Vive Tracking Alignment and Correction Made Easy

Alex Peer* Peter Ullich Kevin Ponto†

Virtual Environments, Wisconsin Institute for Discovery
University of Wisconsin-Madison, USA

ABSTRACT

The alignment of virtual and real coordinate spaces is a general problem in virtual reality research, as misalignments may influence experiments that depend on correct representation or registration of objects in space. This work proposes an automated alignment and correction for the HTC Vive tracking system by using three Vive Trackers arranged to describe the desired axis of origin in the real space. The proposed technique should facilitate the alignment of real and virtual scenes, and automatic correction of a source of error in the Vive tracking system shown to cause misalignments on the order of tens of centimeters. An initial proof-of-concept simulation on recorded data demonstrates a significant reduction of error.

Index Terms: I.3.7 [Computing Methodologies]: Graphics Utilities—Virtual reality

1 INTRODUCTION

Alignment of real and virtual space is often desirable in virtual reality. At a basic level, tracking user movement requires correct recognition of the relative position of users over time, and correctly positioning virtual cameras requires knowing the height of a user's eyes. Some scenarios may involve interacting with physical objects while viewing the virtual world, and so require reproducing them in the virtual space. The virtual space described by tracking systems are arbitrarily aligned to real space, as determined by placement of tracking hardware and initial calibrations. An alignment between the real and virtual space can be established through exacting measurements, which is practical for long-term static installations in a single location; recent consumer grade tracking systems, however, lend themselves to quick deployment in new spaces and dynamic temporary arrangements. A quick method of alignment is needed to work alongside these quick deployments and rearrangements. These alignments are generally a translation in 3-space and a rotation around the y or "up" axis, such that the real and virtual floor planes' positions and orientations match.

The tracking system of the HTC Vive has also been shown to display systemic errors [1], most notably an error in tracked object height that can be described as a rotation around the x and z axes of the tracked space. This misalignment appears to change when tracking of the headset is temporarily lost, and when the Vive software is initialized. The degree and direction of misalignment as yet appears to be random, but may be on the order of tens of centimeters [1]. Niehorster et al. [1] conclude that this error renders the Vive unusable for experiments when loss of tracking is a possibility. They show a correction is possible by individually measuring the tracked position of 36 points on a 9 x 4 grid, then finding a transformation that minimizes the root mean square error between the measured and ideal grids. They conclude this is an impractical method to employ mid-experiment, which is reasonable; however, we suggest that practical methods of correction do exist.

We propose an automated method using only three tracked points, which aligns the virtual and physical spaces, accounts for the errors seen by [1], and is practical for use during experiments. A plane can be derived from three points. If three tracked points are arranged on a known plane in the physical space, deviations from their physical positioning in the virtual tracked space can be taken as error in describing this plane; this can be used to derive an alignment between tracked and real space (a translation and a rotation around y), and will also correct for the "tilt" errors in height seen by [1] (rotations around x and z). Here, an implementation of such a system using Vive Trackers is proposed, which would facilitate continuous, automated alignment; we explore its practicality by simulating corrections on previously measured data.

2 METHOD

2.1 Materials and Apparatus

A 3m x 3m grid was marked on the floor at 0.5m intervals, yielding a 7 x 7 grid of 49 points. Grid points were marked by black lines on pieces of white tape, and a 4m x 4m square around the grid was marked in white tape. An HTC Vive purchased in 2016 was used. The Vive's lighthouses were mounted on tripods at a height of 2.4 meters from the floor, and placed at opposing corners of the 4m square around the grid. The Vive headset was located on a table just outside of the grid. The Unity game engine and SteamVR library were used to acquire tracking information from the Vive system, using a coordinate system where negative z is forward and y is up. Three Vive Trackers were affixed to PVC pipe using 3D printed fittings, arranged to form two perpendicular 0.5m vectors on the ground plane. This apparatus can be seen in Figure 1.

2.2 Measurement

The tracked location of individual grid points was measured by individually placing the central Vive tracker at each point, with the other trackers positioned at neighboring grid points to assist in alignment. The tracked position of the Vive tracker was sampled for three seconds, and the average position was taken as the measured position at that grid point. The tracker was then moved to another point, and another 3 second measurement was taken; this process was repeated for all grid points.

2.3 Correction

Though our eventual goal is real-time automated correction using the Vive Tracker apparatus herein described, this work approximates its function using the measured location at grid points (0,0,0), (0.5,0,0), and (0,0,0.5). It should be noted that these measurements all come from the same Vive Tracker, and may lack some intra-device variance; they also may be less well aligned than three different pucks rigidly affixed to enforce a right angle as depicted in Figure 1. However, they serve well as a proof-of-concept.

