A Tracker Alignment Framework for Augmented Reality

Yohan Baillot and Simon J. Julier
ITT Advanced Engineering & Sciences
2560 Huntington Ave
Alexandria, VA 22303
{baillot,julier}@ait.nrl.navy.mil

Dennis Brown and Mark A. Livingston
Naval Research Laboratory
4555 Overlook Ave SW
Washington, DC 20375
{dbrown,markl}@ait.nrl.navy.mil

Abstract

To achieve accurate registration, the transformations which locate the tracking system components with respect to the environment must be known. These transformations relate the base of the tracking system to the virtual world and the tracking system's sensor to the graphics display. In this paper we present a unified, general calibration method for calculating these transformations. A user is asked to align the display with objects in the real world. Using this method, the sensor to display and tracker base to world transformations can be determined with as few as three measurements.

1. Introduction

Almost all Augmented Reality (AR) systems use a tracking system to capture motion of objects in the real world and map them into the computer generated environment. The most important relationship is head tracking — whenever the user moves their head in the "real world", the viewpoint in the graphics system should move accordingly. Similarly, if tracked props or interaction devices are moved in the real world, their movements should follow accordingly.

However, registration errors are the result of three error sources:

1. *Tracking errors.* These occur when the measurement returned by the tracker does not agree with the real pose of the tracker.

2. *Display calibration errors.* These arise when the optical characteristics of the display are unknown. It includes parameters such as field-of-view, distortion and centre of projection. Although these parameters can vary (for example, a camera with a zoom lens), in many applications these parameters are constant and generally it sufficient to calibrate the display once.

3. *Tracker alignment errors.* The sensor measurements must be transformed so that the graphics are rendered in the display at the correct viewpoint. The transformation requires the knowledge of the world-to-base transformation (where is the origin of the tracker coordinate system in the world?) and the sensor-to-manipulator transformation (how is the sensor placed relative to the display?) Although these parameters tend to stay constant with time, they can vary when (1) the tracker base is moved (e.g. the magnetic emiter of a magnetic tracker is moved) (2) the sensor is moved on the HMD (e.g. relocated on the HMD or HMD's headband adjusted).

The alignment problem we are concerned with is to determine where the base is with respect to the origin of the virtual world and where the manipulator is with respect to the sensor to which it is attached. Many authors have considered the problems of tracking errors and display errors [2, 4, 6–8]. However relatively few authors have considered the problem of tackling the alignment errors. There are many strategies that can be used to find the correct alignment of the tracker but there are no unified method that can be used for any tracking system.

This is a surprisingly difficult problem for prototyping and developing AR systems. In many systems, a flexible display such as the Sony Glasstron allow the user to fit it correctly to her head. However, every time an adjustment is done the transform between the display and the sensor is changing. In addition it might be necessary to change the location of the sensor on the headband, which pose each time the problem of locating the new sensor with respect to the display.

These problems are exacerbated if multiple sensing systems are used to form hybrid trackers. One example is the mobile AR system which is shown in Figure 1. This system uses multiple tracking devices to track position and orientation of the user's head. These devices are not referenced the same way and the relative attitudes between each others

Figure 1. The problems of tracker alignment are exacerbated when multiple tracking systems are used simultaneously. This mobile system uses an inertial navigation system and a GPS.

and with the display attitude are not known. In addition, the use of an inertial sensor stabilized by compass and a GPS require the knowledge of the frame of references within which these sensors give their measurement. Similar problems occur in hybrid tracking systems where the observations from multiple sensors are fused together in a central estimation algorithm such as a Kalman filter [15].

Early systems, such as those described in [1], often used open-loop calibration. That is, they essentially had to trust the measurement of the location of the tracking system in the environment and on the HMD. Bajura, for example, formed a closed-loop system with a video see-through system by using the HMD camera to provide direct feedback for the registration [4]. Although this approach solved the sensor to HMD transform, it did not solve the tracker emitter to world transform. Furthermore, the approach explicitly assumed a video see-through display and cannot be used for optical see-through display systems. To calibrate its set of trackers (Flock of Birds, Faro mechanical arm, and video cameras), the UNC ultrasound system exploited the fact that the regions of each device overlapped [13]. However, this configuration is highly specialized. Tuceyran proposed a calibration method for getting the unknown rigid transforms in the GRASP system [11]. While the method was presented as being applicable to any AR system, it uses a video see-through setup and a tracked pointer as part of the calibration procedure, however this is not the configuration of every AR systems. One means of calibrating multiple trackers was by basically letting the trackers "overlap" their operating region. Kutulakos and Vallino [10] used a projective world and markers to align the virtual objects

but this setup required a video see-through setup to work. In the general case when a video see-through system is used, there is no need to know the location of the tracker components in the world because the sensor is collocated with the display and the world is collocated with the pattern tracked. Fuhrmann [6] described a method for fast calibration in AR, but the description of the method is very succinct and therefore difficult to reproduce and it is not clear that the transform we are concerned with are actually determined. Tuceyran introduced the Single Point Active Alignment Method (SPAAM) [7, 14] to perform the calibration of an optical see-through HMD. However, this method required many points (minimum 6, recommended 12) to be sighted by one user, and stereo must be used to judge and align correctly the depth, which is a difficult task.

