Location Correction

The correction of location errors involves the construction of a function that interpolates (or approximates) the measured data points. The variety of interpolation techniques are well described in (Kindratenko, 2000). The large majority of approaches use conventional 3D interpolation techniques, with various degrees of continuity. Often the original data points are pre-processed to interpolate to a 3D grid of points that is regular in the uncorrected coordinate space. This simplifies the real-time calculations; tri-linear interpolation between these derived grid points is often used.

Several projects describe the use of tetrahedral interpolation techniques (Ellis et al., 1999; Jayaram and Repp, 2002; Borst, 2004) that are, in some ways, similar to the method that we present below. In the discussion of our interpolation techniques we will describe the ways in which our methods differ from these prior approaches.

Other correction schemes have been used, such as neural nets (Saleh et al., 2000) and interactive visual correction (Czernuszenko et al., 1998). Motion tracking based on image feature recognition used in conjunction with electromagnetic tracking has also been been implemented (State et al., 1996).

3.3 Orientation Correction

As with location correction, the correction of orientation errors involves the construction of an interpolating or approximating function. Typically each project used an orientation interpolation scheme that parallels the location interpolation method. Interpolation of orientation is complicated by the geometry of the rotational space, which is decidedly different than Euclidean space.

A key feature of each interpolation method is the representation of orientation. There are many ways of numerically representing orientation. Each of these three forms have been used as the basis of interpolation schemes:

- Euler angle triples
- 3x3 special orthogonal matrices
- Unit quaternions

There are six methods of orientation interpolation that are discussed in the literature:

- Interpolate each Euler angle independently (Kindratenko, 1999). (We will refer to this method as the Euler-Component method.)
- Interpolate each component of the 3x3 matrix; then normalize and orthogonalize (Zachmann, 2000). (Matrix-Component)
- Take the weighted sum of each component of the 3x3 matrix; then use a singular value decomposition (SVD) to derive an average (Livingston and State, 1997; Curtis et al., 1993). (Matrix-SVD)
- Interpolate each component of the quaternion; then normalize (Kindratenko and Bennett, 2000). (Quaternion-4)

- Interpolate the three imaginary components of the quaternion; then derive the fourth component to form a unit quaternion (Ikits et al., 2001). (Quaternion-3)

- Interpolate between two quaternions by spherical linear interpolation, commonly called SLERP (Livingston and State, 1997; Shoemake, 1985).

Each of these methods is problematic. With the exceptions of Matrix-SVD and SLERP, problems arise because they are constructed as formal manipulations of the representations, but lack appropriate geometric meaning.

The Matrix-SVD method does have a clear geometric meaning, but the geometric meaning gives unexpected (and undesired) results. This technique interpolates by minimizing the integral, over the unit sphere, of the weighted sum of the squares of point displacement in real Euclidean space (Curtis et al., 1993). In spite of the evident geometric significance of this formulation, it is very unclear that this minimization problem really represents the rotational interpolation problem that we want to solve.

Matrix-SVD is formulated in terms of distances in Euclidean 3-space rather than a measure of distances in SO(3), the space of rotations. In effect, the Matrix-SVD method defines the magnitude of a rotation by using Euclidean distance. But we know how to quantify the magnitude of a rotation; it is the magnitude of the angle of rotation. This contradiction between the the distance metric used by the SVD method and the inherent geometry of rotations causes the method to produce results that are are substantially at odds with reasonable expectation, as we will show below.

Finally, SLERP (spherical linear interpolation) is a well known technique for interpolating between two orientations and it is well described by Shoemake (Shoemake, 1985). This technique uses quaternions to traverse geodesics in the space of rotations. The problem is that it does not accommodate the interpolation of a group of rotations, which is clearly required in the current context.

The only prior project that uses SLERP (Livingston and State, 1997) uses it only after first preprocessing the measured data to produce a grid of points that are regularly spaced in measured coordinates. The interpolation to this regular grid of points is accomplished by the Matrix-SVD method.

As a way of seeing the difficulties in these methods, consider a comparison of the results of these six interpolation techniques on a very simple linear interpolation problem:

> Find the rotation that is one quarter of the way between
>
> *rotation of 10 degrees about axis (0.371, 0.557, 0.743)*
>
> and
>
> *rotation of 110 degrees about axis (0.371, 0.557, 0.743).*

Clearly, the expected result is:

> *rotation of 35 degrees about axis (0.371, 0.557, 0.743).*

We expect this result because the difference between these two rotations is simply a 100 degree rotation about (0.371, 0.557, 0.743). One quarter of this rotation is a 25 degree rotation about the same axis. We compose this quarter-way rotation with the starting orientation and we get a 35 degree rotation about (0.371, 0.557, 0.743). The results of the interpolation for each of the six methods is presented in Table 1. We note that rather than using rotations of 10 and 110 degrees above, we could just as easily have used 0 and 100 degrees. The results would have been equivalent.

