

Optical See-Through Calibration with Vision-Based Trackers: Propagation of Projection Matrices

Yakup Genc[†] Mihran Tuceryan[‡] Ali Khamene[†] Nassir Navab[†]

[†]Siemens Corporate Research
Imaging and Visualization Department
Princeton, NJ 08540, USA

[‡]Dept. of Computer and Information Sci.
Indiana U. Purdue U. (IUPUI)
Indianapolis, IN 46202, USA

Abstract

Recently, Tuceryan and Navab introduced a method for calibrating an optical see-through system based on the alignment of a set of 2D markers on the display with a single point in the scene while not restricting the user's head movements (the single point active alignment method or SPAAM in short) [33]. This method is applicable with any tracking system, provided that it gives the pose of the sensor attached to the see-through display.

When cameras are used for tracking, one can avoid the computationally intensive and potentially unstable pose estimation process. A vision-based tracker usually consists of a camera attached to the optical see-through display, which observes a set of known features in the scene. From the observed locations of these features, the pose of the camera can be computed. Most pose computation methods are very involved and can be unstable at times. Here, we propose to keep the projection matrix for the tracker camera without decomposing it into intrinsic and extrinsic parameters and use it within the SPAAM method directly. The propagation of the projection matrices from the tracker camera to the virtual camera, representing the eye and the optical see-through display combination as a pinhole camera model, allows us to skip the most time consuming and potentially unstable step of registration, namely, estimating the pose of the tracker camera.

1 Introduction

Augmented reality is a technology in which a user's view of the *real* world is enhanced with additional information generated from a computer model. The enhancements may include labels, 3D rendered models, or shading and illumination changes. Augmented reality allows a user to work with and examine the physical world, while receiving additional information about the objects in it. Some target application areas of augmented reality includes computer-aided surgery, repair

and maintenance of complex engines, facilities modification, and interior design.

In a typical augmented reality system, a user's view of a real scene is augmented by superimposing graphics on the view. The graphics are generated from geometric models of both virtual objects and real objects in the environment. In order for the graphics and the video to align properly, the pose and optical properties of the real and virtual cameras must be the same. The position and orientation of the real and virtual objects in some world coordinate system must also be known. The locations of the geometric models and virtual cameras within the augmented environment may be modified by moving its real counterpart. This is accomplished by tracking the location of the real objects and using this information to update the corresponding transformations within the virtual world. Once these capabilities have been brought together, real objects and computer-generated graphics may be blended together, thus augmenting a dynamic real scene with information stored and processed on a computer.

In order for augmented reality to be effective the real and virtual objects must be accurately positioned relative to each other and properties of the components of the system must be correctly specified. This implies that certain measurements or *calibrations* need to be made. These calibrations involve measuring the pose of various components such as the trackers, cameras, etc. What needs to be calibrated in an augmented reality system and how easy or difficult it is to accomplish this depends on the architecture of the particular system and the types of components used.

For calibration purposes, there are two major modes of display. These determine the types of technical issues that need to be addressed: (i) video see-through HMD systems and (ii) optical see-through systems. The calibration issues in a video see-through system are described in detail elsewhere [32]. We define an optical see-through HMD system as the combination of

an optical see-through display and a person's eyes.

In this paper, we look at the calibration issues in an augmented reality system equipped with an optical see-through HMD. In particular, we concentrate on the calibration of a monocular optical see-through HMD equipped with a rigidly attached tracker camera and describe a method of calibration of such a system. The method is also applicable to stereoscopic displays (see [11] for extension of SPAAM into the stereoscopic case).

Recently, Tuceryan and Navab introduced a method for calibrating an optical see-through system based on alignment of a set of 2D markers on the display with a single point in the scene while not restricting the movement of the user (the single point active alignment method or SPAAM in short) [33]. This method is applicable with any tracking system provided that it gives the pose of the sensor attached to the see-through display.

A vision-based tracker usually consists of a camera attached to the optical see-through display. The camera observes a set of features with known locations in the scene. From the observed locations of these features, the pose of the camera can be computed. Most pose computation methods are very involved and can be unstable at times. Here, we propose to reduce the complexity and instability by keeping the projection matrix for the tracker camera without decomposing it into intrinsic and extrinsic parameters and use it in the SPAAM method directly. The propagation of the projection matrices from the tracker camera to the virtual camera representing the eye and the optical see-through display combination as a pinhole camera model allows us to skip the most time consuming and potentially unstable step of registration, namely estimating the pose for the tracker camera.

