Practical Calibration Procedures for Augmented Reality

Anton Fuhrmann, Dieter Schmalstieg, Werner Purgathofer

Vienna University of Technology,
Karlsplatz 13/186, A-1040 Vienna, Austria
{fuhrmann, schmalstieg, purgathofer}@cg.tuwien.ac.at

Abstract. Augmented Reality overlays computer generated images over the real world. This requires precise knowledge of the viewing projection of the head-mounted display (HMD) and its position. Most of the previously published methods are complicated or use special equipment for the calibration process. We present a collection of calibration methods usable for fast and easy calibration of camera parameters, object/camera to tracker transformations, and image rectification, which do not need additional instrumentation or complicated procedures. They are applicable for both see-through and video-based HMDs.

1 INTRODUCTION

Virtual Environments (VEs) present to the user computer generated images in the same way a real environment would be perceived: each eye perceives an image depending on its position and the direction of view. These images are generated according to the users heads position in space. The position is tracked by special sensors and used to move the virtual cameras inside the VE accordingly.

Slight errors in this process - introduced by erroneous measurements or wrongly defined camera parameters - result in a wrong perspective or parallax, giving a false apparent position of the virtual objects.

In an immersive setup these effects can in many cases be overcome. For example the hand-to-eye coordination of the user adapts to the altered visual feedback given by the virtual image of the hand.

In *Augmented Reality* [6], where computer generated images overlay the users view of the real surroundings, such distortions cannot be tolerated. When a user perceives two different loci of interaction, one given by the real image of his hand and one by the virtual image on a different position, the perceived clues conflict and hand-to-eye coordination is severely impaired.

In Figure 1 the *Personal Interaction Panel* (PIP) [14] is shown, a simple tracked board, which is augmented to act as an instrument panel for controlling the parameters of a scientific visualization. Figure 1a shows correct overlaid computer graphics, Figure 1b a slight misalignment as described above. Moving a virtual slider while seeing the real and virtual pen in different places is irritating and leads to problems when interacting with the virtual input elements.

Several approaches have been proposed in the past to attack the problems of calibration and registration. However, they are generally not designed to be operated by untrained users. This is a fundamental problem as every user is different, and consequently some per-user setup is inevitable for high-resolution AR. However, a complex or cumbersome process will lack user acceptance and will mostly be ignored.

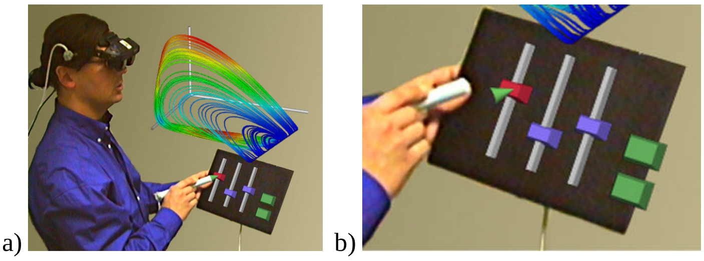

Fig. 1. Personal Interaction Panel: a) calibrated; b) miscalibrated

When augmented reality is used outside of research labs, procedures must be simple and may not require trained operator intervention. In this paper, we present a simple, stable and interactive method to implement camera calibration. The method requires little effort on the side of the user and should thus be more appealing to novice users and experts in proliferated lab practice alike. It is usable for see-though HMDs, does not require additional hardware, and is numerically stable. Furthermore a fast solution for the correct alignment of real and virtual objects is presented.

2 Registration in Augmented Reality

The alignment of virtual objects and their associated real counterparts is called registration. The desired result is correct registration of the images of the real and virtual objects. To achieve this alignment, a couple of different criteria have to be fulfilled.

- **Correct Sensor Data:** Tracking data - especially data acquired by magnetic trackers as in our case - is, like all measurements, prone to errors. Time lag, nonlinear distortions and noise have to be compensated, otherwise no registration is possible. This lies beyond the scope of this paper; relevant publications are [4][1].
- **Registration of Real and Virtual Objects:** Stationary and moving real objects in the working volume, which are to be augmented, have to be registered to their virtual counterparts. E.g. to attach a virtual label to a specific part of a real model, the system would have to know the models position and dimensions. A simple interactive method for registering virtual to real geometry is discussed in section 8.
- **Camera Calibration:** To achieve image registration, meaning alignment of real and virtual world when projected on the retinae of the user, we also need a precise description of the projection from the real world onto the retina. This step, called camera calibration for the remainder of this paper, has to determine the intrinsic and

extrinsic parameters of the virtual camera, which has to mimic the projection of the real environment.

