

A User-Friendly Approach to Registering a 6-DOF Tracker with a Display

Andrew Forsberg and Nicholas Yang
Brown University
Department of Computer Science
Providence, RI 02912
{asf,nyang}@cs.brown.edu

ABSTRACT

Registering a six degree-of-freedom tracker with a display can be time-consuming and frustrating. Yet, establishing a consistent coordinate system shared by a display and a tracker is a necessary first step for many virtual reality applications.

We present a new approach for registering a tracker's and a display's coordinate system that has simple instructions, can be completed in under a minute, and does not involve understanding or manipulating any underlying linear algebra. A related technique for registering tracked stereo glasses given an arbitrarily mounted head-tracker is also presented.

We demonstrate, through a series of experiments, that the registration technique is fast and easy to perform, and that it can effectively be used as a basis for: 1) perceiving depth from stereopsis, 2) changing vantage points of a virtual scene with head movements, and 3) performing six degree-of-freedom manipulations with hand-held tracked objects. Data collected during our experiments also suggested approaches that may help developers improve their visualizations.

KEYWORDS: registration, 6 degree-of-freedom tracking, virtual reality, Fishtank VR.

INTRODUCTION

Three attributes of many immersive virtual reality systems are:

- stereo viewing
- head-tracked viewing
- six degree-of-freedom interaction techniques

To provide these, virtual reality (VR) applications require a clear relationship between a 6 degree-of-freedom (DOF) tracker coordinate system and the display. While there are infinitely many possible unifying coordinate systems, usually a tracker's coordinate system is transformed into a display's

coordinate system, since tracked motions are performed relative to visuals shown on the display.

For example, during a positioning task one would expect that if a tracker was moved to the right then the virtual object it was a proxy for should also move to the right. Similarly, if a head-tracked user shifted their head to the right, you would expect the virtual eye point to shift to the right commensurately. Similar expectations apply for rotations.

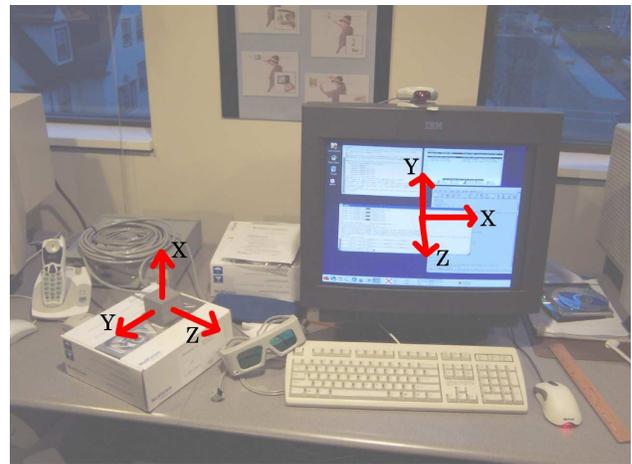


Figure 1: An example of common display and tracker coordinate systems. The display's origin is centered on the display and the positive X-axis extends "to the right", Y-axis extends "up", and Z-axis extends "out of the display". A possible tracker coordinate system is shown to the left of the monitor near the stereo glasses and is very different in position and orientation from the display coordinate system.

However, a tracker's default coordinate system is almost never the same as the display's for a variety of reasons. For example, it physically is not possible in some cases. Due to technology limitations it is sometimes necessary to install tracking hardware some distance away from a display. Other installation constraints, such as mounting surfaces, may influence the tracker's coordinate system. A translation, rotation, and scale may need to be applied to tracker samples to transform them to the desired coordinate system. A common situation is illustrated in Figure 1.

Typically, finding the required transforms is left to the programmer or user and can be a time-consuming and non-intuitive process. Furthermore, if a VR application will be used at a remote site, someone unfamiliar with tracker-display registration may be required to learn how to perform the registration.

PREVIOUS WORK

This section presents work related to registering a 6-DOF tracker with a display and evaluates its quality. In general, immersive VR display systems include tracked glasses that help produce a stereo effect. Additional trackers, possibly mounted on physical objects, may be used to allow a user to interact with the application. Examples of such systems include CAVEs, Powerwalls, and Fishtank VR.