The rest of the paper is organized as follows. Section 2 reviews the previous work done in the area. Section 3 describes the calibration algorithm and the results our experiments are presented in Section 4. We conclude with a summary of our findings and planned future work in Section 5.

2 Background

Research in augmented reality is recent but expanding. We briefly summarize the research conducted to date in this area. Baudel and Beaudouin-Lafon [3] have looked at the problem of controlling certain objects through the use of free hand gestures. Feiner et al. [9] have used augmented reality in a laser printer maintenance task in which the augmented reality system aids the user in the steps required to open the printer and replace various parts. Bajura et al. [2] have used augmented reality in medical applications in which the ultrasound imagery of a patient is superposed on the

patient's video image. Once more, the various registration issues, realism, etc. are open research questions which need to be studied and improved. Lorensen et al. [25] use augmented reality system in surgical planning applications.

Calibration has been an important aspect of research in augmented reality, as well as in other fields, including robotics and computer vision. Camera calibration, in particular, has been studied extensively in the computer vision community (e.g., [26, 34, 24]). Its use in computer graphics, however, has been limited. Deering [7] has explored the methods required to produce accurate high resolution head-tracked stereo display in order to achieve sub-centimeter virtual to physical registration. Azuma and Bishop [1], and Janin et al. [21] describe techniques for calibrating a see-through head-mounted display. Recently, Kato and Billinghurst describe an interactive camera calibration method [23] that uses multiple points on a grid. Gottschalk and Hughes [13] present a method for auto-calibrating tracking equipment used in augmented and virtual reality. Gleicher and Witkin [12] state that their through-the-lens controls may be used to register 3D models with objects in images. Grimson et al. [14] have explored vision techniques to automate the process of registering medical data to a patient's head.

Some researchers have studied the calibration issues relevant to head mounted displays [2, 6, 1, 19, 22]. Others have focused on monitor based approaches [32, 4, 15, 18, 28, 29]. Both approaches can be suitable depending on the demands of the particular application.

2.1 HMD Calibration using SPAAM

In [33], calibration of a single eye for a see-through head-mounted display (HMD) is discussed and an algorithm is presented. The algorithm is based on the observation of a single point in 3D space for varying viewing position of the user.

A simple pinhole model is used for the camera, which defines the basic imaging geometry with which the 3D objects are projected onto the 2D image plane. This is an ideal model commonly used in computer graphics and computer vision to capture the imaging geometry [20, 8]. It does not account for certain optical effects (such as non-linear distortions) that are often properties of real cameras but can be ignored in most cases.

The camera can be modeled by a set of intrinsic and extrinsic parameters. The intrinsic parameters are those that define the optical properties of the camera such as the focal length, the aspect ratio of the pixels, and the location of the image center, where the optical axis intersects the image plane. One last intrinsic parameter is the skew of the image plane axes. The

extrinsic parameters define the position and orientation (pose) of the camera with respect to some external world coordinate system. The 3D points in the world coordinate system get projected onto the image plane of the camera to form the image points.

The transformation that maps the 3D world points into the 2D image coordinates can be characterized by writing the transformation matrices for (i) the rigid transformation matrix defining the camera pose and (ii) the projection matrix defining the image formation process. Let R and T represent the camera pose in the world coordinate system, in which R is a 3×3 rotation matrix and T is a 3D vector representing the translation.

The rigid transformation can also be written as a 4×4 homogeneous matrix

$$M = \begin{pmatrix} R & T \\ \mathbf{0} & 1 \end{pmatrix}. \quad (1)$$

The perspective projection that projects the 3D points onto the image plane can be modeled by a 3×4 projection matrix given by

$$A = \begin{pmatrix} \alpha_u & s & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (2)$$

where α_u and α_v are horizontal and vertical scales in the image plane, u_0 and v_0 are the coordinates of the center of the image and s is the parameter that models the skew in the image plane coordinate axes. Note that M describes the extrinsic parameters of the camera and A describes the intrinsic camera parameters. The overall camera projection is then given by the product

$$\Pi = AM. \quad (3)$$

Given a set of 3D points and their images, the transformation Π can be recovered using linear least-squares methods.

The SPAAM includes some modifications to the traditional camera calibration method, which allows the user to move his head during the calibration procedure. We have a tracking system that measures the rigid transformation from the world coordinate system to the tracker coordinate system. Because the tracker (the receiver for magnetic tracking case and the tracker camera for the vision-based tracking case) is attached rigidly to the HMD, the camera can be defined and calibrated with respect to the tracker coordinate system. Therefore, taking this approach, we have the camera transformation fixed and unaffected by the head motion.

