

Comprehensive Calibration and Registration Procedures for Augmented Reality

A.L. Fuhrmann, R. Splechtma and J. Příkryl

VRVis Competence Center for Virtual Reality and Visualization,
Lothringerstraße 16/4, A-1030 Vienna, Austria
e-mail: [fuhrmann|splechtma|prikryl]@vrvis.at

Abstract: Augmented Reality - best described as adding computer-generated virtual content to the real environment – needs more adjustments to work properly than immersive virtual environments. To be perceived as an augmentation of reality, the virtual environment has to be properly aligned to the real world. This *registration* process has to be done at least once for every hardware set-up, but may have to be repeated in part or completely for each user, prop or device to be included both in the real and the virtual world. In this paper, we propose a comprehensive process for registration and calibration tasks necessary to implement correct augmentation. This includes procedures for calibrating projective and head-mounted displays, tracking systems, tracked input devices and props. Our method unifies the necessary tasks of world-to-augmentation alignment, display calibration and registration of tracked and static props in one, interactive set-up process, which can easily be conducted by the untrained user.

1 Motivation

The *Studierstube* augmented reality system [1] uses different techniques to overlay computer graphics onto a user's view of the real world. Primarily we have been using see-through head-mounted displays to accommodate individual viewpoints for a multi-user scenario, but *Studierstube* also supports projection display devices such as the Virtual Table or a stereo projection wall. Our concept includes at least two tracked interaction devices per user: a pen and pad combination called the *Personal Interaction Panel* (PIP) and supports tracking technologies ranging from magnetical to optical and inertial methods.

Since each hardware set-up can consist of almost any combination of the display and tracking technologies mentioned above, most of the parameters of display and tracking system will depend on the set-up. Altering the hardware or setting an environment up in a new location will require a complete registration process.

Studierstube is already in use as development environment in different research institutes in Europe and the United States, but not two of them are employing exactly the same display/tracker/input device combination. In most cases the calibration and registration process proved to be the biggest obstacle when setting up a new site. Even when using the same hardware in a different location – e.g. as demonstration at a scientific conference or a trade show – altered position of tracker origin and projector lead to unnecessary delays in the set-up.

The aim of the described method is to provide a fast, comprehensive calibration and registration procedure. The method should present itself to the user as an interactive sequence of simple, independent tasks, which can easily be performed even without knowledge of the actual parameters they modify. Setting up an augmented reality installation becomes therefore a reliable and reproducible routine.

Fig. 1 illustrates the complexity of this task. While by itself each of the shown transformations is easily grasped, their concatenation leads to non-intuitive results. The determination of all these transformations by hand, even when excluding projective distortion of the display is an error-prone and cumbersome task. Discrepancies between the movement of the stylus and its virtual counterpart for example can be caused by the world-to-tracker transformation (w_t), the world-to-screen transformation (w_s), or the tracker-to-eye transformation (t_e). Finding out which part of the registration is to blame can be difficult, especially when the misregistration results in a completely blank screen.

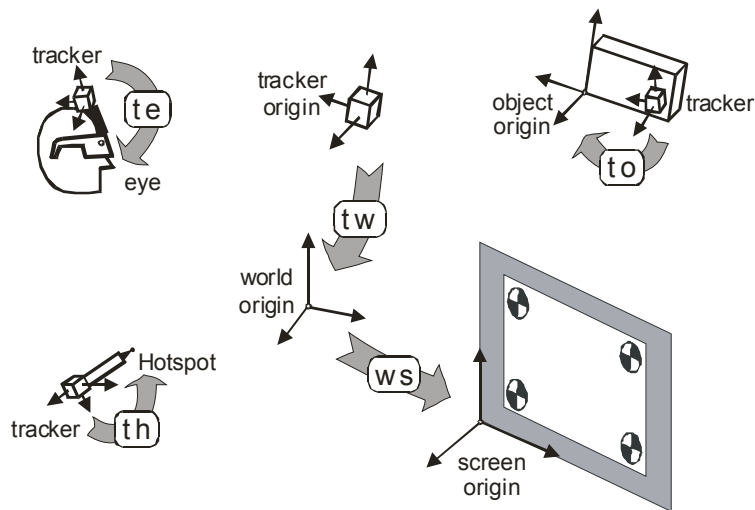


Fig. 1: registration parameters

2 Related Work

Some of the here-discussed problems have already been addressed separately:

Holloway [2] analyzed different aspects of registration error and Hoff and Vincent [3] presented analysis of head pose accuracy for AR applications.

Bajura [4] proposed calibration for video see-through systems based on tracking known features in the working environment. Interactive calibration methods for see-through HMDs have been described by McGarrity and Tuceryan [5], Fuhrmann et al. [6], and Tuceryan and Navab [7]. Czernuszenko et al. [8] describe a tracker and

camera calibration approach for back-projection systems. Summers et al. [9] address registration and calibration of an experimental see-through projection-based system that employs shutter glasses.

Whitaker et al. [10] describe a pointer calibration method together with two different object calibration methods for augmented reality. Automatic, image-based object identification and registration methods have been proposed by Billinghurst [11] and Rekimoto [12]. Image-based methods have the potential to work completely without user intervention, but imply a video-based AR set-up, which until now has not been integrated in Studierstube.

The innovative aspect of our method lies in its unification of separate but connected problems: calibrating the HMD or shutter glasses (**te**) or projection screen for example, works only when the stylus' hotspot has been registered (**th**), but is independent of the world-to-tracker transformation (**wt**). The sequential execution of the different steps of our method takes these interdependencies into account.

A further advantage of our method lies in its simplicity as perceived by the user: every sub-task consists of an easily performed gesture, or the touching of a displayed marker. Additionally all of the used algorithms have fail-safe mechanisms: if the user wrongly positions his head during display calibration, it is detected via the head-tracker and the step is repeated; if the fitting algorithm when registering a prop or the stylus shows discrepancies, a notification message is displayed.

3 Method Description

Our method implies two prerequisites:

First, a tracking system capable of delivering accurate results over the working volume. In the case of a magnetical tracking system, for example, one has to assure linear behavior by appropriate methods like Kindratenkos [13] or Livingston et.al. [14].

Second, a tracked pointing device with one button and precisely defined hotspot (i.e. a point that defines where an action is executed when pressing the button).

We divide the registration process into four distinct tasks:

- Calibration of the registration stylus' hotspot (**th**)
- Registration of tracker coordinates to the world coordinate system resulting in transformation (**wt**)
- Calibration of display, providing us with the proper orientation of the screen coordinate system (**ws**), linear image distortions (e.g. caused by off-center projection), correct eye distance for stereo, and — in case of a head tracked system — the tracker-to-eye sensor offset (**te**).
- Registration of the tracked interaction devices relative to their tracking sensors (**to**) and position of the static props – e.g. desks, walls – in world coordinates

4 Stylus calibration

Stylus calibration determines the offset of the tracker sensor as measured to its tip or hotspot (th). This has to be done at the start of the whole process if the tracking system does not provide an implicitly calibrated stylus, e.g. a digitizing wand.

We can accomplish this quite easily by fixing the stylus' tip in a small pit drilled into a table (Fig. 2), and moving the tracker sensor on a hemisphere measuring its position. Having fitted a sphere to the measured sensor positions, the center of the sphere gives us the position of the hotspot with respect to the sensor.