The method we use for camera calibration [8] is simple, stable and interactively applicable. While specifically developed for our shared augmented environment *Studierstube* [15], the calibration procedure is generally applicable wherever quick adaptation of camera parameters to a specific user needed.

3 Previous Work

In photogrammetric camera calibrations [5][9][16][17], as used in data extraction from aerial photos and computer vision, data points in 3D and their 2D projection are measured. From these quintuples an optimization algorithm produces camera parameters, in many cases via the intermediary form of a projection matrix, from which camera parameters have to be extracted afterwards.

The main disadvantage of these methods when applied to our scenario lies in the relatively large sample of data points they need to converge (50-100) and tendency to behave unstable when presented with erroneous data.

Other camera calibration schemes use additional hardware for calibration:

Oishi et.al. [11] use a "shooting gallery", which presents calibration patterns at varying distances to the user, whose head has to be fixed in a previously defined position. The user has to tag all these points to achieve calibration.

Azuma et.al. [1] use orientation markers on a calibrated box, which requires the user to align at the same time two pairs of these markers to each other and a virtual to a real crosshair to define the eye position and orientation. Additional markers allow the measurement of FOV and aspect ratio.

Stationary objects in the real environment can be registered offline by measuring their position and dimensions in the tracker coordinate system.

By using image processing techniques [10][18] this can also be done for moving (rigid) objects during the simulation, but the necessary processing power and the low precision [18] and the lack of the necessary depth information [3] preclude us from following this approach. Furthermore they are only easily applicable for video-based augmented environments and suffer from line-of-sight problems.

Best results have been achieved when using pointer-based object registration [18], where users have to touch predefined landmarks on a real object with a tracked pointing device.

Like [18] we are using magnetic trackers on our movable augmented objects (PIP and the associated pen which also doubles as a 6DOF mouse). Methods of registering these movable tracked objects in the Augmented Environment are discussed in section 8.

4 Requirements

In the following, we discuss the requirements for a calibration method that can be used for augmented reality setups in everyday situations. More concretely, this section analyses requirements for a method that allows untrained users to calibrate the HMD to their personal parameters, resulting in improved registration.

- **Reduced User Effort:** As stated before [1][2] calibration requires in most cases direct interaction of the user. In the stated cases this interaction required a principal understanding of what different calibration steps were supposed to achieve and complex interactions with the system). We reduce user interaction to a guided approach, which in few, simple steps allows the untrained user to calibrate the HMD.
- **Usable for See-Through HMDs:** When using see-through HMDs, which optically overlay the computer images over reality, the users eyes - specifically their position relative to the display surface of the HMD - are an integral part of the registration problem and have to be taken into account when calibrating the virtual cameras. To achieve correct registration for a specific user we have to calibrate the HMD while this user is wearing it. Even slight differences in eye-distance and distance between eye and optical system of the HMD lead to misregistration.
- **No Additional Hardware:** Previous approaches [1][11] make use of additional hardware for calibration. They deliver high-quality calibration results at the expense of a complicated setup and considerable user effort. We avoid the use of additional hardware as far as possible, since it raises additional calibration problems (How to register the calibration hardware itself?) and reduces mobility of the whole system.
- **Numerically Stable:** Since the precision of the calibration depends on the users interaction, we have to find a method which presents us with a stable solution. This means that errors in some of the input data points should still produce a viable solution and not render the resulting calibration completely unusable. This instability may happen when a projection matrix is optimized without regard to its inherent redundancies. The requirement of reduced user effort implies an upper limit of the amount of input data, which could further increase instability.

5 Calibration Procedure

We use a simple pinhole camera model for calibration, since non-linear lens distortions are to be rectified independently (section 7).

The following parameters have to be calibrated: eye position, image plane position and orientation, aspect ratio, and field of view.