Interpolation	Interpolated Rotation		Error from
Method	angle	axis	Expected Result
Euler-Component	34.15°	(0.13, 0.60, 0.79)	8.51°
Matrix-Component	33.21°	(0.57, 0.37, 0.73)	9.26°
Matrix-SVD	29.21°	(0.37, 0.56, 0.74)	5.79°
Quaternion-4	33.75°	(0.37, 0.56, 0.74)	1.25°
Quaternion-3	31.35°	(0.37, 0.56, 0.74)	3.65°
SLERP	35.00°	(0.37, 0.56, 0.74)	0.00°

Table 1: Results of rotation interpolation methods on sample problem.

Of course, when the angular difference between the two rotations is less than 100 degrees, the errors will be smaller as well. While in practice the differences among these techniques may sometimes be small, there seems no need to use techniques that give incorrect results. The basic problem with each of these methods other than SLERP is that they do not honor the underlying geometry of the space of rotations. SLERP does honor the geometry of rotations, but it is designed for interpolating only between pairs of rotations. The techniques we describe below remedy this deficiency.

4 Our Approach

In many respects our approach is similar to previous work. We collected tracker data at known locations and orientation; we then used these data to construct functions that correct reported data. Our approach is distinguished from prior work primarily in our handling of the interpolation of orientation. Our orientation interpolation method works entirely within the space of rotations and does not assume that the space of rotations is Euclidean.

We use a quaternion representation of orientations, and we interpolate orientation corrections directly from measured orientation errors. First we perform a Delaunay tetrahedralization of previously measured data points; then we calculate barycentric coordinates for the each new measurement relative to the tetrahedron that contains that point. The tetrahedralization and determination of barycentric coordinates is based only on locations. The barycentric coordinates (which sum to 1 by construction) are used as weights for performing weighted averaging of the data at the vertices of the containing tetrahedron.

For orientation averaging, we use these weights with a spherical weighted averaging technique developed by Buss and Fillmore (Buss and Fillmore, 2001) to average the correction rotations at each of the four vertices of the tetrahedron. This use of barycentric coordinates with spherical weighted averaging has a much clearer geometric rationale than previous methods. All of the interpolation steps are done in real time.

4.1 Data Collection

There were three phases to our data collection effort:

Phase 1 Orientation data collected to study dependency of location and orientation error on actual orientation

Phase 2 Location and orientation data collected to construct the correction function

Phase 3 Location and orientation data collected for validation of the correction method

The apparatus we used for all of the data collection phases was required to be non-metallic so that the apparatus itself would not distort the reported data. We accomplished this by using plastic components. We used crates obtained from a local office supply store; they are rectangular, consistent in size, light, and interlocking. The size of each crate is 35.1 cm x 42.7 cm x 26.4 cm. The crates provided a stable platform to which we could attach a tracker sensor at a well controlled location and orientation. Figure 3 shows the apparatus for all phases of data collection.

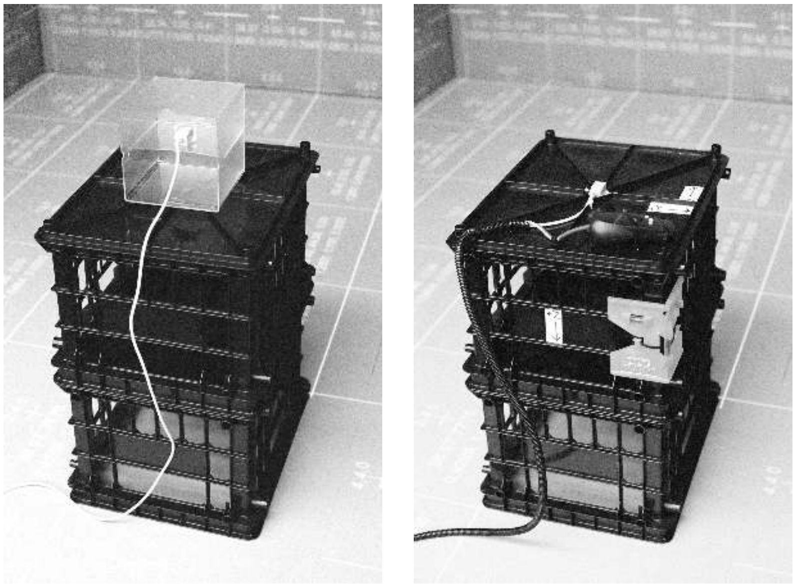

Figure 3: Apparatus used for the data collection. On the left is the equipment used for Phase 1 data collection and on the right is the equipment used for Phases 2 and 3.

The Phase 1 data collection was intended to determine whether orientation errors varied with actual orientation as well as with location. To do this we mounted a sensor in the center of a plastic cube that was approximately 15 cm on a side. The cube was then easily positioned in each of 24 different orientations simply by placing a face of the cube on the top of one of the crates in a rigidly controlled orientation and location. Each of the six faces can be oriented in four ways when it is placed on the crate, yielding 24 different axis-aligned orientations for the cube. Because the sensor is at (or very near) the center of the cube, the location of the sensor is essentially the same for each measurement. We made these 24 measurements of location and orientation at each of four widely spaced locations within the tracked volume.