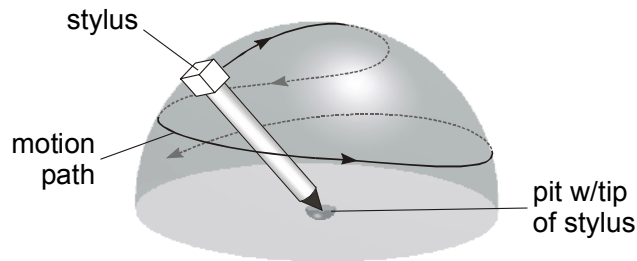


Fig. 2: calibrating the stylus

For fitting an optimal sphere to the measured data points, we optimize the offset vector from tracker sensor to the stylus' hotspot (th , see Fig. 1) using the variance V of the hotspot positions for the measured data points as a metric:

$$V = \sum_n (th_n - th_{mean})^2$$

This gives us a least-squares fit solution for the hotspot offset th . To enforce stability of the solution, the user has to cover a large part of the hemisphere, ideally by sweeping the sensor along two orthogonal great-arcs.

5 Registration of World to Virtual Coordinate System

The first step in the proposed calibration method is registration of the tracker system so that it corresponds to the selected world coordinate system, i.e. the transformation tw .

The registration process consists of three steps (Fig. 3):

First, the world coordinate system origin is determined by placing the stylus' hotspot to the desired location of the origin and pressing the stylus' button. Knowing the origin of the coordinate system, the up-vector can be defined by lifting the pen with pressed button along a vertical axis.

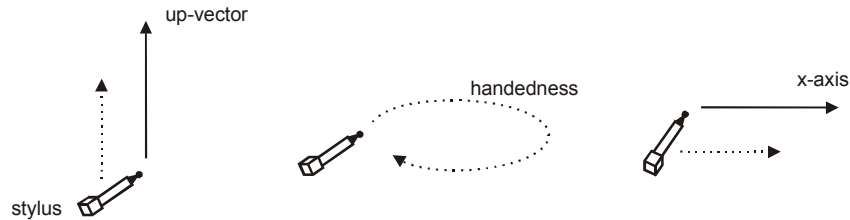


Fig. 3: registering world-to-tracker coordinate transformation

If an axis of the tracking coordinate system lies reasonably near this direction, the user will be asked if the up-vector should be changed to this axis on the assumption that the tracking system has been intentionally aligned. If this is not the case, the manually specified direction is used.

In order to determine whether the coordinate system is left- or right-handed we ask the user to circumscribe a clockwise-oriented circle in a horizontal plane of the world coordinate system.

Then the user is asked to move the pen horizontally to the right to determine the last remaining degree of freedom, the rotation around the up-vector. This step can be skipped for projection-based displays, where an alignment to the projection screen is mostly preferred.

6 Display device calibration

As already mentioned, the *Studierstube* system permits the use of see-through head-mounted displays (HMDs) or different kinds of front or back projection displays. Although both setups are to some extent similar, we shall address the calibration processes for projection and HMD setup separately.

6.1 Calibrating projection systems

Here we use a variation of the calibration method we applied to HMDs in [6], which differs from Czernuszenkos [8] method in the use of the stylus instead of a transparent reference grid, which would have to be built for each site.

First the position and orientation of the display screen has to be determined (transformation \mathbf{ts} , Fig. 1). This is done by projecting reference markers near the corners of the screen and measuring their position with the tracked stylus. This gives us the absolute position and orientation of the projection plane in tracker coordinates (Fig. 1, right).

To determine interpupillary distance and eye to tracker offset we project two or more markers — of which we now know the absolute location in tracker coordinates — on the screen. The user now brings the tip of the stylus in alignment with each marker (Figure 4) and presses the button time accordingly. This is repeated for each eye and gives us their absolute positions in tracker coordinates.