Using these points, the necessary alignment transformation can be derived. Given our three points marking the intended origin, x-axis, and z-axis:

$$P^o = grid[0,0,0]$$
$$P^x = grid[0.5,0,0]$$
$$P^z = grid[0,0,0.5]$$

---

*e-mail: alex.peer@wisc.edu
†e-mail: kbponto@wisc.edu

We can derive two vectors describing the directions of the axes:

$$\vec{x} = P^x - P^o$$
$$\vec{z} = P^z - P^o$$

And normalize both vectors, to yield $\hat{x}$, $\hat{z}$:

$$\hat{\imath} = \frac{\vec{i}}{||\vec{i}||}$$

The cross product of these vectors yields a unit vector on the intended y-axis:

$$\hat{y} = \vec{z} \times \vec{x}$$

And these three vectors directly describe a 4x4 rotation matrix needed to align the virtual and real spaces:

$$M_{rot} = \begin{pmatrix} \hat{x}_x & \hat{x}_y & \hat{x}_z & 0 \\ \hat{y}_x & \hat{y}_y & \hat{y}_z & 0 \\ \hat{z}_x & \hat{z}_y & \hat{z}_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We also need a translation to align the scene to the origin:

$$M_{pos} = \begin{pmatrix} 1 & 0 & 0 & -P^o_x \\ 0 & 1 & 0 & -P^o_y \\ 0 & 0 & 1 & -P^o_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

And composing these transforms through matrix multiplication yields our final alignment transformation:

$$A = M_{rot} M_{pos}$$

Multiplying any tracked space position by the alignment matrix A will align it to the real space.

2.4 Results

Measurements were taken of the tracked position of 49 points on a 7 x 7 grid, with grid points at 0.5m intervals. An alignment matrix was derived from the measures taken in one corner of the grid, as if our apparatus had been placed there (as discussed in §2.3). A partial alignment matrix was also derived, using only $\hat{x}$ and P^o to describe a 2D translation and rotation about the y-axis. The partial alignment allows the real and virtual coordinates to align for the sake of generating figures, without correcting for ”tilt” errors in height. Table 1 and Figures 3 and 2 show the results of these measures, with the *uncorrected* entries having the partial alignment matrix applied, and the *corrected* entries having the full alignment matrix applied. Improvements between *uncorrected* and *corrected* cases are due to the elimination of the errors described in [1]. The corrected case is more level, as seen in the more uniform coloring of Figure 3. As seen in Table 1, the standard deviation, range, and total error in height are all reduced by an order of magnitude. Some error still remains; the range of error is perhaps most descriptive, as it shows participants in the corrected case would still experience almost 3cm of difference in, say, eye height when walking from one end of the tracked space to the other. This is an improvement over the 20cm in the uncorrected case, but may still influence experiments that require extremely accurate representation of space.

Table 1: Summary of variation in tracked object height, in meters.

	Mean	SD	Range	Total Error
Uncorrected	0.075	0.061	0.214	3.91
Corrected	-0.006	0.007	0.028	0.38

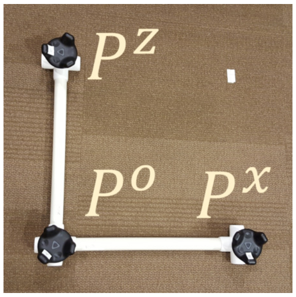

Figure 1: The proposed Vive Tracker apparatus.

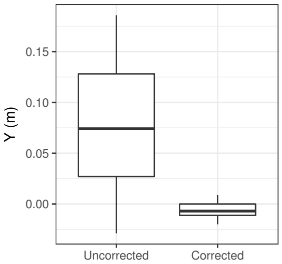

Figure 2: Difference in tracked object heights at the same physical height. Deviation from 0 is error.

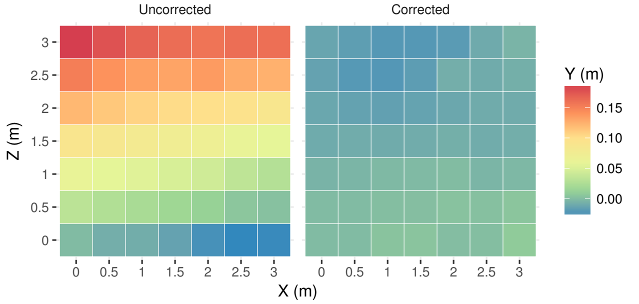

Figure 3: Error in tracked object height at various grid points in uncorrected and corrected cases.

3 CONCLUSION

This work proposes a method using 3 Vive Trackers to address the general need for real and virtual space alignment, and account for the tilt errors described by [1]; a proof-of-concept simulation shows that this method largely accounts for the observed error. Some error remained, ranging over 2cm; this may be unavoidable noise in the Vive tracking system, but it may also be that our three points didn’t fully characterize the error, and additional Trackers distributed more widely through the space might serve better.

It should be noted that the nature of the tilt error is still underexplored. For instance, it appears to have more impact on height (y) than on ground plane position (x and z); separate corrections may be needed for height and ground plane position. It is also unclear how the error affects users; anecdotal evidence suggests it goes largely unobserved, which is surprising. The degree of error is also highly variable; the uncorrected case presented here was chosen from the several recorded due to showing enough error that the correction was obvious. This amount of error is not atypical, but also not constant; the error seems to change every time the Vive headset acquires tracking. More work is needed to quantify the expected behavior of this error, as well as how it is experienced by users.

REFERENCES

[1] D. C. Niehorster, L. Li, and M. Lappe. The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception*, 8(3):2041669517708205, 2017.