In Phase 2, we fixed a sensor to the top of one of the crates. We then recorded the location and orientation reported by the tracker at points on a regular 7x6x6 grid over the central viewing area of our immersive system. We ensured that the orientation of the sensor was essentially identical for all measurements. The grid spacing in each dimension was equal to the corresponding dimension of the crates, specified above. This yielded a grid that spanned a region that was 210.6 cm x 213.5 cm x 132.0 cm.

The third and final phase of data collection was done using the same apparatus as the second phase. We used the same spacing between grid points, but the points were offset in each dimension from the Phase 2 grid. This provided us with a 5x4x5 set of measurements with which we could validate our correction methods.

For all data collection phases, we positioned the plastic crates horizontally by projecting a grid onto the floor of the immersive system. Vertical positioning was achieved by stacking the interlocking crates.

Our underlying assumption, based on our understanding of our data collection equipment and methods, is that the nominal sensor location and orientation can be taken as the true placement with appropriate estimations of uncertainty. There are several factors that may contribute to sensor positioning errors when using this apparatus, such as non-linearities in the projector that displays the horizontal positioning grid, human error in manually positioning the crates, and inconsistencies in the plastic crate manufacture. We estimate that ”almost all” (upwards of 98%) of all actual sensor locations will fall within 2 cm of the nominal location. Assuming a normal distribution, this yields a standard uncertainty of ±0.67 cm for sensor location (Taylor and Kuyatt, 1994). By similar reasoning, we estimate a standard uncertainty of ±2 degrees for sensor orientation. While this apparatus does not give us extreme precision in positioning the sensor, these positioning errors are well below the tracker errors that we were measuring.

4.2 Implementation of Correction Methods

As a preprocessing step, we construct a Delaunay tetrahedralization (in measured space) of the points collected in Phase 2 described above. Note that no interpolation of data is done in this preprocessing step; we simply determine the topology of the tetrahedralization.

So each of the measured points is a vertex of the tetrahedralization. The location of the vertex is the X, Y, and Z coordinates reported by the tracker. With each vertex we also store its X, Y, and Z coordinates in real space (i.e., the "true" location) as well as a rotation correction represented as a quaternion.

At run time, the interpolation subsystem is initialized by reading in the data points and tetrahedralization. Then, in real time, for each tracker measurement, we perform these steps:

1. Determine in which tetrahedron the measured location lies (Kenwright and Lane, 1996).

2. Report the barycentric coordinates of the measured location relative to that tetrahedron.

3. Calculate the corrected location and the orientation correction for the measured point by a weighted average of the data at the tetrahedron's vertices using the barycentric coordinates as the weights.

The corrected location is thus calculated directly and the corrected orientation is generated by applying the interpolated orientation correction to the measured orientation. Observe that the corrections (both location and orientation) are generated as a function of the current tracker location, but no use is made of the current measured tracker orientation. This approach is based on the understanding that location and orientation errors are essentially independent of sensor orientation; this is addressed in more detail in the discussion of the results from Phase 1.

In step 3, the weighted average of the X, Y, and Z coordinates are done in the conventional way. We do the weighted average of the rotation corrections by the spherical weighted average methods described in (Buss and Fillmore, 2001). It should be noted that the spherical weighted average method of Buss and Fillmore addresses the rotation averaging problem by formulating an optimization problem that is analogous to that solved by the Matrix-SVD method. But Buss and Fillmore's method uses a distance metric that is based on distances in the space of rotations, rather than in Euclidean space. When calculating weighted averages of only two rotations, this method yields results that are identical to SLERP.

The mapping of rotation correction is continuous, but it is not smooth over the boundaries of tetrahedra. Buss and Fillmore present an algorithm based on splines for ensuring a smooth mapping but, in the interest of real-time performance, we use only the simplest form of the method.

This method is distinguished from prior tetrahedral interpolation approaches (Ellis et al., 1999; Jayaram and Repp, 2002; Borst, 2004) in several ways. Firstly, all of these prior projects derive their tetrahedral decomposition from a 3D grid of points; we use a Delaunay tetrahedralization. The use of a Delaunay tetrahedralization provides substantial advantages. Data need not be collected on a grid; scattered points can be used. This enables the collection of data points more densely in regions of greater distortion. Secondly, none of the prior projects use spherical weighted averaging for orientation correction interpolation. In addition, Jayaram and Repp interpolate individual Euler angles and use a very different method for searching the containing tetrahedron. The method of Ellis et al. seems similar to ours in that it is derived in part from the work of Kenwright and Lane (Kenwright and Lane, 1996), as is our own; their method of interpolating orientations, however, is not described explicitly. The work of Borst is the most similar to our method; this work uses barycentric coordinates as interpolation weights as described by Kenwright and Lane (Kenwright and Lane, 1996), but differs from our work in the ways described above. Our use of Delaunay tetrahedralization in conjunction with the spherical weighted average method for rotation interpolation at scattered points substantially distinguishes our techniques from prior tracker correction efforts.