This paper focuses on registration rather than calibration. Calibration attempts to minimize the difference between the coordinates of a tracked object in some coordinate system and the coordinates reported by the tracking system. Calibration is necessary because most tracking technologies have limitations. Given a calibrated system, registration is an orthogonal process that relates two coordinate systems through a linear transformation. If a tracking system is not calibrated accurately, even a properly registered system may produce errors in the images being displayed. Many calibration techniques exist including [7][1][5], some of which report reducing positional errors by up to 79%.

Augmented reality (AR) requires very accurate registration in order to accurately overlay graphics on the user's view of the surroundings. Our approach was not targeted at registering an AR system; it was made for making registration easier for the simpler types of systems described above. We were not able to apply our registration technique to an AR system but suspect it is not appropriate for achieving the degree of registration accuracy required for AR.

Fuhrmann presents a comprehensive process for registration that can be accomplished by a novice user [3]. The process can register projective and head-mounted displays, tracking systems, and tracked props. Fuhrmann's technique involves four distinct steps, whereas ours involves two. Both Fuhrmann's technique and ours are limited by the accuracy of human input. When determining the interpupillary distance and eye-to-tracker offset in Fuhrmann's technique, users must hold their heads in a constrained position facing the center of the display while aligning a stylus tip with multiple markers. Fuhrmann's system requires an accurate tracking system over the working volume, in addition to having a stylus. Our approach, on the other hand, can still work when error exists within the working volume (e.g. electromagnetic interference near a CRT), and our approach also does not use a stylus.

[4] proposes a technique for registration and calibration that was implemented on the Visual Haptic Workbench. Some of

their goals are similar to ours, such as stressing the importance of quick and easy registration with no external dependencies. Their approach follows a series of different steps to achieve registration. It requires more individual steps than our method. It is worth noting that some combinations of displays and trackers are not always accurate near the corners of a display. [2] proposed a multi-point sampling technique for registering a tracker with a responsive workbench-like display. It, too, requires more steps than the technique presented here and may give a poor registration because it requires sampling 3D positions directly on the display surface, which may not be possible in cases where the display interferes with the tracking system.

Some installations in use today are registered in an ad-hoc manner, by determining a series of linear transformations based on observation and measurement of the physical environment.

In terms of evaluating a registration's quality, [6] presented a collection of heuristics and simple tests for evaluating the quality of a projection-based virtual reality display. We repeated many of these tests in our experiments below.

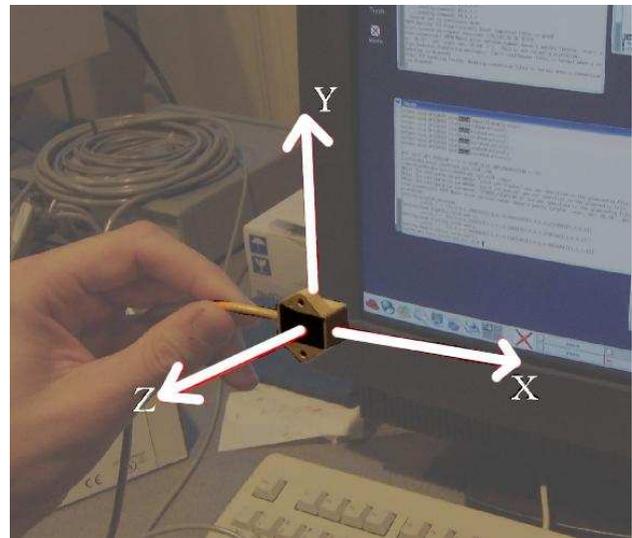


Figure 2: The local coordinate system of a tracked sensor. The origin is the point at which the three axes intersect.

REGISTRATION BY EXAMPLE

The key idea behind our approach is to specify the desired coordinate system by example. Using only a ruler to help measure a point some distance away, orthogonal to the display, first a tracker and then tracked glasses are posed and recorded at a known point and orientation relative to the display. The tracker data from these two steps is sufficient to register the system.

The following is a description of the process from the user's point of view.

