Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99,
pp.85-94, October 20-21, 1999 San Francisco

Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System

Hirokazu Kato[1] and Mark Billinghurst[2]
[1]*Faculty of Information Sciences, Hiroshima City University*
[2]*Human Interface Technology Laboratory, University of Washington*
kato@sys.im.hiroshima-cu.ac.jp, grof@hitl.washington.edu

Abstract

We describe an augmented reality conferencing system which uses the overlay of virtual images on the real world. Remote collaborators are represented on Virtual Monitors which can be freely positioned about a user in space. Users can collaboratively view and interact with virtual objects using a shared virtual whiteboard. This is possible through precise virtual image registration using fast and accurate computer vision techniques and HMD calibration. We propose a method for tracking fiducial markers and a calibration method for optical see-through HMD based on the marker tracking.

1. Introduction

Computers are increasingly used to enhance collaboration between people. As collaborative tools become more common the Human-Computer Interface is giving way to a Human-Human Interface mediated by computers. This emphasis adds new technical challenges to the design of Human Computer Interfaces. These challenges are compounded for attempts to support three-dimensional Computer Supported Collaborative Work (CSCW). Although the use of spatial cues and three-dimensional object manipulation are common in face-to-face collaboration, tools for three-dimensional CSCW are still rare. However new 3D interface metaphors such as virtual reality may overcome this limitation.

Virtual Reality (VR) appears a natural medium for 3D CSCW; in this setting computers can provide the same type of collaborative information that people have in face-to-face interactions, such as communication by object manipulation, voice and gesture [1]. Work on the DIVE project [2], GreenSpace [3] and other fully immersive multi-participant virtual environments has shown that collaborative work is indeed intuitive in such surroundings. However most current multi-user VR systems are fully immersive, separating the user from the real world and their traditional tools.

As Grudin [4] points out, CSCW tools are generally rejected when they force users to change the way they work. This is because of the introduction of seams or discontinuities between the way people usually work and the way they are forced to work because of the computer interface. Ishii describes in detail the advantages of seamless CSCW interfaces [5]. Obviously immersive VR interfaces introduce a huge discontinuity between the real and virtual worlds.

An alternative approach is through Augmented Reality (AR), the overlaying of virtual objects onto the real world. In the past researchers have explored the use of AR approaches to support face-to-face collaboration. Projects such as Studierstube [6], Transvision [7], and AR2 Hockey [8] allow users can see each other as well as 3D virtual objects in the space between them. Users can interact with the real world at the same time as the virtual images, bringing the benefits of VR interfaces into the real world and facilitating very natural collaboration. In a previous paper we found that this meant that users collaborate better on a task in a face-to-face AR setting than for the same task in a

fully immersive Virtual Environment [9].

We have been developing a AR conferencing system that allows virtual images (Virtual Monitors) of remote collaborators to be overlaid on the users real environment. Our Augmented Reality conferencing system tries to overcomes some of the limitations of current desktop video conferencing, including the lack of spatial cues [10], the difficulty of interacting with shared 3D data, and the need to be physically present at a desktop machine to conference. While using this system, users can easily change the arrangement of Virtual Monitors, placing the virtual images of remote participants about them in the real world and they can collaboratively interact with 2D and 3D information using a Virtual Shared Whiteboard. The virtual images are shown in a lightweight head mounted display, so with a wearable computer our system could be made portable enabling collaboration anywhere in the workplace.

In developing a multi-user augmented reality video conferencing system, precise registration of the virtual images with the real world is one of the greatest challenges. In our work we use computer vision techniques and have developed some optimized algorithms for fast, accurate real time registration and convenient optical see-through HMD calibration. In this paper, after introducing our conferencing application, we describe the video-based registration and calibration methods used.

2. System overview

Our prototype system supports collaboration between a user wearing see-through head mounted displays(HMD) and those on more traditional desktop interfaces as shown in figure 1. This simulates the situation that could occur in collaboration between a desk bound expert and a remote field worker. The user with the AR head mounted interface can see video images from desktop users and be supported by them. Remote desktop users can see the video images that the small camera of the AR user grabs and give support to the AR user. This system dose not support video communication among desktop users. If necessary, however, they could simultaneously execute a traditional video communication application. In this section we first describe the AR head mounted interface and then the desktop interface.