Our camera model takes into account the physically decoupled nature of eyepoint and image plane (Figure 3) in a see-through HMD. Since the users eye generally does not lie centered over the projection plane not only the determination of the viewing direction but also of the orientation of the image plane is necessary.

We have implemented a two-step optimization procedure, which optimizes these cameras parameters for a given set of data quintuple. Each quintuple contains the 3D coordinates of one sample point and its 2D projection.

The full calibration procedure consists of the following steps:

1. **Acquisition of Calibration Data** The user samples the positions of virtual markers with a 6DOF input device in an interactive process.
2. **Geometric determination of camera parameters** Using inherent geometric

properties of the acquired data, a viable solution is determined geometrically.

3. **Numerical Optimization** A further optimization step calculates a solution for off-axis projection.

5.1 Acquisition of Calibration Data

The properties of see-through HMDs make their calibration significantly different from the calibration of video-based HMDs.

Calibration of video-based augmentation [2] only determines the parameters of the video camera. Differences between the cameras parameters (FOV, interpupillary distance, etc.) and the users eyes are not taken into account, since the alignment of real and virtual images can be guaranteed in the first step. One advantage of this setup is that the video images of the real environment can be used to extract the coordinates of the projected data points using image-processing techniques.

The only place where the complete augmented image is visible when using a see-through HMD is at the retina of the user. While one may use a video camera in the position of the users eye, the resulting calibration will only be valid for the position of this camera. The data gathering stage of our calibration scheme, in which we have to acquire a quintuple of 3D coordinates of a point and its 2D projection, therefore has to rely on the user to identify whether a real point in space and its virtual image match.

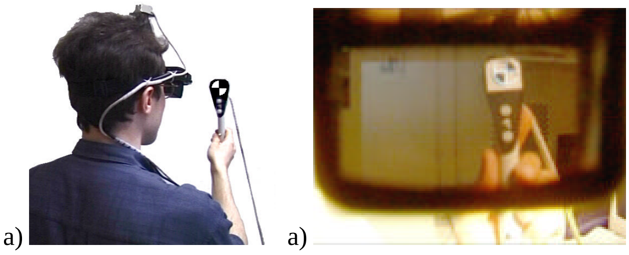

Fig. 2. a) physical setup for calibration;
b) View through the HMD - virtual marker overlays real marker

To achieve this, we turn the image processing approach - presenting a real calibration pattern and identifying points on its projection - around and present the user with a virtual calibration pattern, on which real points have to be aligned. The user sees a real marker on a tracked pen (Figure 2a), which he has to align with a virtual marker presented via the HMD. When the alignment is achieved (Figure 2b) the user presses a button on the pen and the next virtual marker is displayed. At the press of the button, tracking data of the sensors attached to the pen *and* the HMD is sampled. The position of the pen is transformed in the coordinate system of the HMD tracker sensor, which eliminates influences of the head position. The resulting 3D point gives us - together with the known 2D location of the virtual marker - one quintuple of calibration data.

5.2 Geometric Determination of Camera Parameters

Since we want to keep the number of sampled data points low (section 4), we need to maximize their information content with respect to our problem. We do this by imposing geometric constraints on the sampled points to allow direct determination of a viable start solution for our numerical optimization step.

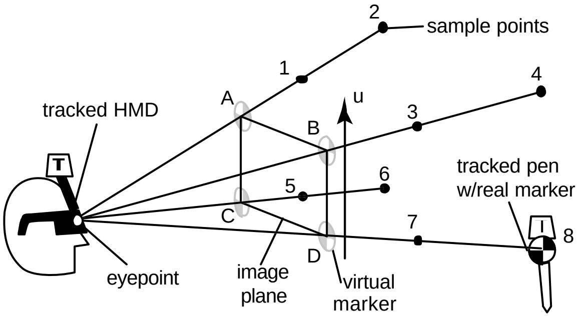

Fig. 3. geometric determination of HMD parameters

In Figure 3 the distribution of sample points for correct calibration is depicted as black circles (1-8). Every pair of samples lies on a line connecting one corner of the image plane with the eye point, essentially defining in this way the viewing pyramid.