The structure of this paper is as follows. Section 2 defines the notation and discusses the problem statement in detail. In Section 3 we consider the problem of calculating the sensor-to-manipulator transformation when the world-to-base transformation is assumed to be known. Section 4 extends this to the case when neither the sensor-to-manipulator nor the world-to-base transformations are known. The implementation of this framework is outlined in Section 5 and a set of results for a test case are given in Section 6. Summary and conclusions are given in Section 7.

2. Problem Statement

Throughout this paper we use the following notation. Let a referential (or rigid coordinate system) X be written as X. Let XY be the motion or homogeneous transformation that aligns the X referential with a second referential Y[1]. Furthermore, from the properties of inverses of transformations, $\mathrm{YX} = (\mathrm{XY})^{-1}$.

The problem of aligning the head mounted display with the graphics system is illustrated in Figure 2. A user wears a tracked head mounted display. The part of the head mounted display responsible for generating the graphics is fixed to the manipulator with referential M. A sensor with referential S is rigidly attached to the headband of the head mounted display. The sensor base (origin of the tracking system) is B. Therefore, the tracking system actually measures $\mathbf{BS}$. To render the graphics properly, the attitude of the graphics display in the world ($\mathbf{WM}$) must be known.

From the figure, these transformations are given by

$$\mathbf{WM} = \mathbf{WB} \cdot \mathbf{BS} \cdot \mathbf{SM}. \tag{1}$$

In other words, $\mathbf{WM}$ can be calculated if the world-to-base ($\mathbf{WB}$) and sensor-to-manipulator ($\mathbf{SM}$) transformations are known. In some situations these quantities can be

---

[1]In other words, if the transformation matrix of X is $\mathbf{M}_X$ and the transformation matrix of Y is $\mathbf{M}_Y$, then $\mathrm{XY} = \mathbf{M}_X^{-1}\mathbf{M}_Y$.
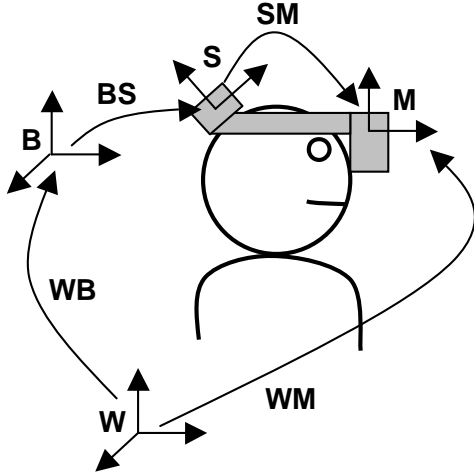
Figure 2. The referentials and transformations which are relevant for tracker alignment.

determined in advance. However, in some circumstances it can be difficult to determine these quantities in an offline manner.

The sensor-to-manipulator transformation $\mathbf{SM}$ can be difficult to calculate for two reasons. The first is that it can be difficult to determine the point at which the measurement is made. Sensing devices are of finite size, and it can be difficult to work out the point within the sensor which is being tracked. The second is that the sensor and manipulator might not be rigidly attached to one another. The Sony Glasstron, for example, includes a hinged joint which allows the display to be translated and rotated with respect to the head band. To counterbalance the weight distribution on the user's head, it is not always possible to attach the tracker to the display itself but rather to the headband. Therefore, as a user adjusts a display (either during a calibration procedure or even during normal use), $\mathbf{SM}$ is changed. Even if the display is fixed with respect to the sensor's mounting point, the sensor might be installed in a manner such that its transformation with the display is not intuitive[2].

At first sight, it might appear that calculating the world-to-base transformation $\mathbf{WB}$ is significantly simpler. The prevailing assumption appears to be that this is a fixed, easy to identify and easy to measure property. As a result, with careful measurement, the transformation can be calculated. However, there are two difficulties with this approach. The first is that it is not always possible to accurately measure

---

[2]One common practice is to try to mount the trackers as 'horizontally as possible on the display". Assuming the tracker is horizontal, only the yaw needs to be corrected. However, in general the sensor will not be properly aligned and this leads to coupling of pitch and yaw rotations in highly non intuitive ways.

the base of the tracker in the physical world. In our mobile augmented reality work indoor users (who are tracked with by an InterSense IS900) must be able to see and interact with outdoor mobile users (whose positions are tracked using a GPS). Therefore, the base of the IS900 must be expressed in world fixed (longitude/latitude) coordinates. However, it is not immediately obvious how this can be calculated[3]. A second problem is that some tracking systems simply do not have a tangible, physical source which corresponds to the tracker base. The InterSense InertiaCube 2 (IC2), for example, utilizes magnetometers to constrain the yaw of an orientation tracker. The base of the magnetometer is magnetic north. However, as is well known, local magnetic anomalies can distort the magnetic field. In effect the sensor base changes as the tracker moves through the environment.