5 Results

5.1 Dependency of Measurement Errors on Sensor Orientation

As mentioned above, in the Phase 1 data collection, we recorded orientation and location measurements with the sensor at 24 different axis-aligned orientations at each of four locations. We were interested in discovering whether measured location and orientation errors depended on the orientation of the sensor. To do this, we examined the variation of both location and orientation errors over all orientations at each measurement location. If errors are independent of orientation, then we would expect no variation of errors, or at least the magnitude of the variation should be consistent with the inaccuracies of our measurement apparatus and methods. If errors are dependent on orientation, then the variation of those errors will give us an understanding of the magnitude of that dependence.

We first calculated the average error among all of the 24 measurements at a given location; we then looked at the magnitude of the difference between each measurement and the corresponding average. For location errors, we calculated the averages in the normal way. For orientation errors, we formed the error rotations; we then averaged them using the spherical averaging technique described above.

Figure 4 shows a histogram of the magnitude of the deviation of each measured location error from the corresponding location error average. The plot aggregates all of the data derived from Phase 1. The average error deviation is 1.55 cm and the standard deviation is 0.69 cm. We believe that the main source of location deviation is due to the fact that the center of measurement of the physical sensor was not mounted in the exact center of the measurement apparatus, so there was a slight offset for all of the measurements. Additional error was likely introduced by the manual placement of the sensor.

Figure 5 shows a histogram of the angular magnitude of the rotational deviation of each orientation error from the average rotational error. The average deviation is 1.03 degrees with a standard deviation of .38 degrees. Again, there is inevitably measurement error introduced by inaccuracies of our apparatus and manual methods.

In both of these data sets (location and orientation) the observations with the greatest deviation from the mean occurred when the sensor was located at the locations farthest from the tracker transmitter. From our informal observations of error before this study, this is where we expected tracker noise and non-linearity to be at its worst.

Based on our understanding of the inaccuracies of our measurement apparatus and manual measurement methods, we feel that these variations of error are consistent with the assumption that there is no dependence of location or orientation errors on sensor orientation. Furthermore, we believe that if there is, in fact, dependence of error on orientation, the magnitude of the

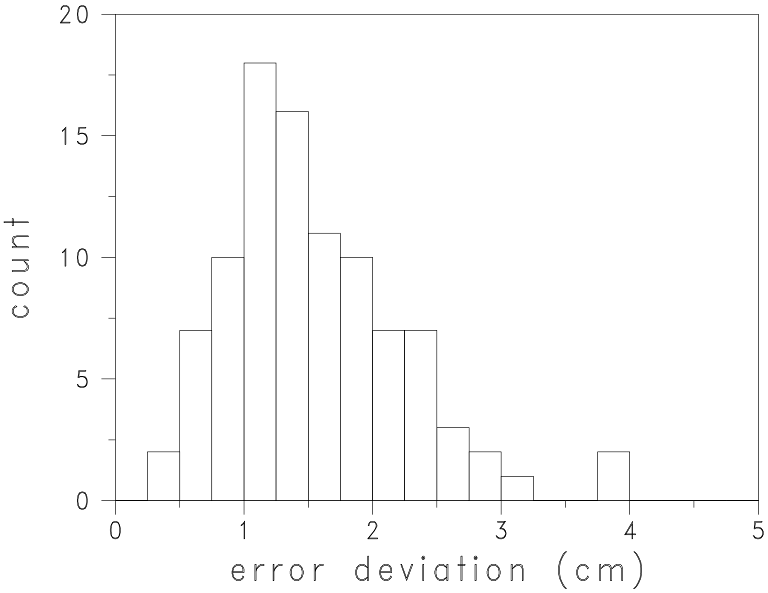

Figure 4: Distribution of the magnitudes of the deviation of location error from location error average for Phase 1 data.

contribution of that dependence to the overall error is sufficiently small that it can safely be ignored. We based this decision first on our preliminary observations and measurements of overall location and orientation errors that were performed before the results reported here. The decision was borne out by the overall errors measured in Phase 2 as described below. We do not contend that these data prove that there is no dependence of error on orientation, but the data do indicate that it is sufficiently small that it is a reasonable working assumption that there is no dependence.

This assumption greatly simplified the Phase 2 effort. The data collection procedure for Phase 2 was not required to sample a range of orientations. As described above, we collected data at many locations, but only for one orientation at each location. Furthermore, the corrections of both location and orientation that are derived from Phase 2 data need only be based on measured location.