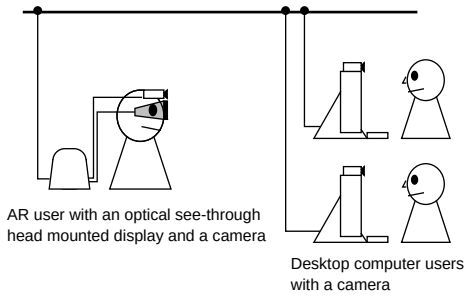

Figure 1. System configuration.

2.1. Augmented Reality Interface

The user with the AR interface wears a pair of the Virtual i-O iglasses HMD that have been modified by adding a small color camera. The iglasses are full color, can be used in either a see-through or occluded mode and have a resolution of 263x234 pixels. The camera output is connected to an SGI O2 (R5000SC 180MHz CPU) computer and the video out of the SGI is connected back into the head mounted display. The O2 is used for both image processing of video from the head mounted camera and virtual image generation for the HMD. Performance speed is 7-10 frames per sec for full version, 10-15 fps running without the Virtual Shared Whiteboard.

The AR user also has a set of small marked cards and a larger piece of paper with six letters on it around the outside. There is one small marked card for each remote collaborator with their name written on it. These are placeholders (user ID cards) for the Virtual Monitors showing the remote collaborators, while the larger piece of paper is a placeholder for the shared white board. To write and interact with virtual objects on the shared whiteboard the user has a simple light pen consisting of an LED, switch and battery mounted on a pen. When the LED touches a surface the switch is tripped and it is turned on. Figure 2 shows an observer's view of the AR user using the interface.

The software components of the interface consist of two parts, the Virtual Monitors shown on the user ID cards, and the Virtual Shared Whiteboard. When the system is running, computer vision techniques are used to identify specific user

Figure 2. Using the Augmented Reality Interface.

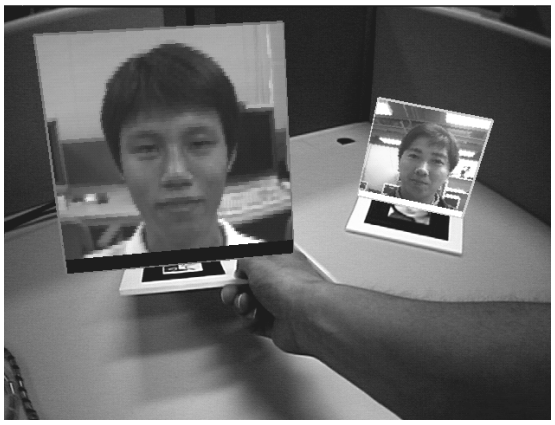

Figure 3. Remote user representation in the AR interface.

ID cards (using the user name on the card) and display live video of the remote user that corresponds to the ID card. Vision techniques are also used to calculate head position and orientation relative to the cards so the virtual images are precisely registered with the ID cards. Figure 3 shows an example of a Virtual Monitor, in this case the user is holding an ID card which has live video from a remote collaborator attached to it.

Shared whiteboards are commonly using in collaborative applications to enables people to share notes and diagrams. In our application we use a Virtual Shared Whiteboard as seen in figure 4. This is shown on a larger paper board with six similar registration markings as the user ID cards. Virtual annotations written by remote participants are displayed on it, exactly aligned with the plane of the physical card. The local participant can use the light-pen to draw on the card and add their own annotations, which are in turn displayed and transferred to the remote desktops. The user can erase

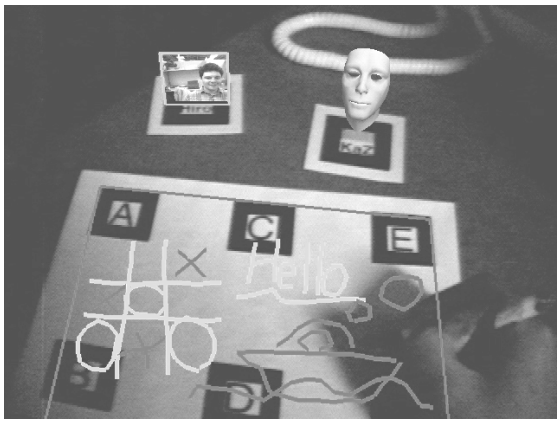

Figure 4. Virtual Shared White Board.

their own annotations by touching one corner of the card. Currently our application only supports virtual annotations aligned with the surface of the card, but we are working on adding support for shared 3D objects.