We assume that a tracker camera is rigidly attached to the HMD. The virtual camera formed by the eye and

the HMD combination, when considered as a pinhole camera, can be modeled with

$$\Pi_{\text{vir}} = \Pi_{\text{cam}} M_{\text{cam}},$$

where M_{cam} is the 4×4 transformation from the world to the tracker camera coordinate system, and Π_{cam} is the 3×4 projection matrix for the camera taking a point in the tracker coordinate system to the virtual image plane.

The projection matrix Π_{vir} varies as the HMD moves, and this movement is captured by the transformation M_{cam} which can be obtained from the tracking system. In the meantime, Π_{cam} is fixed and can be estimated by the SPAAM calibration procedure.

To summarize, in order to calibrate the virtual camera, i.e., to estimate the transformation Π_{cam} , we need to get the image coordinates of known 3D points in the world coordinate system. This is done by calibrating the camera in the tracker coordinate system, i.e., by estimating the transformation Π_{cam} which does not change with user head motion. In order to accomplish this we take the known 3D calibration point and transform it into the tracker coordinate system, then perform the standard camera calibration procedure on the new point.

The standard projective camera calibration is set up as follows. Let there be n calibration points whose image coordinates we measure. There are 12 parameters of the 3×4 projection matrix we need to estimate. However, since the projection matrix is defined up to a scale factor, there are only 11 independent parameters that need to be estimated. Therefore, at least 6 calibration points need to be measured.

2.2 Pose from a Camera

The SPAAM calibration method assumes that the transformation from the world to the tracker camera coordinate system is computed explicitly. When a set of known points are tracked in images obtained from the camera, its pose can be computed using a variety of methods. Here we review some of these for the sake of completeness.

One of the best known methods is due to Tsai [31]. In this method, the camera is first calibrated off-line for its intrinsic parameters. Once the intrinsic parameters are computed, the extrinsic parameters can be obtained for each time step from a set of observations. This has proven to be a good method and has been used in AR systems, see [5, 30] for examples.

Alternative methods abound, most of which assume that some sort of precalibration is performed similar to Tsai's. One particular method is from Zhang which is based on observation of coplanar points [35]. Coplanar points have the advantage of being easy to create.

However, this method does not provide good estimates for the depth of the camera from the scene. The reader is referred to [16, 8] for other alternatives.

3 Propagating the Projection Matrices

Instead of computing the rigid transformation from the world to the camera, one can propagate the normalized projection matrix related to the tracker camera to obtain the projection matrix for the virtual camera. Doing so will avoid the computationally expensive and potentially erroneous step of direct pose estimation. This section introduces a method to achieve this, while using the SPAAM method for calibration.

Figure 1 depicts the choice of the coordinate systems used in the following analysis. Note that the transformation from the tracker camera to the virtual camera representing the HMD and eye combination is rigid and does not change during the operations. On the other hand, the transformation from the world coordinate system to each of the cameras' coordinate systems changes during the operations.

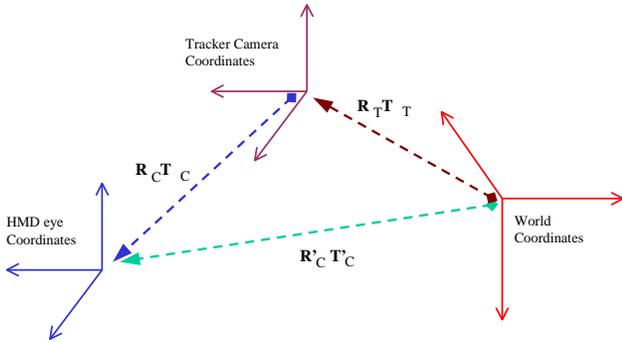


Figure 1: The coordinate systems associated with the tracker and virtual cameras.

Our aim is to estimate the projection matrices associated with the virtual camera in terms of the projection matrices associated with the tracker camera. It should be noted once again that at each instance of measurement, we have different projection matrices associated with both cameras due to the changing transformation between the world and the tracker as well as the virtual cameras.