Basic Registration

Our goal is for tracker values to be reported in the display's coordinate system rather than its own, as illustrated in Figure 1. If you study a tracker used in your tracking system and consult the user's manual, then you can discover its local origin as well as its local positive X , Y , and Z axes. See Figure 2 for an illustration of a particular tracker's local coordinate system. After a little manipulation, a tracker can be tumbled until its axes are oriented to match the display's. Next, the tracker can be moved to a known point relative to the display while maintaining this orientation. When both conditions are met, if the composite 4x4 matrix C containing the tracker's rotational and translational components is captured from the tracking system, then all the data necessary for registration has been obtained.

In practice, there is often more tracker error near a display. Therefore a known point some distance away from the display can be used instead. We used a ruler to measure a point one foot away from the center of the display. We offer two suggestions for this approach. First, it is important that the ruler be orthogonal to the plane of the display. We found this is easier if one holds the ruler in one hand near where it meets the display. Second, it helps to align the top edge of the ruler on a graphical object drawn in the center of the display while posing the tracker on the other end of the top edge of the ruler. Figure 3 illustrates these suggestions. In our implementation, we instructed the user to press a key on the keyboard to indicate when they were ready to sample C . The user then had about 4 seconds to position and hold the tracker at the known position in the correct pose.



Figure 3: To do the basic registration, a tracker is oriented such that its local coordinate system is aligned with the display's coordinate system, and it is positioned at a known point relative to the display. The red marker, emphasized here for clarity, indicates the position of the known point projected onto the display. The small image attached to the lower-right of the monitor is a picture reference that we used in our study, to help users understand the tracker's local coordinate system.

IMPLEMENTATION While the person doing the registration need not understand the linear algebra, the implementor does. Understanding equations (1) and (9) below is sufficient to implement our registration approach. Their derivations are also given. We found it easier to understand and explain the proof of our method using algebraic equations rather than through figures because of the difficulty of conceptualizing and visualizing the underlying linear algebra. We assume column vectors are being used.

The equation to register an unregistered sample is:

$$X_{reg} = (K \cdot C^{-1}) \cdot X_{unreg} \quad (1)$$

where K is a 4x4 translation matrix that transforms a point at the origin (0, 0, 0) to a known point, and C is the sample described above.

The derivation of equation (1) follows. We want to find the transformation M that transforms unregistered tracker samples X_{unreg} to registered tracker samples.

$$X_{reg} = M \cdot X_{unreg} \quad (2)$$

Let K be a 4x4 matrix describing the translation from the display's origin to the known point, and let C be a sample set to X_{unreg} at the moment the user posed the tracker at a known point and orientation. We can then substitute K for X_{reg} and C for X_{unreg} in equation (2) because we sampled the tracker data C at the moment the user posed the tracker at a known position K in display coordinates, with the tracker's axes aligned with the display's.

$$K = M \cdot C \quad (3)$$

The unknown M can be solved for by multiplying both sides of the equation on the right by C^{-1} .

$$K \cdot C^{-1} = M \cdot C \cdot C^{-1} \quad (4)$$

$$K \cdot C^{-1} = M \quad (5)$$

Substituting for M in equation (2) gives equation (1) above.

To provide a specific scenario, if our units are feet, the distance from the edge to the center of our tracker is a quarter inch (approximately 0.021 feet), and a one-foot ruler is used to find the point one foot away from the center of the display as shown in Figure 3, then K would have the value:

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (1 + 0.021) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

One extension we made to our application was writing out the registration matrices to a file in the local directory when they were obtained so that they could be read in on subsequent runs, thus, eliminating the need for re-registration. These cached values could be replaced with the results of another registration if the user desired.



Figure 4: The pose used for registering the tracked glasses is similar to that for basic registration. The point midway between the viewer's eyes is positioned in the same known point, and the glasses' lenses are oriented parallel to the display.

Tracked Glasses Registration

For stereo rendering it is necessary to know where the viewer's eyes are relative to the display. Given the interocular distance d , a matrix H that describes the point midway between a viewer eyes, and the viewer's head orientation that has an x -axis pointing from the left eye to the right eye, then eye_{left} and eye_{right} can be computed as follows:

$$eye_{left} = H \cdot \begin{bmatrix} -\frac{d}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

$$eye_{right} = H \cdot \begin{bmatrix} \frac{d}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (8)$$

Due to practical issues, a tracker is often mounted in a somewhat arbitrary location on the stereo viewing glasses. To achieve correct stereoscopic rendering, the offset and rotation of the tracker to the point between a viewer's eyes must be determined. A registration step similar to the one described above can easily determine the transformation from the head-tracker to the point between the viewer's eyes.