In a first optimization step this gives us the location of the eyepoint as a least-squares solution for the point lying nearest to all of these lines. Averaging their direction gives us a good approximation of the viewing direction. As a first approximation at this stage we assume that the viewing direction is normal to the image plane. Intersecting this approximated image plane with the planes defined by the lines (1/2), (3/4), (5/6) and (7/8) gives us approximations for the horizontal, respective vertical directions in image space. Averaging the verticals (A/C, B/D) and the normals of the horizontals (A/B, C/D) gives us an approximation of the up-vector (u) for our camera model.

The aspect ratio of the display area may be approximated in a similar manner from the intersection points of the lines through the sample pairs with the image plane (A, B, C, D), but in most cases this parameter is known for a given HMD.

This intermediate solution already gives a good approximation of the calibration problem. Differences between calculated and measured projections of the data point are in the range of 1-2%. Since this solution only holds for eye positions on the axis of the optical system, we have to append an optimization procedure to account for off-axis positions of the eye.

5.3 Numerical Optimization of Parameters

The parameters optimized in this step are the normal vector of the image plane, field of view (FOV) and - if needed - aspect ratio. Since the solution at this stage is already very good, we do not have to apply sophisticated optimization techniques to it. Like in section 5.2, a simple multi-dimensional least-squares optimization [12] is being applied to the geometric solution reached in the previous step. The optimizations reaches a stable solution after 20-100 iterations and leads to errors in the range of 0.5-1%.

6 Fast Calibration Procedure

The procedures described above implement a full user-specific calibration of an unknown HMD with attached tracker sensor. It needs user input of 8 sample points per eye (a total of 16 samples per HMD) and yields a virtual camera pair, which is optimized for a specific user, but may be used satisfactory by others, if they are not willing to repeat the full procedure. Ideally we want a calibration which separates the user-specific parameters (eye position) from parameters fixed for a given HMD/tracker sensor setup similar to [7].

Given a sufficiently linear behavior of the optical system of the HMD which projects the display onto a virtual image somewhere in front of the user, we can assume that each point on the display surface corresponds to a point fixed in space relative to the HMD (Figure 3). It is then possible to determine this projection of points in image space (pixels) on a plane in the HMD coordinate system. This reduces the calibration problem to the determination of the eye position. It eliminates the need for more than one sample point along a edge of the sample frustum, since the (now) known 3D location of the virtual marker in real space in combination with the position of the real marker gives us a line in space through the eyepoint. This reduces the number of sample points per eye from 8 to 2.

We are using the full calibration data gathered from a sufficiently large user population to determine the orientation and position of the image plane in real space in a least-squares optimization step similar to section 3. When the eye distances of the users vary sufficiently this optimization terminates with a solution applicable for a wide range of users. While theoretically applicable with only 2 sample points per eye we normally use all 4 corner locations to optimize user effort versus error margin. The resulting errors (0.7-1.5%) are slightly higher than when full calibration is applied but the solution is still better than an "averaged" camera without user-specific calibration.

7 Distortion Compensation

A further problem for correctly registering and Augmented Environment are non-linear distortions introduced by the optical system of the HMD. Especially in video-based HMDs the short focal length of the cameras lens can introduce radial distortions as the typical barrel distortion depicted in Figure 4a. These distortions result from different focal lengths of the optical system for different points in the image plane. Equivalent distortions are introduced by the optical system inside a see-through HMD.

Photogrammetric calibration schemes like [16] mostly include some parameter(s) for describing radial distortions, which allows a rectification of image data on a point-by-point basis.

Since this approach does not work on a standard OpenGL graphics hardware, where only linear and perspective distortions are possible, we use an approach similar to the one used in [2], where video images are non-linearly distorted before being fed into a video-based HMD.

To implement a general image distortion using OpenGL hardware acceleration we take advantage of the OpenGL texture mapping mechanism. To rectify an image we use it as a texture and map it onto a screen-aligned segmented plane, the rectification grid. By using the inverse of the distortion function the image becomes rectified (Figure 4b).

In most cases a precise definition of the distortion function is not available, therefore we have to acquire it by measuring the distortion via a calibration pattern. Photogrammetric approaches deliver closed form solutions for the distortion function, [16] for example gives a first-order radial-lens distortion parameter for rectification.