Let us assume that we are observing a point in the scene, which is denoted by \mathbf{P} . The image of this point in the tracker camera will be obtained by the following equation assuming a pinhole camera model:

$$\rho_t \mathbf{Q}_t = \Pi_t \mathbf{P} = A_t [R_t \ \mathbf{T}_t] \mathbf{P}, \quad (4)$$

where A_t is the 3×3 upper triangular matrix representing internal parameters of the tracker camera, and R_t and \mathbf{T}_t represent the Euclidean transformation from

the world coordinate system to the tracker camera coordinate system. $\mathbf{Q}_t = (u_t, v_t, 1)^T$ is the image of the scene point $\mathbf{P} = (x, y, z, 1)^T$ as observed in this camera.

Similarly, the image of the point \mathbf{P} in the virtual camera can be obtained by the following:

$$\rho_c \mathbf{Q}_c = \Pi_c \mathbf{P} = A_c [R'_c \ \mathbf{T}'_c] \mathbf{P}, \quad (5)$$

where A_c holds the intrinsic parameters of the virtual camera, and R'_c and \mathbf{T}'_c represent the rigid transformation from the world coordinate system to the virtual camera coordinate system. Once again, $\mathbf{Q}_c = (u_c, v_c, 1)^T$ is the image of the point \mathbf{P} for this camera. This projection can also be written in terms of the tracker camera projection matrix as:

$$\rho_c \mathbf{Q}_c = ((A_c R_c A_t^{-1}) \lambda A_t [R_t \ \mathbf{T}_t] + [0_{3 \times 3} \ A_c \mathbf{T}_c]) \mathbf{P}$$

which in turn can be rewritten as

$$\rho_c \mathbf{Q}_c = ((A_c R_c A_t^{-1}) \lambda \Pi_t + [0_{3 \times 3} \ A_c \mathbf{T}_c]) \mathbf{P}, \quad (6)$$

where λ is the scale factor for Π_t (with the intrinsic matrix as defined in (2) and a rigid transformation, λ is in fact the z -coordinate of the translation from the world coordinate system to the tracker camera coordinate system), R_c and \mathbf{T}_c represent the transformation from the tracker camera coordinate system to the virtual camera coordinate system and $0_{3 \times 3}$ is a 3 by 3 zero matrix.

Observation 1: In (6), we have a motion dependent unknown λ , which is the z -coordinate of the translation between the world and tracker camera coordinate systems. This is true when the tracker camera moves in the world coordinate system¹.

Observation 2: We can estimate λ as described in Appendix A up to a sign ambiguity. This sign ambiguity is not so crucial since we have a knowledge about where the tracker camera is with respect to the markers, and hence, the world coordinate system (see [17] for an elegant analysis of this ambiguity for general case).

Thus, equation (6) can be parameterized as:

$$\rho_c \mathbf{Q}_c = (\lambda \Pi'_{c1} \Pi_t + [0_{3 \times 3} \ \Pi'_{c2}]) \mathbf{P}, \quad (7)$$

where the 12 parameters for $\Pi'_{c1} = A_c R_c A_t^{-1}$ and $\Pi'_{c2} = T A_c \mathbf{T}_c$ stay fixed when the tracker and the virtual cameras are fixed². In other words, the virtual

¹This has some additional implications such that the rotation component of the Euclidean transformation is orthonormal and the camera intrinsic matrix is normalized. This, in fact, ensures a Euclidean transformation is taking place which is what we want.

²The virtual camera is fixed when the HMD is fixed with respect to the user's head

camera has the following projection matrix taking a world point into its virtual image plane:

$$\Pi_c = \lambda \Pi'_{c1} \Pi_t + [0_{3 \times 3} \ \Pi'_{c2}]. \quad (8)$$

In summary, we can use the tracker camera projection matrix to calibrate the see-through system as follows:

- **Off-Line (Modified SPAAM):** Repeat the following at least 6 times.
 - Display a cross to the user.
 - Ask the user to align the cross with a known scene point.
 - Compute the tracker camera projection matrix Π_t from (4).
 - Plug in Π_t in (7) to obtain an observation for the alignment.
 - For 6 or more alignments estimate the 12 parameters (up to a scale) for Π'_{c1} and Π'_{c2} using (7).
- **On-Line:** Repeat the following for each tracker update.
 - Compute the tracker camera projection matrix Π_t from (4).
 - Obtain the virtual camera projection matrix from (8) using Π_t .
 - Plug the projection matrix in OpenGL pipeline as discussed in [33].

4 Preliminary Experiments

We have implemented and tested the method described in this paper on our optical see-through system as well as on a video see-through system to obtain quantitative results. For the sake of completeness, we briefly describe the optical see-through system here.