Similar to the basic registration process, if the tracked glasses are posed with the plane of its lenses parallel to the display and positioned such that the mid-point between the viewer's left and right eyes is at the known point as shown in Figure 4

and at that moment a registered sample D is recorded, then all the information needed to compute H as used in equations (7) and (8) has been obtained.

IMPLEMENTATION The equation to compute H is:

$$H = X_{reg} \cdot (R^{-1} \cdot (K \cdot T^{-1})) \quad (9)$$

where X_{reg} is a registered sample from the tracker mounted on the glasses, sample D from above is decomposed into the rotation matrix R and translation matrix T such that $D = T \cdot R$, and K is a translation matrix that transforms a point at the origin to the same known point described above.

The derivation of equation (9) is as follows. To compute H from the tracker data, we must find the vector v that translates a point from the tracker to the mid-eye point in the *tracker's* coordinate system. If v is written as the translation matrix V , then H could be written as:

$$H = X_{reg} \cdot V \quad (10)$$

Since 1) T is a translation matrix that transforms a point at the origin to the tracker's location relative to the display, and 2) K is a translation matrix that transforms the origin to the known point midway between the left and right eyes, the 4x4 translation matrix V that translates a point at the tracker's position to the known point is:

$$V = K \cdot T^{-1} \quad (11)$$

However, V is in the display's coordinate frame and does not account for the arbitrary mounting of the tracker on the glasses. If V is first transformed by a matrix E (which we define as the matrix that will compensate for the arbitrary mounting of the tracker on the glasses), then we will have the correct equation. Thus, we correct equation (10) by rewriting it as follows:

$$H = X_{reg} \cdot (E \cdot V) \quad (12)$$

Given the tracker's position when D was sampled and the value for V from equation (11), we can substitute values for all variables but E in equation (12) and then solve for E as follows:

$$K = D \cdot (E \cdot (K \cdot T^{-1})) \quad (13)$$

$$D^{-1} \cdot K = E \cdot (K \cdot T^{-1}) \quad (14)$$

$$D^{-1} \cdot K \cdot (K \cdot T^{-1})^{-1} = E \quad (15)$$

$$(T \cdot R)^{-1} \cdot K \cdot (K \cdot T^{-1})^{-1} = E \quad (16)$$

$$R^{-1} \cdot T^{-1} \cdot K \cdot T \cdot K^{-1} = E \quad (17)$$

Because translation matrices are commutative, we can rearrange the equation and simplify as follows:

$$R^{-1} \cdot (T^{-1} \cdot T) \cdot (K \cdot K^{-1}) = E \quad (18)$$

$$R^{-1} \cdot (I \cdot I) = E \quad (19)$$

$$R^{-1} = E \quad (20)$$

Substituting for E and V in equation (12) gives equation (9) above. Finally, note that in the case K is purely translational equation (9) simplifies to:

$$H = X_{reg} \cdot (D^{-1} \cdot K) \quad (21)$$

ERROR ANALYSIS

There are several sources of error in this system. First, the oriented tracker and tracked glasses are positioned at the known point by hand and thus are only as accurate as the person doing the positioning. We used a ruler to aid in positioning the devices because it was a readily available object to both our group and our collaborators off-site, although a specially designed prop probably would have worked better. We have not done rigorous studies of the impact of this error, but the report below provides anecdotal feedback on performing the registration and on using the results of the registration for five other tasks. Error may also come from dynamic or static error for which there exist a variety of techniques to compensate.

RESULTS AND DISCUSSION

We wrote an OpenGL GLUT program to implement and evaluate this approach to registration. We had eight people (students and staff) in our graphics lab perform the registration procedure and perform several tasks aimed at evaluating how well the registration approach met our goals of supporting head-tracked stereo viewing and 6-DOF direct manipulations.