The position and pose of this paper board can be estimated by using the same vision methods used for the virtual monitors. However, since the user's hands often occlude the registration markers, the estimation has to be done by using only visible markers. We can reliably estimate the card position using only one of the six markers. The LED light-pen is on while it touches the paper board. When this happens the system estimates the position of the pen tip relative to the paper board from the 2D position of the LED in the camera image and the knowledge that the tip of the pen is contact with the board. Users can pick up the card for a closer to look at the images on the virtual whiteboard, and can position it freely within their real workspace.

2.2. Desktop Interface

The AR user collaborates with remote desktop users that have a more traditional interface. The desktop users are on networked SGI computers. Users with video cameras on their computer see a video window of the video image that their camera is sending, the remote video from the AR head mounted camera and a share white board application. The video from the AR user's head mounted camera enables the desktop user to collaborate more effectively with them on real world tasks. They can freely draw on the shared white board using the mouse, and whiteboard annotations and video frames from their camera are send to the AR user.

3. Video-based registration techniques

Our AR conferencing interface relies heavily on computer vision techniques for ID recognition and user head position and pose determination. In the remainder of the paper we outline the underlying computer vision methods we have developed to accomplish this. These methods are general enough to be applicable for a wide range of augmented reality applications.

Augmented Reality Systems using HMDs can be classified into two groups according to the display method used:

Type A: Video See-through Augmented Reality

Type B: Optical See-through Augmented Reality

In type A, virtual objects are superimposed on a live video image of the real world captured by the camera attached to the HMD. The resulting composite video image is displayed back to both eyes of the user. In this case, interaction with the real world is a little unnatural because the camera viewpoint shown in the HMD is offset from that of the user's own eyes, and the image is not stereographic. Performance can also be significantly affected as the video frame rate drops. However, this type of system can be realized easily, because good image registration only requires that the relationship between 2D screen coordinates on the image and 3D coordinates in the real world is known.

In type B, virtual objects are shown directly on the real world by using a see-through display. In this case, the user can see the real world directly and stereoscopic virtual images can be generated so the interaction is very natural. However, the image registration requirements are a lot more challenging because it requires the relationships between the camera, the HMD screens and the eyes to be known in addition to the relationships used by type A systems. The calibration of the system is therefore very important for precise registration.

Azume reported a good review of the issues faced in augmented reality registration and calibration[11]. Also many registration techniques have been proposed. State proposed a registration method using stereo images and a magnetic tracker[12]. Neumann used a single camera and multiple fiducial markers for robust tracking[13]. Rekimoto used vision techniques to identify 2D matrix markers[14]. Klinker used square markers for fast tracking[15]. Our approach is similar to this method.

We have developed a precise registration method for the optical see-through augmented reality system. Our method overcomes two primary problems; calibration of the HMD and camera, and estimating an accurate position and pose of fiducial markers. We first describe a position and pose estimation method, and then HMD and camera calibration method, because our HMD calibration method is based on the fiducial marker tracking.

4. Position and pose estimation of markers

4.1. Estimation of the Transformation Matrix

Size-known square markers are used as a base of the coordinates frame in which Virtual Monitors are represented (Figure 5). The transformation matrices from these marker coordinates to the camera coordinates ($\mathbf{T}_{cm}$) represented in eq.1 are estimated by image analysis.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{3\times3} & \mathbf{W}_{3\times1} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T}_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad \text{(eq. 1)}$$

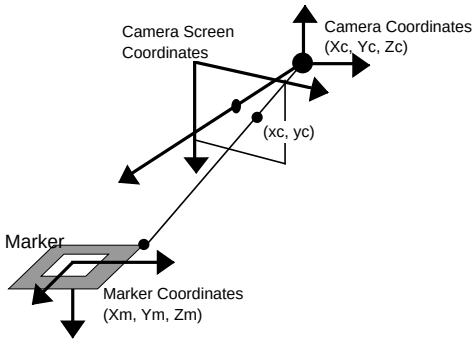

Figure 5. The relationship between marker coordinates and the camera coordinates is estimated by image analysis.