Since we are sampling the inverse distortion function at discrete points - the vertices of the rectification grid - we do not need to determine the closed form of the function at all. All we have to do is take a distorted image of the calibration pattern in Figure 4a, measure the positions of the intersections and map these positions as *texture coordinates* onto the undistorted rectification grid. The texture hardware then rectifies the distorted images, using linear interpolations inside the grid squares.

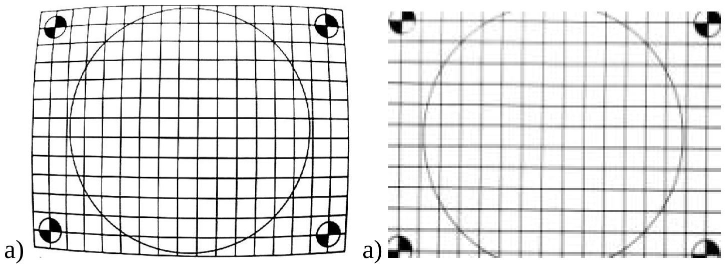

Fig. 4. a) distorted video image of calibration pattern
b) distortion compensated image

To apply this method to see-through HMDs we distort the computer-generated image instead of a video image. This is done by rendering into a separate region of the frame buffer and then using this region as a texture map. In this case of course the distortion has to be the same as the one seen through the HMD. Calibration data for this case can be gathered by letting the user click on markers lined up on the intersections of the rectification grid or - a simpler and faster approach - by capturing an image of the calibration pattern through the HMD. The coordinates of the intersections can then be measured by hand or with a feature recognition algorithm.

8 Registration of Virtual to Real Objects

Registration of real objects to virtual counterparts has several uses. In the case of a physically based simulation for examples a virtual ball has to bounce on a real table, or annotations and instructions have to appear aligned to components of a real object [6]. We have used tracked approximations of the users head and limbs to simulate the occlusion of virtual objects by the users body [7].

We have used an approach similar to [18] for registering static objects, but use a simpler method for registration of *tracked objects*. Tracked objects are objects like the PIP [14], or the occluding body geometry in [7], which in the virtual world are represented as rigid objects moving according to the tracker sensors attached to their real counterparts.

In our approach, the user is presented with an augmented view of the real object, overlaid with its (non-aligned) virtual representation (Figure 5a). Using keystrokes, the augmented geometry may be move *screen-aligned* to cover the actual object. Rotation around the center of the object in the image plane is supported also. The user aligns the virtual object with the real from one viewing direction as shown in Figure 5a and Figure 5b, then turns the physical object around to look at it from another direction (Figure 5c). The procedure is repeated until satisfactory registration is achieved (Figure 5d).

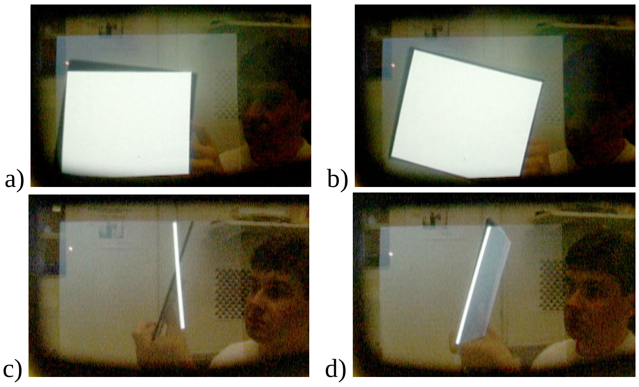

Fig. 5. registration of virtual to real object; a) front view – misaligned; b) front view – aligned; c) side view – misaligned; d) registered

This procedure is easily implemented and the user interface is intuitive enough for most users. Since the registration process has to be performed for each tracked object only once - baring changes in the attachment position of the tracker sensor - the necessary user effort is kept small. It allows for fast registration of arbitrarily shaped objects without the need to attach markers on the object.