The hardware configuration consisted of a Fishtank VR system. Specifically, an IBM P202 monitor, a Polhemus Fastrak magnetic tracker with two trackers (one mounted on the stereo glasses and one free tracker), an Intel-based two-processor machine (3 GHz each) running Redhat 9, a 3Dlabs Wildcat 6210 graphics card, and NuVision 60GX stereo glasses. UNC's VRPN reported raw tracker data to our software. Finally, the Fastrak was configured to operate at 38400 baud and in its "X+ hemisphere".

No attempt was made to calibrate the tracking system. We did not perform calibration for two reasons: 1) our working environment is dynamic and subject to changes, which we expect would require frequent re-calibration, and 2) so far, the tracking data appears to be reliable enough for our needs, except when our magnetic trackers are moved near the CRT monitor or beyond the 3-4 foot working volume of our Fishtank VR system. If tracker error is found to be conspicuously prohibitive in the future, we will apply a calibration step to

resolve the problem. For example, it appears Task 5 could have benefited from calibration.

The tracking system we used had two trackers: one was not attached to anything and one was mounted on the stereo glasses. This made it easier to collect the two samples C and D described above that are used for registration. If your tracking system has only one tracker, then the same tracker needs to be used for collecting both C and D . However, since the tracker can be mounted arbitrarily on the stereo glasses, it may be possible to mount it such that it need not be removed when collecting sample C . For example, notice that in Figure 4, the tracker could easily be oriented to assume the pose used in Figure 3 without removing it from the tracked glasses.

In general, because the two tracker poses used in the basic and tracked glasses registration steps are different, it is not possible to do tracked glasses registration without first doing the basic registration. The only case where it is possible is if the tracker for the glasses is mounted at the mid-eye point, in which case C and D would be the same. However, this situation isn't realistic with today's tracking technology because the tracker would have to be behind the bridge of the user's nose.

Our study consisted of 6 tasks targeted at testing the effectiveness of the registration technique and overall performance of the registered system. We supplied each user with a printed tutorial of the tasks to be performed, as well as verbal explanations and brief demonstrations. A trial consisted of performing the registration task (Task 1) and then evaluation tasks (Task 2-6). Each person performed the trial twice, but on separate days. Only one person had previous experience registering a Fishtank VR system.

We believe hardware limitations may have contributed to some user difficulties, but overall, the registration technique itself was successful. The one subject who had experience in registering a Fishtank VR system expressed interest in adopting our technique. The feedback for each of the tasks from the user study follows.

TASK 1

The first task was the registration of the tracking system and tracked glasses. We wanted to find out if users could perform it easily and also gather their feedback on the approach. Before starting, we asked users about their previous background with Fishtank VR and what methods they thought would be best to register the system. All but one of the users commented that they had not used Fishtank VR before, although many of them had seen it. Several users said that they would use a transformation from the tracker's base coordinates to the screen's coordinates in order to achieve registration. One user said to record the value of the tracker at different points on the screen, such as at the four corners, and using those values, transform reported coordinates into the tracker system. While most users responded with the same basic underlying

idea, few of them could explain a clear and easy procedure to correctly obtain this transformation.

Our registration consisted of holding the tracker at a single known position, one foot in front of the center of the screen, and pressing a key to begin the registration. After the key was pressed, four slow beeps were sounded, and on the fourth beep, the program used the current tracker position to calculate the correct registration transform. An analogous procedure was done for the registration of the glasses.

Most users did not have a problem with our registration procedure; they reported it was straightforward and easy to do, but they did have several comments about some of the inconveniences. A majority of the users thought the keyboard interface was not very intuitive, especially since two different keys were used to register each device. Many preferred that the spacebar be used for all the registrations, since it was the easiest key to press. The other significant problem reported was having to hold up a ruler and tracker after having had to press a key—users felt they needed “three hands”. They also wanted more audible and clearer beeps to signal the registration progress, and one user suggested having a visual countdown. To help align the tracker’s coordinate system with the display’s, a user suggested having physical marks on the tracker devices. In general, users did not like the idea of being timed to get the tracker positioned correctly. Although one user did not complain about the setup, several others did and pointed out that a stand or a type of fixture would have helped a lot; however, no user could describe how to easily build such a stand or fixture. An “L-ruler” from a hardware store, or a book with defined edges, were suggested to replace the ruler. Having a stand or fixture most likely would have helped with keeping the trackers more perpendicular to the screen.