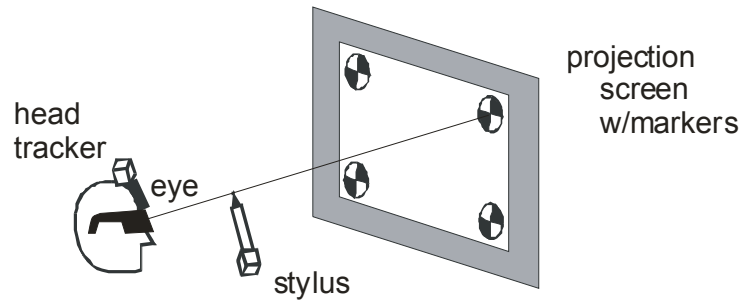


Figure 4: calibrating a projection set-up

For each measurement we now have:

- head position
- stylus' hotspot position
- marker position

Hotspot and marker define a line in space, which we transform into head-relative coordinates relative to the actual head position at sample time. By this we remove the influence of the users head position. The gathered lines can now be used to determine the users eyes' position relative to the head-tracker (t_e in Fig. 1).

Since in most cases the lines will not precisely intersect, we use an optimization procedure as we used in [6], resulting in an eye position with minimal distance to all measured lines.

Two criteria for positioning the stylus during data acquisition have to be observed by the user:

- the stylus should be much nearer to the users eye than to the marker on the screen
- the users head should always point in the general direction of the screen center (not in the direction of the active marker)

These rules guarantee a valid sample of measurements. We are going to implement some warning messages to notify the user of discrepancies.



Fig. 5: Calibrating the Virtual Table: projection plan calibration (left), eye position calibration (right). For illustration purposes, the displayed marker has been enhanced.

Figure 5 depicts the two typical user gestures during calibration: touching the marker on the screen during projection plane calibration and sighting along the stylus' tip during eye position calibration.

6.2 Calibrating head-mounted displays

For the calibration of the head-mounted display we use the method developed by Fuhrmann et al. [6]. This method guides the user through a sequence of tasks the same way that the whole registration procedure does.

Generally, this HMD calibration consists of displaying points in corners of each HMD display and aligning these points with the hotspot of the stylus. The measured data gives us all the necessary calibration information - eye position, view-plane distance, and aspect ratio. Redundancy in the measurements allows us to detect user errors and repeat steps accordingly.

7 Calibration of Props

Stationary props — e.g. desks, walls — have to be included in the registration process by specifying features on the objects, which can be used as calibration points. These features can be corners or attached markers, which can easily and unambiguously be touched with the stylus. The positions of these features have to be described in the virtual representation of the prop, allowing for inferring the position of the prop in the real world by an appropriate fitting algorithm.

The registration of tracked props — e.g. a pad, or a mock-up of an object to be augmented — is performed accordingly; only the measurements are taken in the objects tracker coordinate system instead of the absolute world coordinate system.

8 Conclusions and Future Work

We have presented a comprehensive method for the usual calibration and registration procedures necessary for using an augmented environment. The method has been implemented as an applet in our *Studierstube* system, allowing inexperienced users to properly set up their working environment. The only information that has to be specified when starting the calibration applet is the setup type (HMD or projection). The applet then guides the user through the calibration and registration process and produces all necessary XML files for our unified tracker interface, *OpenTracker* [15]).

As test setup we used our virtual table (BARCO BARON, Fig. 5), a back-projection desk specifically designed for VR applications.

The precision of our method depends heavily on the input data:

We tested it – as worst-case scenario - with an uncompensated magnetical tracker, which delivers extremely bad precision near our projection setup. The resulting errors amounted up to a few centimeters, essentially making the results unusable. By using our optical tracking equipment [16], which delivers results with about 3mm precision, eye positions were calculated within the same precision range (2-6mm).

A further contributing factor is of course the precision of the users input samples. We are at the moment evaluating different marker combinations and filtering techniques to minimize this error component.

Java3D [17], a high level API for writing three-dimensional graphics applications and 3D applets in Java, supports user interaction with the virtual world for different types of application setups through its unified view model. Since this API supports standard computer displays, (multiple-) projection display rooms, and head-mounted displays, as well as interaction using 6DOF trackers and joysticks, it can be applied for augmented reality applications needing registration. We plan to adapt the output of our applet to generate the files defining the Java3D view branch, which declares display method and tracking of camera and interaction devices.