After thresholding of the input image, regions whose outline contour can be fitted by four line segments are extracted. Parameters of these four line segments and coordinates of the four vertices of the regions found from the intersections of the line segments are stored for later processes.

The regions are normalized and the sub-image within the region is compared by template matching with patterns that were given the system before to identify specific user ID markers. User names or photos can be used as identifiable patterns. For this normalization process, eq.2 that represents a perspective transformation is used. All variables in the transformation matrix are determined by substituting screen coordinates and marker coordinates of detected marker's four vertices for (x_c, y_c) and (X_m, Y_m) respectively. After that, the normalization process can be done by using this transformation matrix.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ 1 \end{bmatrix} \qquad \text{(eq. 2)}$$

When two parallel sides of a square marker are projected on the image, the equations of those line segments in the camera screen coordinates are the following:

$$a_1 x + b_1 y + c_1 = 0, \qquad a_2 x + b_2 y + c_2 = 0 \qquad \text{(eq. 3)}$$

For each of markers, the value of these parameters has been already obtained in the line-fitting process. Given the perspective projection matrix **P** that is obtained by the camera calibration in eq.4, equations of the planes that include these two sides respectively can be represented as eq.5 in the camera coordinates frame by substituting x_c and y_c in eq.4 for x and y in eq.3.

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \qquad \text{(eq. 4)}$$

$$\begin{aligned} a_1 P_{11} X_c + (a_1 P_{12} + b_1 P_{22}) Y_c + (a_1 P_{13} + b_1 P_{23} + c_1) Z_c = 0 \\ a_2 P_{11} X_c + (a_2 P_{12} + b_2 P_{22}) Y_c + (a_2 P_{13} + b_2 P_{23} + c_2) Z_c = 0 \end{aligned} \qquad \text{(eq. 5)}$$

Given that normal vectors of these planes are $\mathbf{n}_1$ and $\mathbf{n}_2$ respectively, the direction vector of parallel two sides of the square is given by the outer product $\mathbf{n}_1 \times \mathbf{n}_2$. Given that two unit direction vectors that are obtained from two sets of two parallel sides of the square is $\mathbf{u}_1$ and $\mathbf{u}_2$, these vectors should be perpendicular. However, image processing errors mean that the vectors won't be exactly perpendicular. To compensate for this two perpendicular unit direction vectors are defined by $\mathbf{v}_1$ and $\mathbf{v}_2$ in the plane that includes $\mathbf{u}_1$ and $\mathbf{u}_2$ as shown in figure 6. Given that the unit direction vector which is perpendicular to both $\mathbf{v}_1$ and $\mathbf{v}_2$ is $\mathbf{v}_3$, the rotation component $\mathbf{V}_{3x3}$ in the transformation matrix $\mathbf{T}_{cm}$ from marker coordinates to camera coordinates specified in eq.1 is $[\mathbf{V}_1^t \mathbf{V}_2^t \mathbf{V}_3^t]$.

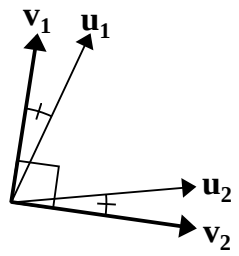

Figure 6. Two perpendicular unit direction vectors: v_1, v_2 are calculated from u_1 and u_2.

Since the rotation component $\mathbf{V}_{3x3}$ in the transformation matrix was given, by using eq.1, eq.4, the four vertices coordinates of the marker in the marker coordinate frame and those coordinates in the camera screen coordinate frame, eight equations including translation component W_x W_y W_z are generated and the value of these translation component W_x W_y W_z can be obtained from these equations.

The transformation matrix found from the method mentioned above may include error. However this can be reduced through the following process. The vertex coordinates of the markers in the marker coordinate frame can be transformed to coordinates in the camera screen coordinate frame by using the transformation matrix obtained. Then the transformation matrix is optimized as sum of the difference between these transformed coordinates and the coordinates measured from the image goes to a minimum. Though there are six independent variables in the transformation matrix, only the rotation components are optimized and then the translation components are reestimated by using the method mentioned above. By iteration of this process a number of times the transformation

matrix is more accurately found. It would be possible to deal with all of six independent variables in the optimization process. However, computational cost has to be considered.