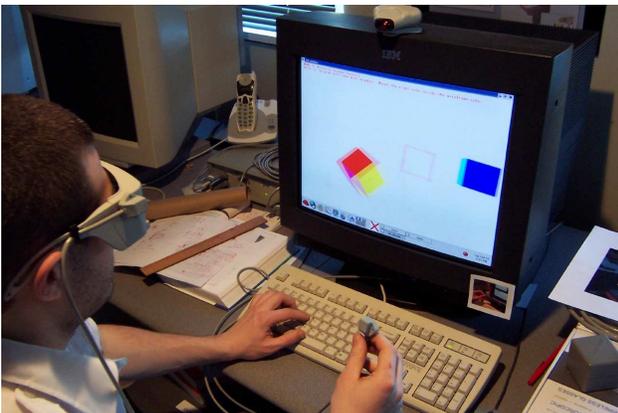


Figure 5: In Task 2, users tried to re-orient the left cube to match the right one.

We allowed users to perform this task as many times as necessary, until they were satisfied with the registration results. Many users said the task became easier with more practice. Multiple registrations tended to help achieve better results.

All users were able to do the registration quicker on their second trial. Once users understood the process, the registration steps took less than a minute to perform.

TASK 2

Task 2 involved moving a virtual cube to match a target cube’s randomly selected orientation. We wanted to find out if users could perceive any error in the registration while performing a direct-manipulation task. The virtual cube was manipulated by holding down the ‘g’ key while translating and rotating the free tracker. We provided a wireframe outline next to the target which users were asked to move their cube into. Both the user’s and the target cube had the same color scheme with no two sides sharing the same color.

Users responded very well to this task, which indicated that the registration was working well. All of them said the movement of the virtual cube was what they expected from rotating and translating the free tracker. They also said the task was relatively easy to accomplish. The images were clear and appeared well in stereo.

One of the troubles that users encountered involved situations where the user tried to rotate the cube and found that it resulted in rotating their hand into an awkward or uncomfortable position. Another inconvenience that several users mentioned was the cable on the tracker. Users believed that a wireless tracker or a spherically shaped tracker, that can rotate freely about the cable, would have helped. However, both of these are characteristics of the interaction technique and hardware, not the registration procedure.

Again, users commented that with more practice, the task became easier. After becoming familiar with the setup, some users were no longer bothered by the tracker’s cable and discovered how to avoid awkward orientations with it. One user commented that the control of the tracker was very similar to a mouse; when the user ran out of space to maneuver, the user let go of ‘g’, repositioned the tracker, and grabbed the cube again, much in the same manner as a user would reposition a mouse when there is no more space on a mousepad. This is the familiar “clutching” operation used in many VR techniques.

TASK 3

In this task, users were presented with a set of 8 squares on a horizontal line at the center of the screen. We wanted to test whether the registration produced a good stereo effect. Without moving their head, users were asked to determine how many different depths the squares lay on. The answer was always 1, 2, or 3. The actual depths varied by three-quarters of an inch, a small value we selected to make the task somewhat challenging. The nearest square was three-quarters of an inch behind the plane of the display. Each square was approximately 1 inch wide, but its scale was randomly altered slightly so that users could not easily use the size of the squares to determine depth. A horizontal gap of about a

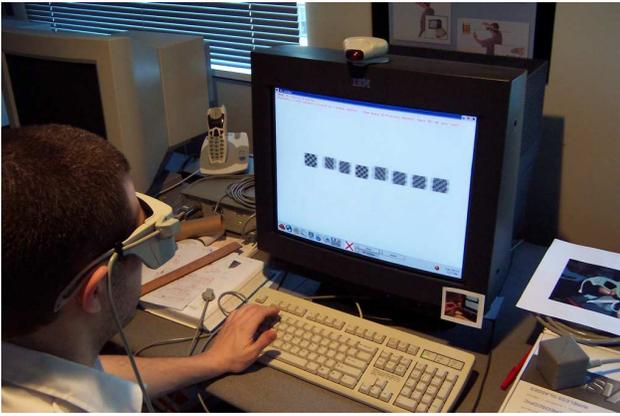


Figure 6: Task 3 consisted of determining how many different planes a set of squares lay on.

half inch existed between each square to prevent occlusion between squares. Initially we drew solid yellow squares on a white background. After preliminary runs indicated depths were difficult to judge, we changed the squares' color from solid yellow to a 5x5 grayscale checkerboard pattern. This texture pattern made the depth of the squares more apparent.