We now describe two calibration procedures. The first calculates $\mathbf{SM}$ under the assumption that $\mathbf{WB}$ is known. The second generalizes this result to determine both $\mathbf{WB}$ and $\mathbf{MS}$.

3. Single Point Calibration Technique

The single calibration point technique uses a single measurement to calculate $\mathbf{SM}$. The technique is built on the observation that $\mathbf{SM}$ can be calculated by inverting Equation 1:

$$\mathbf{SM} = \mathbf{SB}\,.\,\mathbf{BW}\,.\,\mathbf{WM}. \qquad (2)$$

The difficulty with this approach is that the true location of manipulator, expressed in world coordinates, must be known. Because we are using an optical see-through display, this can be achieved by asking the users themselves to align the contents of the display directly with objects in the environment.

The calibration procedure is illustrated in Figure 3: the environment contains two calibration points — a reference point on the ground (C) and a calibration mark on the wall (G). The locations of these marks must be known within the coordinate system of the model. The display renders graphics as if the head mounted display was placed at a known value of $\mathbf{WM}$. The user is asked to stand on C and align the contents of the display with G. When the two are aligned, the user has positioned the display at the known $\mathbf{WM}$ location and a tracker sample (a sample of $\mathbf{SB}$) is recorded.

The transformation $\mathbf{WM}$ can be decomposed into a transformation from the world to C, and from C to M:

$$\mathbf{WM} = \mathbf{WC}\,.\,\mathbf{CM}. \qquad (3)$$

Under the assumption that C is not rotated[4], $\mathbf{WC}$ is of the

---

[3]Because the tracking system is mounted indoors, GPS cannot be used to directly measure position.

[4]This assumption is valid because C specifies the location of the user's feet. The orientation of the user's head is determined by $\mathbf{CM}$.
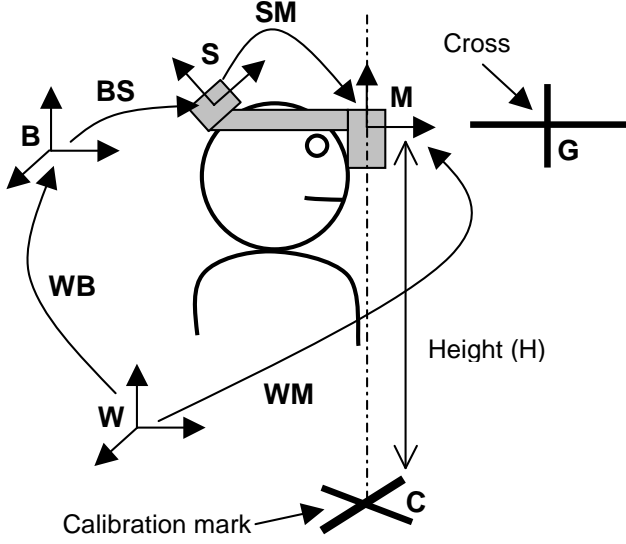
Figure 3. The single point calibration technique. Assuming that the world-to-base transformation is known, the sensor-to-manipulator transformation can be calculated by asking the user to align the contents of the display with appropriate objects in the environment.

form:

$$\mathbf{WC} = \begin{bmatrix} 1 & 0 & 0 & x_{wc} \\ 0 & 1 & 0 & y_{wc} \\ 0 & 0 & 1 & z_{wc} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

To calculate $\mathbf{CM}$ we use three assumptions. The first assumption is that the user looks directly at G. Therefore, when the display is correctly aligned with the real world, G is projected into the center of the screen[5]. The second assumption is that that the tracker is aligned such that the roll component is zero. The third assumption that the height of the tracker off the ground, H, is known.

Decomposing $\mathbf{CM}$ into a pure translation and a pure rotation,

$$\mathbf{CM} = \mathbf{T}_{cm} \cdot \mathbf{O}_{cm}. \quad (5)$$

The first component is the vertical translation,

$$\mathbf{T}_{cm} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

The second component is the rotation needed to align the displays. Using the assumption that the roll angle is zero,

---

[5]This is only true if the display is monoscopic. If it is stereoscopic, the projection is shifted for each eye according to the eye separation.

this can be decomposed into an azimuth rotation ψ (about the body-fixed z-axis) followed by an elevation rotation ϕ (about the body-fixed x-axis)

$$\mathbf{O}_{cm} = \mathbf{R}_z \cdot \mathbf{R}_x. \quad (7)$$

ψ and ϕ can be calculated from $\mathbf{MG}$. $\mathbf{MG}$ is the transformation from the manipulator to the calibration mark. The translation (x_{MG}, y_{MG}, z_{MG}) is not a function of the orientation. Therefore,

$$\psi = \tan^{-1}\left(\frac{-x_{MG}}{y_{MG}}\right)$$

$$\phi = \sin^{-1}\left(z_{MG}/\sqrt{x_{MG}^2 + y_{MG}^2 + z_{MG}^2}\right)$$

The resulting procedure is simple and effective. The user is merely asked to stand in a known location and align the contents of the head mounted display with the environment. We have extensively used it in our own system and have applied it to many demonstrations using the system at many different sites. This calibration is easy and allows the flexibility to place the sensor anywhere on the user head and with any orientation.