4.2. An Extension for the Virtual Shared White Board

The method described for tracking user ID cards is extended for tracking the shared whiteboard card. There are six markers in the Virtual Shared White Board, aligned around the outside of the board as shown in figure 7. The orientation of the White Board is found by fitting lines around the fiducial markers and using an extension of the technique described for tracking user ID cards.

Using all six markers to find the board orientation and align virtual images in the interior produces very good registration results. However, when a user draws a virtual annotation, some markers may be occluded by user's hands, or they may move their head so only a subset of the markers are in view. The transformation matrix for Virtual Shared White Board has to be estimated from visible markers so errors are introduced when fewer markers are available. To reduce errors the line fitting equations are found by both considering individual markers and sets of aligned markers. Each marker has a unique letter in its interior that enables the system to identify markers which should be horizontally or vertically aligned and so estimate the board rotation. Though line equations in the camera screen coordinates frame are independently generated for each of markers, the alignment of the six markers in Virtual Shared White Board means that some line equations are identical. Therefore by extracting all aligned sides from visible markers for the line-fitting, each line equation is calculated by using all the contour information that is on the extracted sides. Furthermore by using all the equations of the detected parallel lines, the direction vectors are estimated and the board orientation is found.

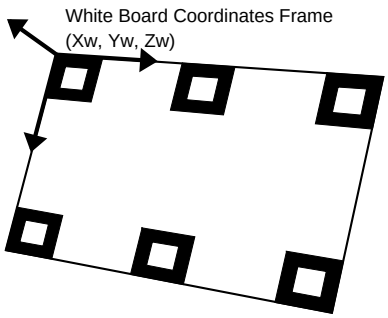

Figure 7. Layout of markers on the Shared White Board.

4.3. Pen Detection

The light-pen is on while touching the shared whiteboard board. Estimation of the pen tip location is found in the following way. First, the brightest region in the image is extracted and the center of the gravity is detected. If brightness and area of the regions are not satisfied with heuristic rules, the light-pen is regarded as turned off status.

Since pen position (X_w, Y_w, Z_w) is expressed relative to the Virtual Shared Whiteboard it is detected in the whiteboard coordinate frame. The relationship between the camera screen coordinates and the whiteboard coordinates is given by eq.6. (x_c, y_c) is a position of the center of gravity that is detected by image processing. Also Z_w is equal to zero since pen is on the board. By using these values in eq.6, two equations including X_w and Y_w as variables are generated and their values are calculated easily by solving these equations.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{(eq. 6)}$$

5. HMD and Camera Calibration

In an optical see-through HMD, a ray from a physical object reaches the focal point of the eye through the HMD screen. Then, a 3D position represented in the eye coordinates whose origin is the focal point of the eye can be projected on the HMD screen coordinates by the perspective projection model. This assumes that the Z axis perpendicularly crosses the HMD screen, and the X and Y axes are parallel to X and Y axes of the HMD screen coordinates frame respectively.

Figure 8 shows coordinates frames in our calibration

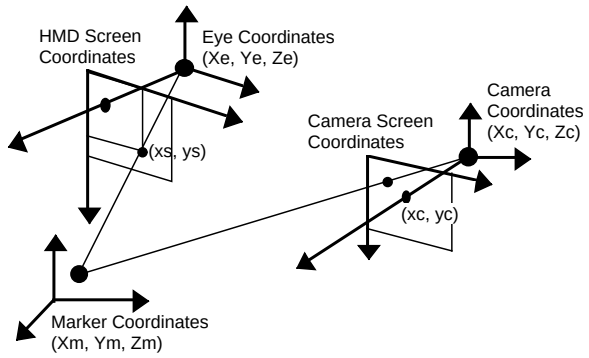

Figure 8. Coordinates frames in our calibration procedure.

procedure. As mentioned in section 4, position and pose estimation of a marker is done by calculating the transformation matrix from marker coordinates to camera coordinates: $\mathbf{T}_{cm}$ (eq.1). The perspective projection matrix **P** (eq.4) is required in this procedure. Camera calibration is to find the perspective projection matrix **P** that represents the relationship between the camera coordinates and the camera screen coordinates.