Our optical see-through system is illustrated in Figure 2. The HMD is a pair of *i-glasses*TM used in see-through mode. The graphical image is generated by a PC hardware and displayed on a monitor which is fed at the same time to the HMD over a VGA port. An infra-red vision-based tracker provides the system with the position and orientation of the tracker camera attached to the HMD. This camera observes a set of retro-reflective markers in the scene providing 12 points for pose estimation.

For comparison, we have implemented the original SPAAM algorithm where the pose of tracker camera is estimated using either Tsai’s or Zhang’s method as described earlier. We have also implemented the new method where the tracker camera observes 12 points 10

Tracker Camera

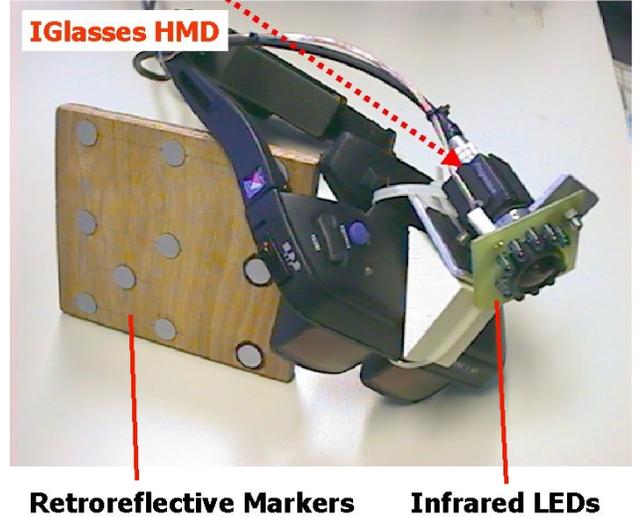


Figure 2: The hardware for the optical see-through system used in our experiments. A Sony XC-55BB camera with an infra-red filter is attached to an off-the-shelf HMD. The camera observes a set of retroreflective markers for tracking.

of which are coplanar. The method worked comparable to the pose estimation done using Tsai’s method and was more stable compared to Zhang’s method. Note that we are not able to provide any quantitative data for these experiments since we do not have any way of measuring the registration as observed by the user. One option is to measure the registration accuracy using the recent evaluation method proposed in [10, 27]. We plan to do this evaluation in the near future.

To obtain quantitative results, we diverted from our actual aim and used a set of cameras that are attached together rigidly observing a set of non-coplanar points. This is in fact a setup used in an existing video see-through system [30]. Figure 3 shows the particular configuration of the cameras used in the experiments. In the ideal case, this is analogous to our problem of optical see-through system when the virtual camera, defined by the combination of the eye and the HMD, represents an actual camera.

Figure 4 shows the re-projection of the scene points on the right camera for three different cases. In each case we obtain the projection matrix for the right camera with:

- Linear estimation of the projection matrix using the observations $[LM]^3$.

³This is provided here for the sole purposes of providing a

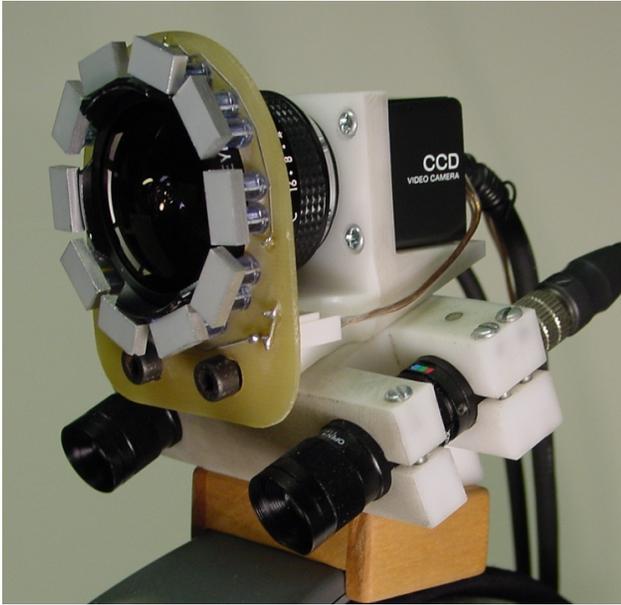


Figure 3: The setup used to obtain quantitative results. A tracker camera and two other cameras are attached rigidly together. One of the cameras are used to estimate the projection matrix for a second camera using the method proposed in this paper.