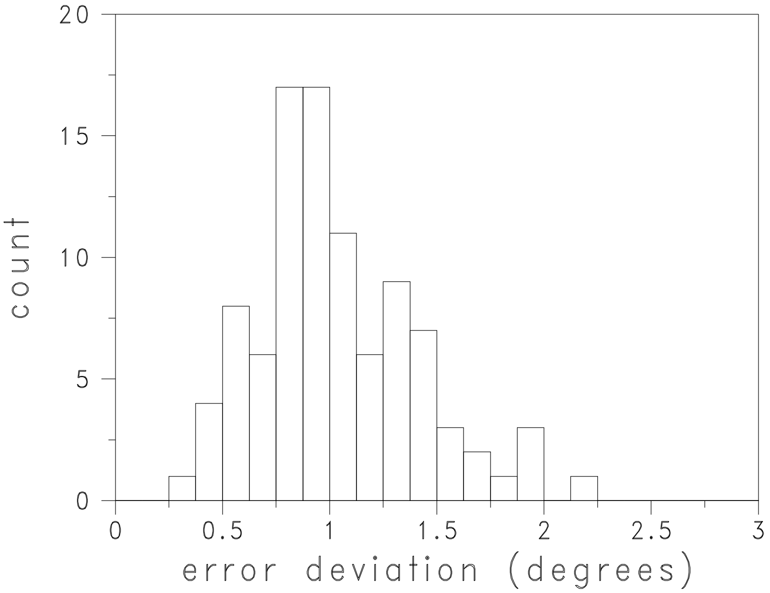

Figure 5: Distribution of the angular magnitudes of the rotational deviation of orientation error from orientation error average for Phase 1 data.

5.2 Observed Errors

Figure 6 shows the distorted and undistorted grid of points collected in Phase 2. The image on the left shows the actual locations of the collected points, the image on the right shows the location of the points as reported by the tracker.

Table 2 shows statistics derived from location errors and angular errors for Phase 2 and Phase 3 data. Note that even though the Phase 3 data points were taken over a slightly smaller and more centrally located volume, they still display very large errors.

Figure 7 shows a plot of measurement errors versus the the distance of the measurement from the transmitter. Clearly the error increases as the distance between transmitter and sensor increases. The error varies (very) roughly with the square of the transmitter/sensor distance. The angular magnitude of the orientation error varies similarly with distance between sensor and transmitter. Previous work (Nixon et al., 1998) suggests that

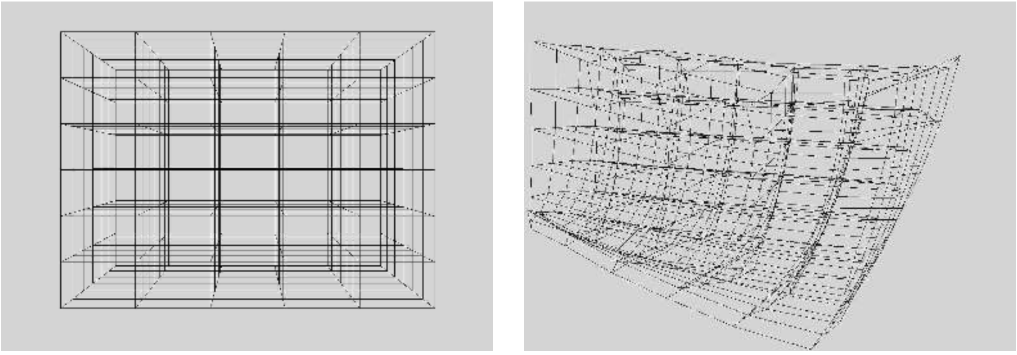

Figure 6: On the left we show the true locations of the points at which we collected data in Phase 2. On the right we show the locations that the tracker reported for these points.

Data		Location Errors		Orientation Errors	
Source	Num Pts	Average	Std Dev	Average	Std Dev
Phase 2	296	37.27 cm	24.52 cm	20.15°	11.36°
Phase 3	100	37.51 cm	19.30 cm	19.57°	8.96°

Table 2: Error statistics for Phase 1 and Phase 2 data.

errors should vary with the fourth power of the transmitter/sensor distance, but this is only when interfering fields are only in close proximity to either the transmitter or sensor. This is not the case in our installation.

5.3 Residual Errors After Correction

We apply our correction methods (based on Phase 2 data) to derive corrected locations for the Phase 3 points. Table 3 shows the results. We see approximately a 95.5% improvement in location errors and approximately a 87.0% improvement in orientation errors. Given the relatively inexact manual data collection method, we consider this to be a very good correction result.