9 Implementation details

The HMD calibration has been implemented in Open Inventor (OIV) [13], the same graphics toolkit in which our VE [15] has been implemented. The extended camera model has been implemented as an OIV extension, making it possible to save user-specific calibrated cameras in the standard OIV file format. The calibration runs as an applet and allows the user to calibrate the HMD on the fly anytime during his work in the environment. While this is normally done only once per user, passing a HMD to another user may make this necessary during a working session.

The procedure itself has been implemented as user-guiding step-by-step process, comparable to the "setup wizards" of commercial software. The user is shown virtual markers directly corresponding in size and shape the real marker, therefore not only prompting for a correct alignment in two dimension, but also prompting the correct distances. Figure 6a and b show the markers according to the upper right corner at near and far location respective. When the optimization procedure detects one or more measurements beyond the tolerated error margin the user gets prompted to repeat the necessary steps. During the calibration process the user may repeat measurements by using a "back" button on the pen.

As an unforeseen advantage proved the separation between calibration of left and right eye. Since some people have problems closing a single eye, they expected to have difficulties with having to align markers. These problems did not occur, since the virtual marker is only presented to one eye at a time.

The calibration method has been applied to stationary cameras, thereby fixing their position in space without the expense of an additional tracker sensor. We use this setup to generate videos for documentation purposes.

All the augmented illustrations in this paper have been captured with a digital camera directly through the optical system of our HMD - virtual-io i-glasses - and have been generated using a SGI Indigo 2 graphics workstation.

The distortion compensation has been implemented and tested on a SGI O2 R5000 workstation, using the O2 system camera as video source.

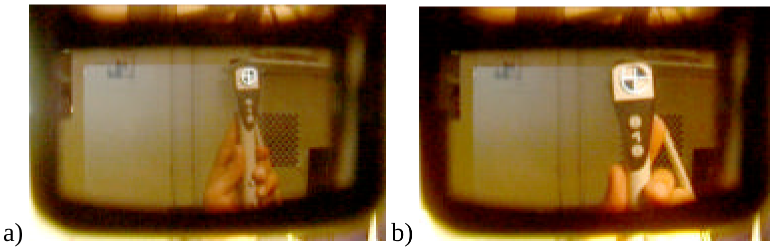

Fig. 6. guiding the user by differently sized markers. a) far sample point, b) near sample point

10 Results

Figure 7 shows the results of a full registration procedure. The virtual marker is in this case tracked and follows the real marker on the tracked pen over the whole working volume, independent of the users heads position and orientation.

The video images in Figure 8 demonstrate the real-time distortion compensation. The texture mapping method described in section 7 has been applied to a life video stream captured by a Silicon Graphics O2 system camera. The distinct barrel distortion in Figure 8a has been compensated in Figure 8b.

A first simple implementation of this method rectified the video at a rate of 22 frames per second (including video capturing time) on a SGI O2 R5000.

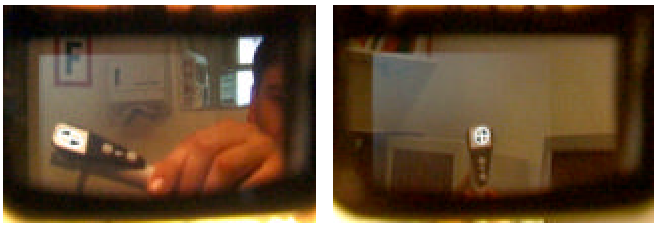

Fig. 7. resulting registration of virtual to real marker

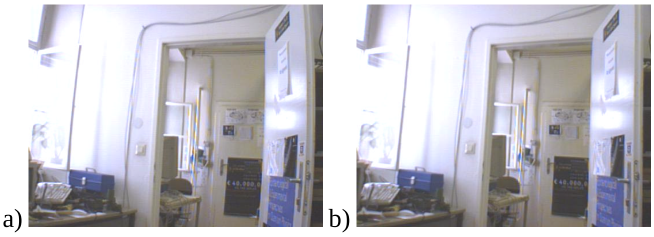

Fig. 8. a) distorted video image, b) distortion compensated image

11 Conclusions and future work

The advantages of the presented calibration methods lie mainly in their simplicity of implementation and ease of application. While testing other methods we observed a certain reluctance in many participants to apply the calibration procedure when presented with a not user-specific calibrated HMD or object. Our methods – especially when embellished by the user-guiding interface - seem to be more tolerable to the ordinary user. The simple task of matching one distinctive image with another proved to be easier to accept than more taxing schemes.