However, this algorithm relies on the assumption that $\mathbf{CM}$ can be calculated. In the approach presented here, this is equivalent to assuming that the height of the display off the ground, H, is known. There are two ways of addressing this problem. The first is to attempt to measure the user height accurately. The second is to place the calibration target as far from the user as is practical. However, if more accuracy is required, the SPAAM method could be used [14]. Another possibility is to use a video see-through display looking at a known landmark. In this specific case, the camera is collocated with the graphics referential (the manipulator) and therefore the pose of the camera recovered by vision tracking during the calibration phase directly gives the pose of the manipulator. In the case where an optical see-through display must be used, a similar approach could be done by rigidly attaching a camera to the display and calibrating the camera to display transformation once for all. We are currently working on such a calibration method[6].

---

[6]The calibration of the properties of an optical see-through display (such as the field of view and the distortion) are done using a calibration grid on which graphical patterns seen through the display are aligned by a user. At the same time, a camera attached to the display can use the same calibration grid to locate itself with respect to the grid. Since one of the result of the optical calibration of the display is to determine where is the focal point of the display, the transform from the camera focal point to the manipulator (or graphical referential). Once know, if we use some known pattern instead of the the calibration cross currently used, then the camera can locate itself precisely with repect to the pattern at calibration time, and consequently the graphics referential can be located. In effect the user will not be performing the calibration but roughly aligning the field of view of the camera so that it can see the pattern. The camera will then locate itself with respect to pattern and provide and precise attitude for the manipulator, much precise that if it was obtained by the user.
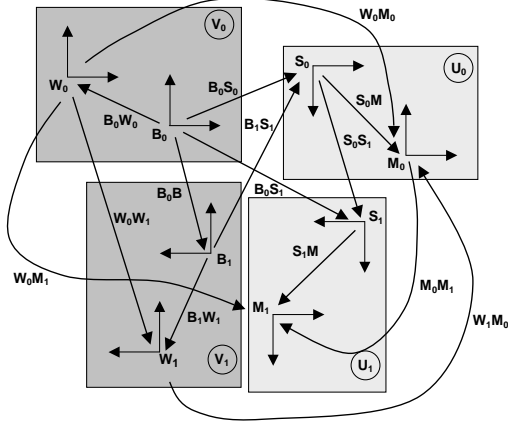
Figure 4. The Multi Point Calibration technique.

Although this approach is effective for cases where the base is known, it cannot be used when the pose of the base is unknown. The next section introduces a method to solve this problem.

4. Multiple Point Calibration Technique

When the world-to-base transformation $\mathbf{WB}$ is not known, the single point calibration technique described in the last cannot be used because Equation 2 cannot be evaluated. However, the necessary information can be gleaned from looking at how the *relative* transformations which occur when the user calibrates on a pair of calibration points.

Consider the situation shown in Figure 4: U_0 and U_1 represent the user head that includes the sensor S and manipulator M (which in this specific case is the graphical referential). V_0 represents the rigid group that include the world W and the base of the tracking system B. When the user moves her head between two calibration marks (motion from U_0 to U_1), the sensor produces the motion S_0S_1 and the manipulator produces the motion M_0M_1. S_0S_1 and M_0M_1 are related through the transformation $\mathbf{S}M$ between the sensor and the manipulator. The problem can be inverted to find $\mathbf{B}W$, the transformation between the base and the world, using no additional motion from the user. In effect, a motion of the head with respect to the world can be seen as a motion of the world with respect to the head. In this case, the head is fixed in U_0 and the world and tracker base are moving from V_0 to V_1. In this case B_0B_1 and W_0W_1 are related through the transformation $\mathbf{B}W$.

More formally, suppose the user stands at two different locations (C_1 and C_2) and looks at two different calibration marks (G_1 and G_2). The sensor referentials in these two locations are S_1 and S_2, and the manipulator referentials are M_1 and M_2.

Writing out Equation 2 for each measurement,

$$\mathbf{SM} = \mathbf{S}_1\mathbf{B} \cdot \mathbf{BW} \cdot \mathbf{WM}_1 \qquad (8)$$

$$\mathbf{SM} = \mathbf{S}_2\mathbf{B} \cdot \mathbf{BW} \cdot \mathbf{WM}_2. \qquad (9)$$

Rearranging Equation 8,

$$\mathbf{BW} = \mathbf{BS}_1 \cdot \mathbf{SM} \cdot \mathbf{M}_1\mathbf{W}.$$

Substituting into Equation 9,

$$\mathbf{SM} = \mathbf{S}_2\mathbf{B} \cdot \mathbf{BS}_1 \cdot \mathbf{SM} \cdot \mathbf{M}_1\mathbf{W} \cdot \mathbf{WM}_2.$$