- Estimation of the projection matrix using the pose from the tracker camera and the calibration parameters from Tsai method [TM].
- Linear estimation of the projection matrix by propagation of the projection matrix linearly estimated for the tracker camera [PM].

Similar results are obtained for 6 other cases for the right as well as the left camera. The results in Figure 4 show that the reprojected points for LM, PM, and observed points coincide to the point that the individual shapes marking them in the plot become indistinguishable. TM reprojection has some visible reprojection error. This is also reflected in the error statistics in Figure 5.

We also obtained the error statistics for these experiments. We have looked at the distances in re-projection for the three cases with respect to the observed point locations. The results are provided in Figure 5. These preliminary results suggests that the new method performs better than the Tsai pose estimation method. Note that this level of performance is obtained just by linear estimation of the projection matrices and their

ground truth for comparison. In reality, neither in optical nor in video see-through cases we have 3D to 2D correspondences to estimate the projection matrix for the camera where the augmentation takes place.

propagations. Compared to non-linear optimization used in the Tsai method, the new method is more efficient computationally.

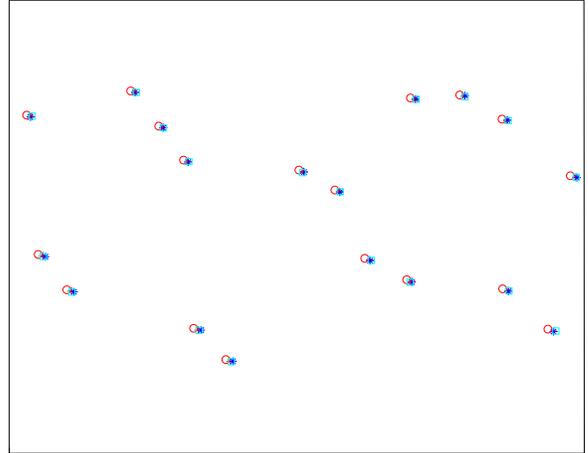


Figure 4: The re-projection of the points using the different projection matrices for comparison: the shapes “star”, “circle”, “cross” and “square” represent the point locations for LM, TM, PM and observations respectively (see the text for details).

Another experiment has been carried out to compare run-time complexities of the proposed method and the alternative based on Tsai’s calibration algorithm. When we use a pattern of 30 points in a non-generic configuration, the proposed algorithm runs in 15ms on the average. On the other hand, the Tsai based method runs in 25ms on the same computer. This is expected since Tsai’s calibration method is based on non-linear optimization whereas our method is based on linear least-squares.

5 Conclusions and Future Work

We have presented a direct method to estimate the projection matrix associated with the virtual camera representing the HMD and eye combination by using only the projection matrix of the tracker camera instead of explicitly estimating the motion and intrinsic parameters for the tracker camera. In this context, the SPAAM calibration method [33] for optical see-through systems stays the same except the pose estimation is not performed. The same method is also applicable to video see-through systems.

Our preliminary experiments demonstrated the viability of the method. In fact, it performed slightly better than the case where the pose is explicitly estimated using Tsai’s method while reducing the computational complexity. Note that the Tsai method involves non-linear optimization whereas the new method