On average, users scored 80% correct. Most users found this task easy to perform. Each user was given about 5 or 6 sets of squares, and each user got at most 2 wrong. The initial accuracy of the users varied from person to person. Some users were better able to immediately perceive the different depths from the stereo alone while others required some more practice before they could distinguish between them. After making their initial guesses, users were then allowed to move their head. A lot of the users said that headtracking helped tremendously, although a few said it was not necessary. Several users said headtracking was a helpful method to *confirm* their initial guesses.

After being shown sets of squares having each of the possible depths, the users' performance improved. One user commented that this helped to understand the degree of depth differences that we were looking for.

TASK 4

The fourth task consisted of viewing a cube centered in the display and big enough that users could not see all its sides from any one vantage point. We wanted to test whether users could detect problems with the head-tracked stereo viewing. On the top, right, bottom, and left sides of the cube there were a random number of dots ranging from none to 10. Users were asked to count the number of dots per side and report any difficulties they had performing the task.

Headtracking behaved mostly as users expected. All users were able to see the dots clearly. Viewing the left and right sides were not problematic. Problems arose in viewing the top and bottom sides of the cube; viewing the bottom sometimes resulted in a loss of stereo image, due to the tracker go-

ing out of range, and viewing the top required moving a lot, such that the user was almost standing up. The loss of stereo appears to be a limitation of the tracking hardware rather than our registration technique.

Forcing the user to make exaggerated movements to see the cube's sides was intentional; we wanted to require users to make use of the headtracking to accomplish the goal of this task. The unfortunate side effect of making such large movements is the incorrect reporting of the trackers' positions due to limitations in the tracker technology. Because registration is a linear transformation, shearing and small cube translation errors, reported as a function of head position, can most likely be attributed to tracker calibration.

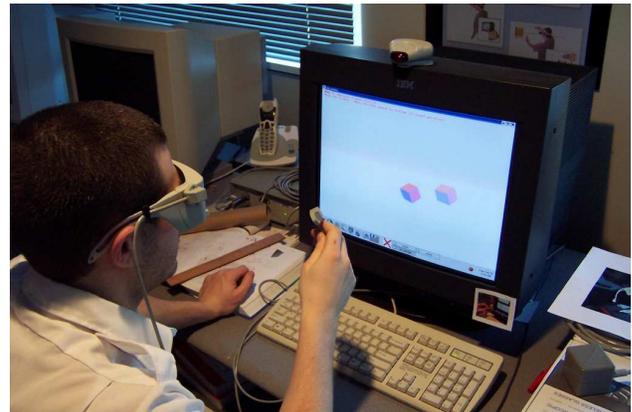


Figure 7: Task 5 displayed a small virtual cube at the position reported back by the tracker.

TASK 5

In the fifth task, a cube just smaller than the tracker was drawn wherever the system reported the tracker was positioned. This was our most rigorous test of system accuracy. If there was no error in the system, then the user would have never been able to see the cube or might have seen it briefly during fast movements, depending on the amount of lag in the system. In an errorless system, the tracker itself would occlude the left and right eye's rendering of the cube projected onto the display screen.

However, this was the least successful task of the study. Instead of drawing the cube at the tracker's position, the cube tended to appear to users slightly above and to the left of the tracker. The reported error distances, all within about 1", varied depending on the actual tracker position. Not only did users experience problems with the cube's position, they also reported severe double-imaging. While double-imaging did stop when the tracker was within about 4"-6" of the screen, magnetic interference caused the cube to start jittering a lot. Otherwise, translations and rotations worked as users expected.

The inconsistencies in error probably span from an incorrect transformation in drawing the cube. Task 2 performed much better, and the only difference between Task 2 and this task is the use of relative positioning as opposed to absolute po-

sitioning. We also believe that calibration could have significantly improved results.