Postmultiplying both sides by $\mathbf{M}_2\mathbf{W} \cdot \mathbf{WM}_1$ gives

$$(\mathbf{S}_1\mathbf{B} \cdot \mathbf{BS}_2) \cdot \mathbf{SM} = \mathbf{SM} \cdot (\mathbf{M}_1\mathbf{W} \cdot \mathbf{WM}_2). \qquad (10)$$

This is exactly the same as the so-called "hand-eye" calibration framework problem which is frequently encountered in robotics [12]. In a typical robotics application, a manipulator is rigidly attached to the actuator. The transformation from the actuator to the manipulator is not known. However, both the actuator and the manipulator contain tracking systems. In a typical configuration, the actuator might be a robotic arm (whose geometry is known and whose joint angles are measured) and the manipulator contains a camera. The problem is conventionally posed as

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B} \qquad (11)$$

where $\mathbf{A}$ is the motion of the first referential, $\mathbf{B}$ is the motion of the second referential, and $\mathbf{X}$ is the transformation that aligns the first referential with the second one.

If $\mathbf{A}$ and $\mathbf{B}$ could be measured perfectly, solving this equation would be a trivial linear algebra problem. However, because $\mathbf{A}$ and $\mathbf{B}$ are measured by noise-corrupted sensors, more sophisticated techniques must be used to ensure that $\mathbf{X}$ is a properly formed homogeneous transformation matrix. Within the robotics literature, a number of different approaches have been proposed. For this paper we used a closed-form solution developed by Park and Martin [5] . This solution, described in detail in the appendix, uses Lie Bracketing Algebra and matrix logarithms and yields an extremely compact and easy to implement solution.

A similar approach can be taken to solve $\mathbf{WB}$. Substituting Equation 8 into Equation 9,

$$\mathbf{S}_1\mathbf{B} \cdot \mathbf{BW} \cdot \mathbf{WM}_1 = \mathbf{S}_2\mathbf{B} \cdot \mathbf{BW} \cdot \mathbf{WM}_2.$$

Premultiplying by $\mathbf{BS}_2$ and post multiplying by $\mathbf{M}_1\mathbf{W}$,

$$(\mathbf{BS}_2 \cdot \mathbf{S}_1\mathbf{B}) \cdot \mathbf{BW} = \mathbf{BW} \cdot (\mathbf{WM}_2 \cdot \mathbf{M}_1\mathbf{W}). \qquad (12)$$

Once again, this is in the form of the "hand-eye" calibration problem and can be solved in exactly the same manner[7].

---

[7]Another way to consider the problem is that, to solve for $\mathbf{SM}$, we

5. Implementation

The calibration framework described in the previous section was implemented within the Battlefield Augmented Reality System (BARS) [9]. The interactive authoring system described in [3] was extended to allow users to annotate an environment model with a set of N calibration cross referentials $\mathbf{C}_i$ and calibration marks $\mathbf{G}_i$. $\mathbf{SM}_i$ is calculated from Equation 3 for each $\mathbf{C}_i/\mathbf{G}_i$ pair.

When solving Equations 10 and 12, it is possible to exploit the fact that Equations 8 and 9 apply for *any* pair of relative transformations. For example, when solving Equation 10, it is possible to construct $N(N-1)/2$ equations of the form

$$(\mathbf{S}_i\mathbf{B} \boldsymbol{.} \mathbf{BS}_j) \boldsymbol{.} \mathbf{SM} = \mathbf{SM} \boldsymbol{.} (\mathbf{M}_i\mathbf{W} \boldsymbol{.} \mathbf{WM}_j) \boldsymbol{.}$$

where $i, j \in [1, \ldots, N]$ and $i \neq j$. As shown in the appendix, this can greatly increase the performance of the solution.

We now demonstrate the use of this approach at a test environment.

6. Example

The alignment framework was used to align the sensors in an indoor mobile augmented reality system. The geometric model of the environment is shown in Figure 5 — the environment consists of a single room and a number of pieces of laboratory equipment. The model has been augmented to include the calibration crosses (the nth cross is labeled AXXBn) and the calibration marks. Most of the calibration marks in this model are preexisting features such as the corners of doors or walls. One artificial calibration mark, a cross, can be see on the right of the picture.

A set of 7 calibration crosses and calibration marks were created. To test the effectiveness of this configuration, a sensitivity analysis was performed using a Monte Carlo analysis. It was assumed that the measurement error (which includes tracker error and misalignment errors by the user) has a standard deviation of 0.05m in position and 0.5 degrees in orientation[8]. Figures 6(a) and 6(b) show the 2σ standard deviation of the error in $\mathbf{SM}$. These plots show that the 2σ error in $\mathbf{SM}$ is between 0.05m (X and Z) and 0.15m (Y), and the orientation error is between $0.5°$ (X and Y) and $1.2°$ (Z). We believe that the errors in Y (for position) and Z (for orientation) are much larger as a result of the calibration configuration which was used — namely that all of the calibration marks were at approximately the same

---

assume that the tracker base B is fixed and allow the manipulator M to move. To solve for $\mathbf{WB}$, we assume that M is fixed and allow B to move.