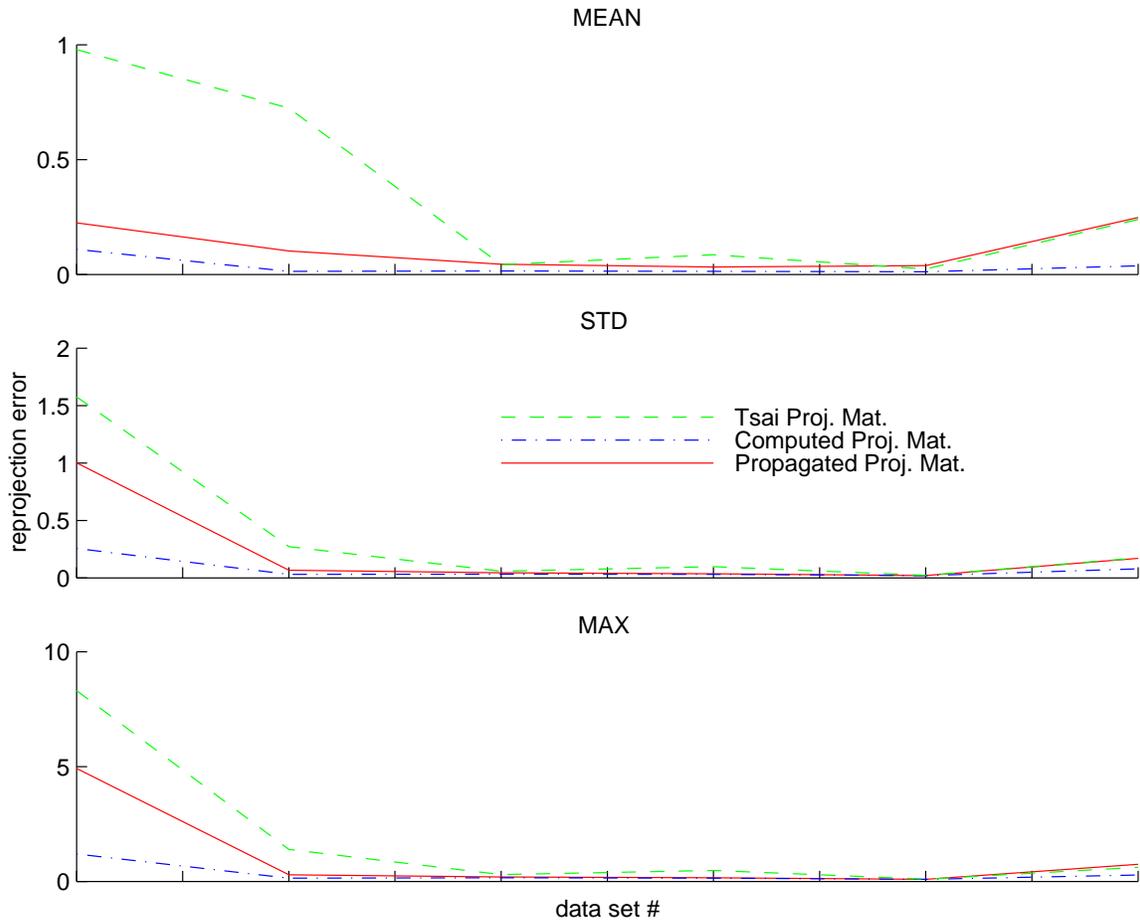


Figure 5: Error plots to compare the three methods. The distance in re-projection for the each method with respect to the observed point locations is taken as the error (in pixel). The new method performs better than the Tsai method. Note that the lower bound is given by the estimate projection matrix using 3D to 2D correspondences for the right camera. In reality, these correspondences are not available.

only solves a linear system of equations.

Future work will include an evaluation of the method using the method described in [10, 27] in order to compare the registration accuracy that can be obtained using different alternatives. We will also conduct more experiments for further validation.

Appendix

A. Recovering λ from a Projection Matrix

Let us assume that we are given the 3×4 projection matrix M taking a three-dimensional point P onto the two-dimensional point p on the image plane with equation:

$$\rho p = MP,$$

where

$$\lambda M = A_3[R \mathbf{T}],$$

with A_3 being the matrix formed by the first three columns of the camera matrix A in (2), R and \mathbf{T} being the Euclidean transformation from the world coordinate system to the camera coordinate system.

Let us further assume that M is normalized such that $M_{3,4} = 1$. This implies that $\lambda = T_z$, i.e., the z -coordinate of the translation vector. In this case, we can recover λ or T_z as follows: First we note that

$$T_z M_3 = AR,$$

yielding

$$T_z^2 = [M_3 M_3^T]_{3,3},$$

where M_3 is the matrix formed by the first three columns of M and $[\]_{3,3}$ is the entry of the matrix $[\]$ at 3rd row and 3rd column. As it can be seen from this equation, we can determine T_z^2 up to a sign ambiguity (the method described in [17] can be used to resolve the sign ambiguity).