Acknowledgments

Special thanks to Dieter Schmalstieg for his constructive criticism and help.

References

-
- [1] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. M. Encarnação, Michael Gervautz, Werner Purgathofer: The Studierstube Augmented Reality Project. Technical report TR-186-2-00-22, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, December 2000..
 - [2] Richard L. Holloway: Registration Error Analysis for Augmented Reality. *Presence*, vol. 6, no. 4, August 1997, pp. 413–432.
 - [3] William A. Hoff, Tyrone Vincent: Analysis of Head Pose Accuracy in

-
- Augmented Reality. To appear in IEEE Transactions on Computer Graphics and Visualization, vol. 6, no. 4, 2000.
- [4] Michael A. Bajura: Merging Real and Virtual Environments with Video See-Through Head-Mounted Displays. Dissertation, UNC, 1997.
 - [5] Erin McGarrity, Mihran Tuceryan: A Method for Calibrating See-through Head-mounted Displays for AR. Proceedings of the 2nd IEEE International Workshop on Augmented Reality (IWAR 99), San Francisco, CA, October 1999.
 - [6] Anton Fuhrmann, Dieter Schmalstieg, Werner Purgathofer: Practical Calibration Procedures for Augmented Reality. Proceedings of the 6th Eurographics Workshop on Virtual Environments, Amsterdam, Netherlands, June 2000.
 - [7] Mihran Tuceryan, Nassir Navab: Single point active alignment method (SPAAM) for optical see-through HMD calibration for AR. Proceedings of the IEEE and ACM International Symposium on Augmented Reality, Munich, Germany, October, 2000, pp. 149–158.
 - [8] Marek Czernuszenko, Daniel Sandin, Thomas DeFanti: Line of Sight Method for Tracker Calibration in Projection-Based VR. Proceedings of the 2nd International Immersive Projection Technology Workshop, Iowa State University, May, 1998.
 - [9] Valerie A. Summers, Kellogg S. Booth, Tom Calvert, Evan Graham, Christine L. MacKenzie: Calibration For Augmented Reality Experimental Testbeds. Proceedings of the 1999 symposium on Interactive 3D graphics, Atlanta, GA, April 1999, pp. 155–162.
 - [10] Ross T. Whitaker, Chris Crampton, David E. Breen, Mihran Tuceryan, Eric Rose: Object Calibration for Augmented Reality. Proceedings of Eurographics'95, Maastricht, Netherlands, 1995, pp. 15–28.
 - [11] Mark Billinghurst, Hirokazu Kato: Collaborative Mixed Reality. Proceedings of the First International Symposium on Mixed Reality, Yokohama, Japan, March 1999, pp. 261–284.
 - [12] Jun Rekimoto: Matrix: A Realtime Object Identification and Registration Method for Augmented Reality. Proceedings of Asia Pacific Computer Human Interaction 1998, Japan, 1998.
 - [13] Volodymyr Kindratenko: Calibration of electromagnetic tracking devices. Virtual Reality: Research, Development, and Applications (The VRS Journal), vol. 4, 1999, pp. 139–150.
 - [14] Mark Livingston, Andrei State: Magnetic Tracker Calibration for Improved Augmented Reality Registration. Presence, vol. 6, 1997, pp. 532–546.
 - [15] Gerhard Reitmayr, Dieter Schmalstieg: OpenTracker — An Open Software Architecture for Reconfigurable Tracking based on XML. Poster, IEEE Virtual Reality 2001, Yokohama, Japan, March 2001. Extended version available as technical report TR-186-2-00-18, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, June 2000.
 - [16] M. Ribo, A. Pinz, A.L. Fuhrmann: A New Optical Tracking System for Virtual and Augmented Reality. To appear in proceedings IEEE IMTC®/2001 (May 21-23, 2001, Hungary)
 - [17] Michael F. Deering, Henry A. Sowizral: Java3D Specification, Version 1.0. Technical Report, Sun Microsystems, August 1997.