Comparing these results to the results of prior projects is often difficult, not the least because of different methods of reporting and characterizing results. Additional obstacles to comparison are the different configurations of tracking environments, such as size of the tracked region, the amount of

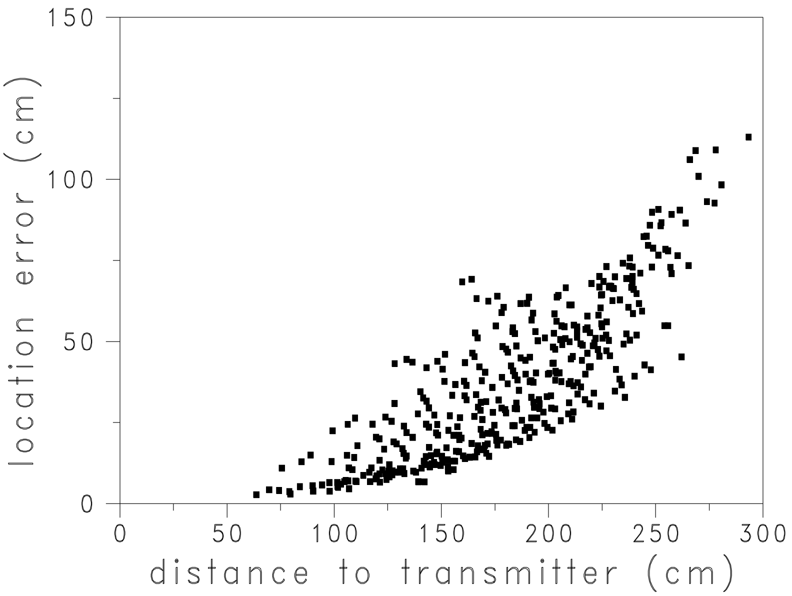

Figure 7: Location error magnitude versus the distance between sensor and transmitter.

distortion, and the different characteristics of tracking apparatus. However, there are results from several projects that can be compared to our own.

In (Livingston and State, 1997) improvements of 78.7% in location and 40% in orientation errors are indicated. The work of (Ellis et al., 1999) shows an "approximately sevenfold" (85.7%) reduction of location error. In (Kindratenko, 1999) an improvement of location error of "as much as 4 times" (75%) and an improvement of orientation error of up to "3 times", (66.7%) are reported, and (Kindratenko and Bennett, 2000) shows an improvement of orientation error by 85.5%. Improvements of 86.6% in location errors and 80.3% in orientation errors are presented in (Ikits et al., 2001). The results in (Borst, 2004) are somewhat difficult to compare to ours because the uncorrected errors are not reported, but the data presented suggests that location errors are reduced by at least 87% and rotation errors are reduced by at least 95% for measurements made in a "high-warp" environment. When comparing these results to our results, note that in Borst's work the tracked region is a volume 41.7 cm x 62.5 cm x 20.8 cm (less than 1% of the volume of our calibrated region) with the tracker transmitter in close proximity to the tracked region. In short, our results compare very well with results from prior work.

Data	Num	Residual Location Errors		Residual Orientation Errors	
Source	Pts	Average	Std Dev	Average	Std Dev
Corrected Phase 3	100	1.69 cm	1.30 cm	2.55°	0.63°

Table 3: Residual location and orientation errors after applying corrections.

Figure 8 shows these residual errors again plotted against the distance between the sensor and the transmitter. We see again that the largest errors come from points that have the greatest separation between transmitter and sensor, but beyond that, no particular structure is evident in the residuals. The distribution of orientation residual errors is similar.

It is interesting to note that Phase 3 data collection was done a full year after the Phase 2 data collection. This indicates both that the errors are very stable over time and that our data collection methodology is sound. Our data collection procedures produced consistent results even though data collection was done by multiple people separated by long spans of time.

5.4 Performance

In actual operation in the immersive system, we notice no performance degradation associated with the correction software. In order to understand performance issues more clearly, we collected about two minutes of continuous raw data for the two sensors that we simultaneously track. This constitutes 12000 individual data points for each of the two sensors. We then isolated the correction algorithm from the other software and performed tests to see how much time was being consumed by the correction algorithm for these two data streams.

We found that the correction algorithm consumed 3.60 seconds of CPU time to process this 120 second stream of tracker data. So the correction accounts for approximately 3 percent of the cycles on a single CPU. Considering that we always run on a multiprocessor machine, this is a computational burden that is very easily borne.