In order to display virtual objects on HMD screen as if those are on the marker, the relationship between the marker coordinates and the HMD screen coordinates is required. Relationship between HMD screen coordinates and eye coordinates is represented by the perspective projection. Also, relationship between camera coordinates and eye coordinates is represented by rotation and translation transformations. eq.7 shows those relationship.

$$\begin{bmatrix} ix_s \\ iy_s \\ i \\ 1 \end{bmatrix} = \mathbf{Q}_{se} \begin{bmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{bmatrix} = \mathbf{Q}_{se}\mathbf{T}_{ec} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{Q}_{se}\mathbf{T}_{ec}\mathbf{T}_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad \text{(eq. 7)}$$

$\mathbf{Q}_{se}$: Perspective transformation matrix

$\mathbf{T}_{ec}$: Rotation and translation matrix

Matrix $\mathbf{T}_{cm}$ representing the transformation from marker coordinates to camera coordinates is obtained in use of the system as mentioned in Section 4. HMD calibration is therefore to find the matrix $\mathbf{Q}_{se}\mathbf{T}_{ec}$ for both of eyes.

5.1. Camera Calibration - Finding the matrix P

We use a simple cardboard frame with a ruled grid of lines for the camera calibration. Coordinates of all cross points of a grid are known in the cardboard local 3D coordinates. Also those in the camera screen coordinates can be detected by image processing after the cardboard image is grabbed. Many pairs of the cardboard local 3D coordinates (X_t, Y_t, Z_t) and the camera screen coordinates (x_c, y_c) are used for finding the perspective transformation matrix **P**.

The relationships among the camera screen coordinates (x_c, y_c), the camera coordinates (X_c, Y_c, Z_c) and the cardboard coordinates (X_t, Y_t, Z_t) can be represented as:

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{P} \cdot \mathbf{T}_{ct} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} s_x f & 0 & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_{ct} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(eq. 8)}$$

where **P** is the perspective transformation matrix which should be found here, f is the focal length, s_x is the scale factor [pixel/mm] in direction of x axis, s_y is the scale factor in direction of y axis, (x_0, y_0) is the position that Z axis of the eye coordinates frame passes, $\mathbf{T}_{ct}$ represents the translation and rotation transformation from the cardboard coordinates to the camera coordinates and **C** is the transformation matrix obtained by combining **P** and $\mathbf{T}_{ct}$.

Since many pairs of (x_c, y_c) and (X_t, Y_t, Z_t) have been obtained by the procedure mentioned above, matrix **C** can be estimated. However, the matrix **C** cannot be decomposed into **P** and $\mathbf{T}_{ct}$ in general because matrix **C** has 11 independent variables but matrices **P** and $\mathbf{T}_{ct}$ have 4 and 6 respectively, so the sum of the independent variables of **P** and $\mathbf{T}_{ct}$ is not equal to the one of **C**. A scalar variable k is added into **P** to make these numbers equal as the following:

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & k & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} \quad \text{(eq. 9)}$$

As a result, the matrix **C** can be decomposed into **P** and $\mathbf{T}_{ct}$. The variable k means the slant between x-axis and y-axis and should be zero ideally but it may be a small noise value.

5.2 HMD Calibration - Finding the matrix $\mathbf{Q}_{se}\mathbf{T}_{ec}$

Formulation of the matrix $\mathbf{Q}_{se}\mathbf{T}_{ec}$ is same as one of the matrix **C** in eq.8. Therefore many pairs of the coordinates (x_s, y_s) and (X_c, Y_c, Z_c) can be used for finding the transformation matrix combining $\mathbf{Q}_{se}$ and $\mathbf{T}_{ec}$. In order to obtain such kinds of data, we use marker tracking technique introduced in section 4.

HMD calibration procedure is done for each eye. A cross-hair cursor is displayed on the corresponding HMD screen. The user handles a fiducial marker and fits its center on the cross-hair cursor as shown in figure 9. The fiducial marker is simultaneously observed by the camera attached on the HMD and the central coordinates are detected in the camera coordinates. While the user manipulates the marker from near side to far side, some marker positions are stored by clicking a mouse button. In this procedure, positions of the cross-hair cursor mean HMD screen coordinates (x_s, y_s) and marker positions mean camera coordinates (X_c, Y_c, Z_c). After iterating this operation in some positions of cross-hair cursor, many pairs of (x_s, y_s) and (X_c, Y_c, Z_c) are obtained. At last the transformation matrix combining $\mathbf{Q}_{se}$ and $\mathbf{T}_{ec}$ is found.