[8]Positions are given in metres. Orientations are expressed in Euler angles using rotation about fixed XYZ axes (the same convention is used by Java3D).
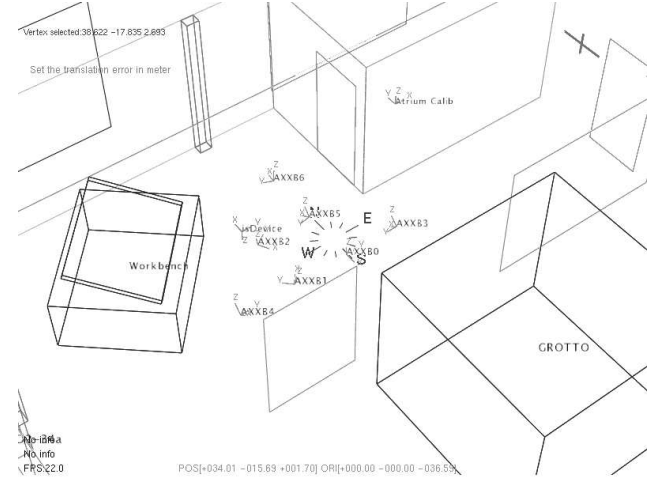

**Figure 5. The sample calibration environment.**

height. The simulation studies confirm that, with a more uniform distribution of marks in three-dimensions, the errors in all rotation angles and positions decrease at approximately the same rate.

The figures also illustrate that, as the number of combinations increase, the magnitude of the error is reduced. Figures 6(c) and 6(d) show the standard deviations of the errors in translation and orientation of $\mathbf{WB}$. The results are very similar to those for $\mathbf{SM}$: the error on the position were ranging between 0.35 and 0.7m, and the error on the orientation were ranging between $0.4°$ and $1.2°$.

These measurements were confirmed by conducting an actual calibration experiment. The tracking system is an InterSense IS900LAT. The user stood at each of the 7 calibration points and was asked to align the display with the appropriate calibration mark. Through careful (and laborious) measurement, the value of $\mathbf{WB}$ was accurately obtained. Table 1. As can be seen, the results are extremely accurate for almost all results and are, in fact, significantly better than those predicted by the covariance analysis. In this experimental configuration, $\mathbf{SM}$ could not be accurately measured independently. However, because observed registration errors in the calibrated display were small, we believe that it was estimated accurately.

7. Conclusions

In this paper we have presented a method for aligning trackers in augmented reality systems. The approach described here is novel in two respects. First, the alignment process is extremely easy and intuitive — a user is asked to stand at a known location and align the display with known objects in the environment. Second, the method is capable
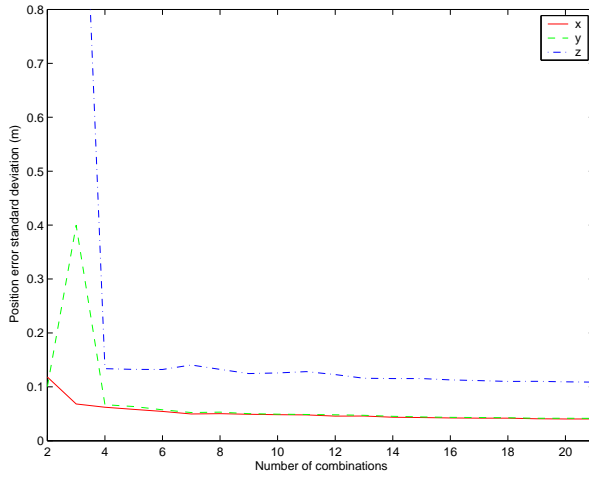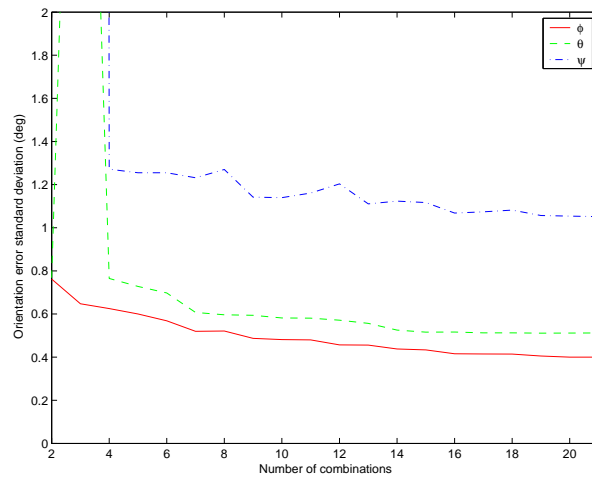
(a) Translation error in **SM**.
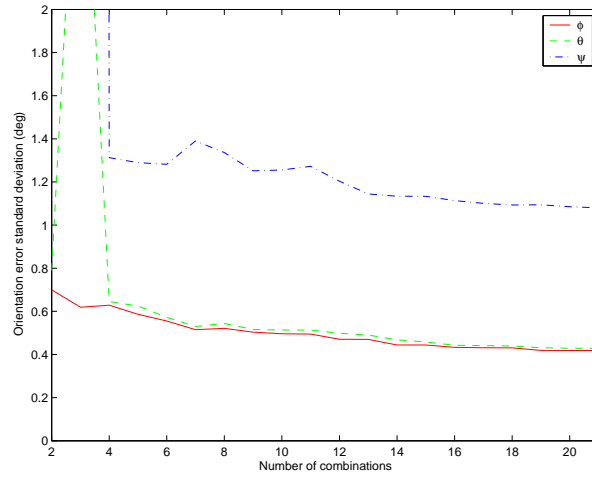
(b) Orientation error in **SM**.
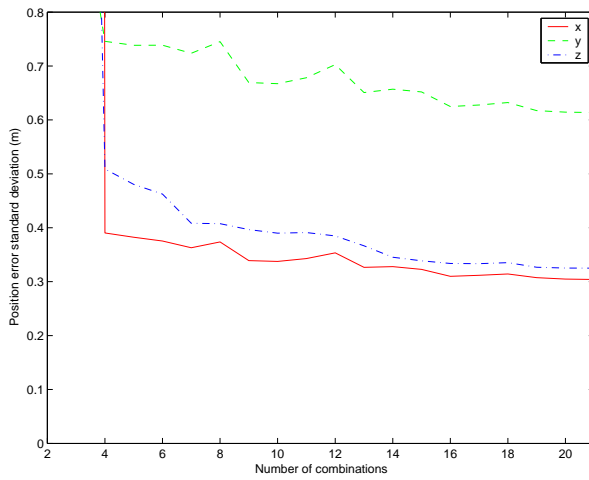
(c) Translation error in **BW**.