TASK 6

In this task, a cube was shown at random depths ranging from 4 feet past the screen to 2 feet in front of the screen. We wanted to try to quantify the z -values that users could fuse 3D objects at when our registration was used. We also randomized cube and background color because as we used the system we noticed color choices seemed correlated with our ability to view objects at different distances. Cube red, green, and blue color components were randomly set to either 0% or 100% intensity (but the color was never set to black or white) and the background was randomly set to 0%, 33%, 66% and 100% gray intensity.

Users were asked to report whether the cube image appeared “good,” in that it converged and produced a stereo image with possibly slight ghosting, or “bad,” in that the cube was uncomfortable to look at or distinct double-imaging occurred. Each user judged about 30 cubes.

In summary, the best results were seen when red cubes were shown on a black or white background at a depth range of 2 feet behind to 1 foot in front of the display. Yellow cubes did as well as red cubes on lighter colored backgrounds. Yellow on backgrounds darker than 33% could only be seen at best from the display surface to 6 inches in front of it. Cubes colored with 0% red, 100% green, and 100% blue, as well as pure green cubes, on a black background did the worst.

FUTURE WORK

The presented registration technique is useful for quickly getting started with a fishtank VR system. The ideas are theoretically readily applicable to larger display systems. For example, the same technique and parameters used on our fishtank VR system could be used on a powerwall. However, if the known point K isn't easy to position a tracker at (e.g., it must be far from a display surface because of high error close to the display surface), then human error may be amplified.

We have two ideas for increasing the accuracy of the system at the cost of some additional work. First, one could build a specialized apparatus that can more accurately position a tracker and glasses at a known point relative to the display. Although we found the ruler worked well for positioning objects at a fixed distance perpendicular to the display, it seems that a simple and more effective prop could have been designed and manufactured to replace the ruler used in our registration technique. Such a prop should make it easy to position an object a fixed distance perpendicular to a point on the display and have rigid mounting points for both a free tracker as well as the tracked glasses. This would help with simplifying the process further and improving accuracy. Second, multiple known points could be sampled instead of just one. Registration results for each of these locations could be averaged to produce a more accurate registration.

Finally, while Task 6 shows that users can successfully view cubes at different depths with our registration technique, it also suggests color choices are strongly connected to depth perception in IVR visualizations and should be further investigated.

CONCLUSIONS

We presented a low-cost technique for registering a tracking system with a display. We demonstrated that it could be performed in under a minute by people who have never registered a 6-DOF tracker with a display before. Through a series of experiments run on a Fishtank VR system, we also demonstrated that it performed fairly well for absolute manipulations and that it worked very well for tasks requiring head-tracked stereo and relative manipulations.

ACKNOWLEDGEMENTS

This work was supported by LLNL Research Subcontract No. B527302 and NSF (CCR-0086065). This work used UNC's VRPN library, which is supported by the NIH National Research Resource in Molecular Graphics and Microscopy at the University of North Carolina at Chapel Hill. We would also like to acknowledge Cullen Jackson's and Juergen Schulze's help in discussing user study design and results.

REFERENCES

1. P. Auer and M. K. Warmuth. Tracking in a multi-user augmented reality system. *Machine Learning*, 32(2):127–150, 1998.
2. A. Forsberg, J. LaViola, and R. Zeleznik. Ergodesk: A framework for two and three dimensional interaction at the activedesk. In *Proceedings of Immersive Projection Technology Workshop*, 1998.
3. A. L. Fuhrmann, R. Splechna, and J. Prikryl. Comprehensive calibration and registration procedures for augmented reality. In *Proceedings of Eurographics Workshop on Virtual Environments*, pages 219–228, 2001.
4. M. Ikits, C. D. Hansen, and C. R. Johnson. A comprehensive calibration and registration procedure for the visual haptic workbench. In *Proceedings of the workshop on Virtual environments 2003*, pages 247–254. ACM Press, 2003.
5. M. Livingston and A. Slate. Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments*, 6(5), 1997.
6. D. Pape and D. Sandin. Quality evaluation of projection-based vr displays. In *Proceedings of Immersive Projection Technology Workshop*, 2000.
7. 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. *Computer Graphics*, 30(Annual Conference Series):429–438, 1996.