The disadvantages of the method in this context lie mostly in its dependence on precise user input and tracker precision. We are planning to apply Kalman filtering directly to the measurements already transformed into head-tracker coordinates. Since tracking delays and loss of high-frequency information are not relevant during the calibration procedure we expect to improve the precision considerably.

ACKNOWLEDGMENTS: Thanks to Rainer Splechtna for his excellent implementation. This work has been supported by the Austrian Science Funds (FWF) under project no. P-12074-MAT.

REFERENCES

[1] Azuma, R. and Gary Bishop, G. Improving Static and Dynamic Registration in an Optical See-Through HMD. Proceedings of SIGGRAPH '94, pp. 197-204, ACM Press, July 1994.

[2] Bajura, M., and Neumann, U. Dynamic Registration Correction in Augmented-Reality Systems. VRAIS'95, 1995.

[3] Berger, M. Resolving Occlusion In Augmented Reality: A Contour Based Approach Without 3D Reconstruction. Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, 1997, Poster Session 1.

[4] Bryson, S. Measurement and calibration of static distortion of position data from 3D trackers. Virtual Reality for Visualisation (IEEE Visualisation), pages 179-189, 1993.

[5] Faugeras, O. Three-dimensional computer vision: a geometric viewpoint. ISBN 0-262-06158-9.

[6] Feiner, S., MacIntyre, B., and Seligmann, D. Annotating the Real World with Knowledge-Based Graphics on a See-Through Head-Mounted Display. Proceedings of Graphics Interface'92, 78-85, 1992.

[7] Fuhrmann, A., Hesina, G., Faure, F. and Gervautz, M: Occlusion in Collaborative Augmented Environments. to appear in Proceedings 5th Eurographics Workshop on Virtual Environments, ISBN 3-211-83347-1, pages 179-190, Springer-Verlag, Wien, 1999.

[8] Fuhrmann, A., Schmalstieg, D., Purgathofer, W.: Simple User-Centered HMD Calibration for Augmented Reality. To appear in: Proceedings VRST'99.

[9] Janin, A., Mizell, D. and Caudell, T. Calibration of head-mounted displays for augmented reality applications. In Proceedings of the Virtual Reality Annual International Symposium (VRAIS '93), pages 246–255, September 1993.

[10] Kutulakos, K. and Vallino, J. Calibration-Free Augmented Reality. IEEE Transactions on Visualization and CG, 4(1), pp. 1-20, January 1998.

[11] Oishi T. and S. Tachi, S. Methods to Calibrate Projection Transformation Parameters for See-Through Head-Mounted Displays. Presence, 5(1), pp. 122-135, 1995.

[12] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W.. Numerical Recipes in C. Cambridge University Press, 1988.

[13] Strauss, P. and R. Carey, R.. An Object Oriented 3D Graphics Toolkit. Proceedings of SIGGRAPH'92, (2):341-347, 1992.

[14] Szalavári, Zs., M. Gervautz, M. The Personal Interaction Panel - A Two-handed Interface for Augmented Reality. Proc. EUROGRAPHICS 97, Budapest, Hungary, 335-346, 1997.

[15] Szalavari, Zs., Schmalstieg, Zs., Fuhrmann, A., Gervautz, M. Studierstube - An Environment for Collaboration in Augmented Reality, Virtual Reality: Research, Development & Applications, 1998

[16] Tsai, R.. An efficient and accurate camera calibration technique for 3D machine vision. Proceedings CVPR '86, pages 364-374, IEEE, June 1986.

[17] Tuceryan, M., Greer, D., Whitaker, R., Breen, D., Crampton, C., Rose, E., and Ahlers, K., Calibration Requirements and Procedures for Augmented Reality, IEEE Trans. on Visualization and Computer Graphics, September 1995.

[18] Whitaker, R., Crampton, C., Breen, D., Tuceryan, M., and Rose, E. Object Calibration for Augmented Reality. Proc. EUROGRAPHICS'95, pp. 15-27, 1995.