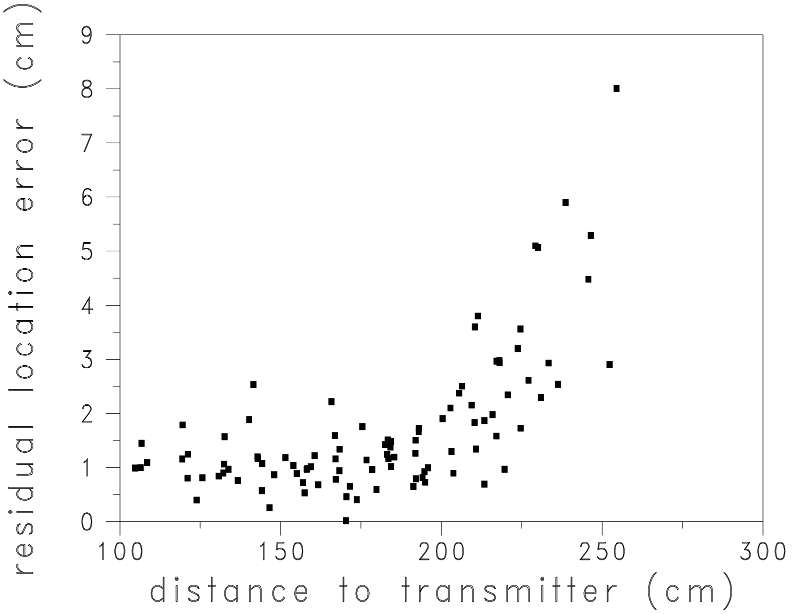

Figure 8: Residual error magnitudes after correction versus distance between sensor and transmitter. This plot shows Phase 3 data points with correction based on Phase 2 data points.

It is also worth noting that this test was made on code that we had made no attempt to optimize, either on the source code level or by the compiler.

5.5 Qualitative Results

When we apply this correction to the tracker in the immersive visualization environment the difference is clear to the user. Objects that should be stable appear to be stationary as the user moves through the virtual scene, the displayed pointer tracks the tracked hand-held device accurately, and there are no visual artifacts when objects cross the divisions between screens. Figure 9 shows an uncorrected and a corrected view that indicate the benefit derived from our tracker correction procedure.

Prior experimental work (Ellis et al., 1999) has indicated that tracker distortion does not clearly influence the subjective sense of realism. However in that study the amount of distortion of the tracked data is described as ”relatively small” and seems to be much smaller than the distortion present in our system. While we have not performed such user studies, our informal observations indicate that our correction methods substantially enhance the subjective sense of immersion and the understanding of the 3D virtual scene.

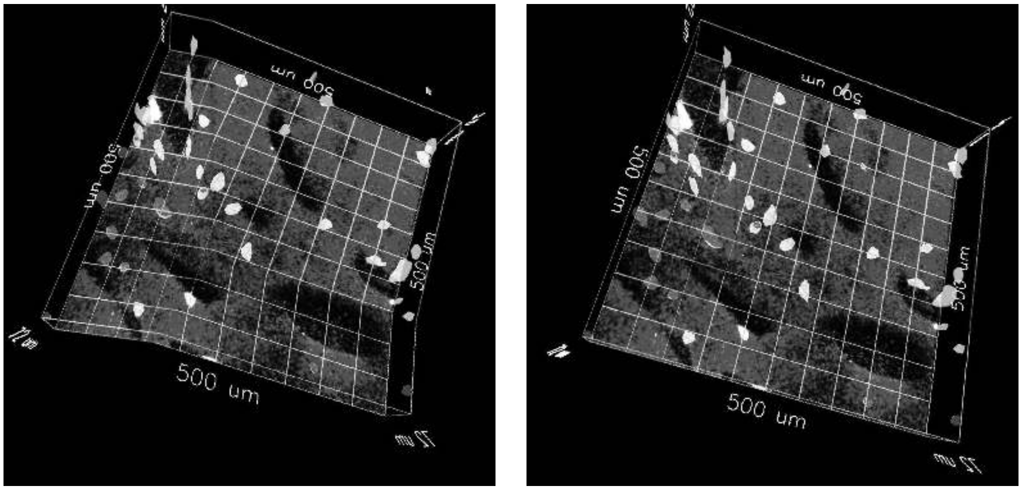

Figure 9: The image on the left shows the visual artifacts due to tracker error. The image on the right shows the same scene with tracker data corrected with the methods described here.

6 Conclusions and Future Work

It is clear that the tracker correction methods that we describe here produced results that substantially improve the accuracy of the tracker equipment and enhance the immersive experience. The performance of our methods is very satisfactory for the real-time environment in which it must operate.

We use a straight-forward method for interpolating rotational data (spherical weighted averages) that has not been applied to tracker calibration before. Unlike previous rotation correction methods this technique is firmly rooted in the geometry of the space of rotations. This technique can be used in other contexts for doing weighted averages of rotations. This method is used in conjunction with Delaunay tetrahedralization (not previously used in this context) and weights derived from barycentric coordinates. Currently our methods provide continuous, but not smooth interpolation; we intend to extend this approach to provide smooth 3D interpolation throughout the tracked volume.

Bringing together the use of spherical weighted averages with Delaunay tetrahedralization and barycentric coordinates provides a valuable approach to correcting tracker data based on samples at scattered points. The correction methods described here are in daily use in our immersive visualization environment and have yielded substantial improvement to the immersive experience.