(d) Orientation error in **BW**.

Figure 6. The error in the calculated values of SM **and** BW **as a function of the number of combinations used. Each figure plots the** 2σ **(95% probability) error bounds.**

	True BW	Calculated BW
x	32.66	32.595
y	-15.07	-14.867
z	0.533	0.53
ϕ	-180	-179.913
θ	0	0.774
ψ	90	90.466

Table 1. The calculated and true values of the base-to-world transformation.

of calculating the sensor-to-manipulator and world-to-base transformations in the same step.

We shall extend this calibration method in the following ways:

- Develop tools that will provide user with feedback in designing a calibration scheme. For example, the algorithm we used to solve the hand-eye calibration problem cannot be applied if the trace of any transformation matrix is -1. Such conditions can be detected as the marks and crosses are being created and surveyed in.

- Explore schemes to automatically optimize the placement of calibration marks to improve the accuracy of the calibration result, based on using this tool and understanding in which situations the logarithm fails.

- Explore the effect of other solvers to see how these are affected by noise and / or marker placement.

- Use this algorithm to solve the relative placement of two trackers used to form an hybrid tracker.

A. Solving the Calibration Equations

The calibration framework relies on the ability to solve the equation

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B} \tag{13}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}$ are transformation matrices. This problem is extremely important in the field of robotics where it is known as the "hand-eye" calibration problem. Given a set of N measurements of $\mathbf{A}$ and $\mathbf{B}$, find $\mathbf{X}$ such that

$$\mathbf{A}_1 \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}_1$$
$$\vdots \qquad \vdots$$
$$\mathbf{A}_N \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}_N$$

A number of different solutions have been developed for this problem. Most of these solutions are iterative, and are typically designed for automatic systems where many hundreds of samples can be taken. Accurate solvers which require few measurements are extremely important.

For this paper, we used an approach which was developed by Park and Martin in [5]. Despite the theoretically complexity of the algorithm (it is based on the matrix logarithm of the transformation matrix) it is extremely easy to implement.

Let $\mathbf{\Theta} \in SO(3)$ be any rotation matrix and let $\mathbf{b} \in \mathbb{R}^3$ be the translation. Therefore, any valid transformation matrix $\mathbf{M}$ has the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{\Theta} & \mathbf{b} \\ 0 & 1 \end{bmatrix}.$$

If trace $[\mathbf{\Theta}] \neq -1$, the logarithm of this matrix is

$$\log \mathbf{M} = \begin{bmatrix} [\omega] & \mathbf{A}^{-1}\mathbf{b} \\ 0 & 0 \end{bmatrix}$$

where $[\omega] = \log \mathbf{\Theta}$ and $\mathbf{A}$ is a matrix whose is irrelevant for solving the calibration problem.

Let ϕ be

$$\phi = \cos^{-1}\left(\frac{\text{trace}\,[\mathbf{\Theta}] - 1}{2}\right) \tag{14}$$

The matrix logarithm $[\omega]$ is

$$[\omega] = \frac{\phi}{2\sin\phi}\left(\mathbf{\Theta} - \mathbf{\Theta}^T\right) \tag{15}$$

This is a skew symmetric matrix

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{16}$$

Therefore, $[\omega]$ can be parameterised as the vector $\boldsymbol{\omega}$ where

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

Let $\boldsymbol{\alpha}_i$ be the matrix logarithm of measurement $\mathbf{A}_i$ and $\boldsymbol{\beta}_i$ be the matrix logarithm of measurement $\mathbf{B}_i$.