References

- [1] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through display. *Computer Graphics*, pages 194–204, July 1994.
- [2] M. Bajura, H. Fuchs, and Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. *Computer Graphics*, pages 203–210, July 1992.
- [3] M. Baudel and M. Beaudouin-Lafon. Charade: Remote control of objects using freehand gestures. *Communications of the ACM*, 37(7):28–35, 1993.
- [4] F. Betting, J. Feldmar, N. Ayache, and F. Devernay. A framework for fusing stereo images with volumetric medical images. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed '95)*, pages 30–39, 1995.
- [5] W. Birkfellner, K. Huber, F. Watzinger, M. Figl, and F. Wanschitz. Development of the variscope ar: A see-through hmd for computer-aided surgery. In *International Symposium for Augmented Reality*, pages 54–59, Munich, Germany, October 2000.
- [6] T. Caudell and D. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Hawaii International Conference on System Sciences*, pages 659–669, 1992.
- [7] M. Deering. High resolution virtual reality. *Computer Graphics*, 26(2):195–202, 1992.
- [8] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, MA, 2001.
- [9] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(2):53–62, 1993.
- [10] Y. Genc, N. Navab, M. Tuceryan, and E. McGarrrity. Evaluation of optical see-through systems. Technical Report SCR-01-TR-688, Siemens Corporate Research, Inc., January 2001.
- [11] Y. Genc, F. Sauer, F. Wenzel, M. Tuceryan, and N. Navab. Stereo optical see-through hmd calibration: A one step method validated via a video see-through system. In *International Symposium for Augmented Reality*, pages 165–174, Munich, Germany, October 2000.
- [12] M. Gleicher and A. Witkin. Through-the-lens camera control. *Computer Graphics*, pages 331–340, July 1992.
- [13] S. Gottschalk and J. Hughes. Autocalibration for virtual environments tracking hardware. *Computer Graphics*, pages 65–72, August 1993.
- [14] E. Grimson, T. Lozano-Perez, W. M. Wells, G. J. Ettinger, S. J. White, and R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–436, Seattle, WA, June 1994.
- [15] W.E.L. Grimson, G. J. Ettinger, S.J. White, P.L. Gleason, T. Lozano-Perez, III W. M. Wells, and

- R. Kikinis. Evaluating and validating an automated registration system for enhanced reality visualization in surgery. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 3–12, 1995.
- [16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [17] R.I. Hartley. Chirality invariants. In *Proc. DARPA Image Understanding Workshop*, pages 745–753, Washington, D.C., 1993.
- [18] C. J. Henri, A.C.F. Colchester, J. Zhao, D.J. Hawkes, D.L.G. Hill, and R. L. Evans. Registration of 3D surface data for intra-operative guidance and visualization in frameless stereotactic neurosurgery. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 47–58, 1995.
- [19] R. Holloway. *An Analysis of Registration Errors in a See-Through Head-Mounted Display System for Craniofacial Surgery Planning*. PhD thesis, University of North Carolina at Chapel Hill, 1994.
- [20] B.K.P. Horn. *Computer Vision*. MIT Press, Cambridge, Mass., 1986.
- [21] A. Janin, D. Mizell, and T. Caudell. Calibration of head-mounted displays for augmented reality applications. In *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, pages 246–255, 1993.
- [22] A. R. Kancherla, J. P. Rolland, D. L. Wright, and G. Burdea. A novel virtual reality tool for teaching dynamic 3D anatomy. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 163–169, 1995.
- [23] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99*, pages 85–94, San Francisco, CA, October 20–21, 1999.
- [24] R. K. Lenz and R.Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-10:713–720, 1988.
- [25] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altabelli, and L. Gleason. Enhancing reality in the operating room. In *Proceedings of the IEEE Conference on Visualization*, pages 410–415, 1993.
- [26] S. J. Maybank and O. D. Faugeras. A theory of self calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.
- [27] E. McGarrity, M. Tuceryan, C. Owen, Y Genc, and N. Navab. Evaluation of optical see-through systems. In *International Conference on Augmented, Virtual Environments and 3D Imaging*, pages 18–21, Mykonos, Greece, May 2001.
- [28] J. P. Mellor. Real-time camera calibration for enhanced reality visualizations. In *Proc. of the IEEE Conference on Computer Vision Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 471–475, 1995.
- [29] O. Peria, L. Chevalier, A. François-Joubert, J. P. Caravel, S. Dalsoglio, S. Lavallee, and P. Cinquin. Using a 3D position sensor for registration of spect and us images of the kidney. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 23–29, 1995.
- [30] F. Sauer, F. Wenzel, S. Vogt, Y. Tao, Y. Genc, and A. Bani-Hashemi. Augmented workspace: Designing an AR testbed. In *International Symposium for Augmented Reality*, pages 165–174, Munich, Germany, October 2000.
- [31] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.
- [32] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255–273, September 1995.
- [33] M. Tuceryan and N. Navab. Single point active alignment method (spaam) for optical see-through hmd calibration for ar. In *International Symposium for Augmented Reality*, pages 149–158, Munich, Germany, October 2000.
- [34] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.

- [35] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Patt. Anal. Mach. Intell.*, 41(11):1330–1334, 2000.