7 Acknowledgments

We would like to thank the NIST Summer Undergraduate Research Fellowship program for enabling one of the co-authors, Whitney Austin, to come to NIST to work on this project. We are grateful to Alex Thibau who provided great help in the data collection phases of the project. We would like to acknowledge Stan Fowler for suggestions on the construction of our measurement apparatus. We would also like to thank Dr. Joy Dunkers of the Polymers Division of the Materials Science and Engineering Laboratory at NIST for the use of the data depicted in Figures 2, 1, and 9. Finally, we thank the reviewers for their many valuable suggestions.

8 Disclaimer

Certain commercial products may be identified in this paper in order to adequately describe the subject matter of this work. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the identified products are necessarily the best available for the purpose.

References

Ascension (2002). Flock of birds installation and operation guide. Technical report, Ascension Technology Corporation.

Borst, C. W. (2004). Tracker calibration using tetrahedral mesh and tricubic spline models of warp. In *Proceedings of the IEEE 2004 Virtual Reality Conference*, pages 19–26.

Bryson, S. T. (1992). Measurement and calibration of static distortion of position data from 3d trackers. In *Proceedings of SPIE Conference on Stereoscopic Displays and Applications III*, volume 1669, pages 244–255.

Buss, S. R. and Fillmore, J. P. (2001). Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics*, 20(2):95–126.

Curtis, W. D., Janin, A. L., and Zikan, K. (1993). A note on averaging rotations. In *Virtual Reality Annual International Symposium*, pages 377–385.

Czernuszenko, M., Sandin, D., and DeFanti, T. (1998). Line of sight method for tracker calibration in projection-based vr. In *Proceedings of the 2nd International Immersive Projection Technology Workshop*. Iowa State University.

Ellis, S., Adelstein, B., Baumeler, S., Jense, G., and Jacoby, R. (1999). Sensor spatial distortion, visual latency, and update rate effects on 3d tracking in virtual environments. In *Proceedings of the IEEE Virtual Reality Conference*, pages 218–221.

Ghazisaedy, M., Adamczyk, D., Sandin, D. J., Kenyon, R. V., and DeFanti, T. A. (1995). Ultrasonic calibration of a magnetic tracker in a virtual reality space. In *Proceedings of the IEEE Virtual Reality Annual International Symposium.*

Ikits, M., Brederson, J. D., Hansen, C. D., and Hollerbach, J. M. (2001). An improved calibration framework for electromagnetic tracking devices. In *Proceedings of IEEE Virtual Reality*, pages 63–70.

Jayaram, U. and Repp, R. (2002). Integrated real-time calibration of electromagnetic tracking of user motions for engineering applications in virtual environments. *Journal of Mechanical Design*, 124:623–632.

Kelso, J., Satterfield, S. G., Arsenault, L. E., Ketcham, P. M., and Kriz, R. D. (2003). Diverse: A framework for building extensible and reconfigurable device independent virtual environments and distributed asynchronous simulations. *Presence: Teleoperators and Virtual Environments*, 12(1):19–36.

Kenwright, D. N. and Lane, D. A. (1996). Interactive time-dependent particle tracing using tetrahedral decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):120–129.

Kindratenko, V. (1999). Calibration of electromagnetic tracking devices. *Virtual Reality: Research, Development, and Applications*, 54:139–150.

Kindratenko, V. (2000). A survey of electromagnetic position tracker calibration techniques. *Virtual Reality: Research, Development, and Applications*, 5(3):169–182.

Kindratenko, V. and Bennett, A. (2000). Evaluation of rotation correction techniques for electromagnetic position tracking system. In *Virtual Environments 2000: Proceedings of the Eurographics Workshop*, pages 13–22. Springer-Verlag.

Livingston, M. A. and State, A. (1997). Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments*, 6(5):532–546.

Nixon, M. A., McCallum, B. C., and Price, N. B. (1998). The effects of metals and interfering fields on electromagnetic trackers. *Presence: Teleoperators and Virtual Environments*, 7(2):204–218.

Raab, F. H., Blood, E. B., Steiner, T. O., and Jones, H. R. (1979). Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(5):709–717.

Saleh, T., Kindratenko, V. V., and Sherman, W. R. (2000). On using neural networks to calibrate electromagnetic tracking system. Technical report, National Center for Supercomputing Applications.

Shoemake, K. (1985). Animating rotation with quaternion curves. *Computer Graphices (SIGGRAPH '85 Proceedings)*, 19(5):245–254.

State, A., Hirota, G., Chen, D. T., Garrett, W. F., and Livingston, M. A. (1996). Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 429–438.

Taylor, B. N. and Kuyatt, C. E. (1994). Guidelines for evaluating and expressing the uncertainty of nist measurement results. Technical report, National Institute of Standards and Technology. NIST Technical Note 1297.

Zachmann, G. (1997). Distortion correction of magnetic fields for position tracking. In *Proceedings of Computer Graphics International*, pages 213–220.

Zachmann, G. (2000). *Virtual Reality in Assembly Simulation — Collision Detection*. PhD thesis, Darmstadt University of Technology, Germany, Department of Computer Science.

List of Figures