The Park-Martin algorithm [5] attempts to find $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} \mathbf{\Theta}_X & \mathbf{b}_X \\ 0 & 1 \end{bmatrix}.$$

The algorithm decomposes the solution into two subproblems. The first is to calculate the rotation of $\mathbf{\Theta}_X$. This can be carried out independently of the translations. The second problem calculates $\mathbf{b}_X$ using the calculated value of $\mathbf{\Theta}_X$

The rotation matrix $\mathbf{\Theta}_X$ is chosen to minimise the cost function

$$\eta_1 = \sum_{i=1}^{p} ||\mathbf{\Theta}_X \cdot \boldsymbol{\beta}_i - \boldsymbol{\alpha}_i||^2 \tag{17}$$

The optimal solution is

$$\Theta_X = \left(\mathbf{M}^T \boldsymbol{.} \mathbf{M}\right)^{-1/2} \boldsymbol{.} \mathbf{M}^T \tag{18}$$

where

$$\mathbf{M} = \sum_{i=1}^{p} \boldsymbol{\beta}_i \boldsymbol{.} \boldsymbol{\alpha}_i^T. \tag{19}$$

If $p = 2$, the third measurements are synthesised as $\boldsymbol{\alpha}_3 = \boldsymbol{\alpha}_1 \times \boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_3 = \boldsymbol{\beta}_1 \times \boldsymbol{\beta}_2$.

The matrix $\mathbf{M}$ has the property that it is *always* guaranteed to be orthonormal even if the data is noisy.

The second optimisation solution minimises

$$\eta_2 = \sum_{i=1}^{p} \| \left(\Theta_{A_i} - \mathbf{I}\right) \boldsymbol{.} \mathbf{b}_X - \Theta_X \boldsymbol{.} \mathbf{b}_{B_i} + \mathbf{b}_{A_i} \|^2. \tag{20}$$

This can be expressed as a standard least squares minimisation problem and its solution is

$$\mathbf{b}_X = \left(\mathbf{C}^T \boldsymbol{.} \mathbf{C}\right)^{-1} \boldsymbol{.} \mathbf{C}^T \mathbf{d}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} - \Theta_{A_1} \\ \vdots \\ \mathbf{I} - \Theta_{A_p} \end{bmatrix}$$

and

$$\mathbf{d} = \begin{bmatrix} \mathbf{b}_{A_1} - \Theta_X \boldsymbol{.} \mathbf{b}_{B_1} \\ \vdots \\ \mathbf{b}_{A_p} - \Theta_X \boldsymbol{.} \mathbf{b}_{B_p} \end{bmatrix}.$$

This equation can be solved even if only 2 measurements are used.

References

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[2] R. Azuma, G. Bishop. Improving static and dynamic registration in optical see-through hmd. In *Proceedings of SIGGRAPH'94*, pages 197–204, July 1994.

[3] Y. Baillot, D. Brown, and S. Julier. Authoring of Physical Models Using Mobile Computers. In *Proceedings of International Symposium of Wearable Computers*, 2001.

[4] M. Bajura and U. Neumann. Dynamic registration correction in video-based augmented reality systems. *IEEE Computer Graphics and Applications*, (5):52–60, 15 1995.

[5] F.C. Park and B. Martin. Robot Sensor Calibration: Solving AX=XB on the Euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, October 1994.

[6] A. Fuhrmann, D. Schmalstieg and W. Purgathofer. Fast Calibration for Augmented Reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 166–167, London, UK, 20–22 December 1999.

[7] Y. Genc, M. Tuceryan, and N. Navab. Pratical Solution for Calibration of Optical See-Through Devices. In *Proceedings of International Symposium of Mixed and Augmented Reality*, 2002.

[8] A. Janin, D. Mizell, and T. Caudell. Calibration of head-mounted display for augmented reality applications. In *Proceedings of Virtual Reality International Symposium*, 1993.

[9] S. Julier, Y. Baillot, M. Lanzagorta, D. Brown, and L. Rosenblum. BARS–Battlefield Augmented Reality System. In *NATO Symposium on Information Processing Techniques for Military Systems*, 2000.

[10] K. N. Kutulakos and J. R. Vallino. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 4(1), – 1998. ISSN 1077-2626.

[11] D. M. Tuceryan. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):255–273, 1995.

[12] Y. C. Shiu and S. Ahmad. Finding the Mounting Position of a Sensor by Solving a Homogenous Transform Equation of the Form AX equals XB. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1666–1671, Raleigh, NC, USA, 1987.

[13] A. State, G. Hirota, D. T. Chen, W. F. Garrett and M. A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 429–438. ACM SIGGRAPH, Addison Wesley, 1996.

[14] M. Tuceryan, and N. Navab. Single Point Active Alignment Method (SPAAM) for Optical See-Through HMD Calibration for AR. In *Proceedings of International Symposium of Augmented Reality*, 2000.

[15] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *IEEE Virtual Reality 2001*, pages 71–78, Mar. 2001.