Some calibration methods for optical see-through HMD have been proposed. However, most of those require that users hold their head position during the calibration[15]. This constraint is a cause of difficulties of HMD calibration.

Figure 9. HMD calibration.

Obviously our calibration method dose not need this kind of constrains. So this calibration method can be used conveniently.

6. Evaluation of registration and calibration

6.1. Accuracy of the marker detection

In order to evaluate accuracy of the marker detection, detected position and pose were recorded while the square marker with 80[mm] of side length was moved in depth direction with some slants. Figure 10 shows errors of position. Figure 11 shows detected slant. This result shows that accuracy decreases the further the cards are from the camera.

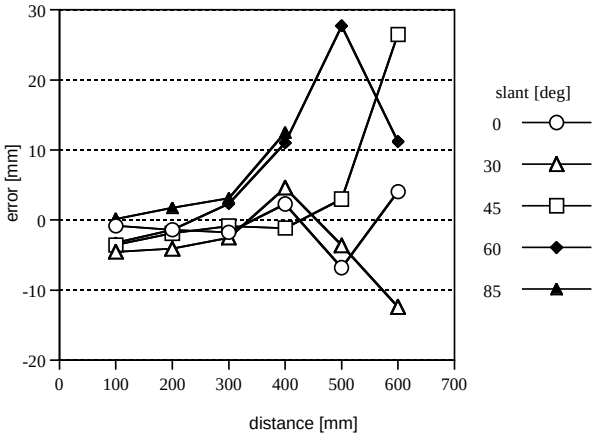

Figure 10. Errors of position.

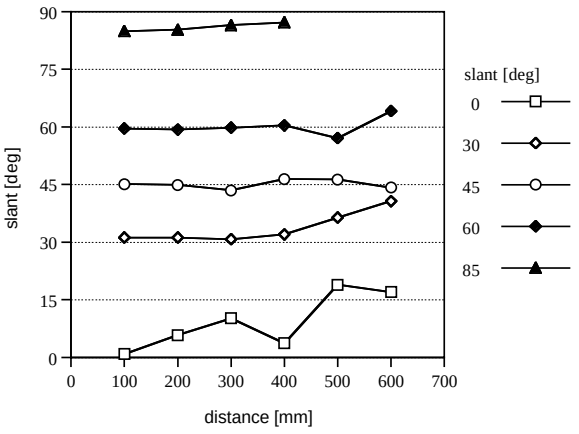

Figure 11. Detected slant.

6.2. Evaluation of HMD calibration

Our HMD calibration method was evaluated by using a program that displays a square of same size as the marker on it. A user with HMD looks at a displayed square on the marker and reports the deviation of a displayed square from the marker. This evaluation was done for 3 tasks:

Task 1: holding the marker.
(Eye-marker distance: 300mm)

Task 2: putting the marker on a desk.
(Eye-marker distance: 400mm)

Task 3: putting the marker far away on a desk.
(Eye-marker distance: 800mm)

Also we had 3 conditions:

(a) Evaluation with standard parameters.
(b) Evaluation with calibrated parameters.
(c) Evaluation with calibrated parameters, but user took off the HMD once after calibration.

Standard parameters mean ones which had been calibrated by another user. 10 times cross-hair cursor fittings were done for each eye. Table 1 shows results of this user testing. This result seems to be good. However, it includes a problem: Focal point of the HMD is on 2-3[m] distance, but a user have to see a virtual object on 300-800[mm] distance. Hereby the user see the virtual object out of focus. This means that reporting a precise deviation is very difficult because of this defocused situation. As a result, test user might report good-will answer. However, we can see the improvement of the registration by using calibrated parameters.

Table 1. Results of user testing.

user	time (min)	condition	task 1 (mm)	task 2 (mm)	task 3 (mm)
A	3	(a)	20	20	20
		(b)	0	0	5
		(c)	3	3	6
B	2	(a)	20	30	35
		(b)	0	5	5
		(c)	0	5	5
C	2	(a)	2	2	2
		(b)	2	2	2
		(c)	0	2	2
D	2	(a)	5	5	10
		(b)	0	0	0
		(c)	0	1	2
E	2	(a)	10	10	10
		(b)	0	0	0
		(c)	1	2	3

7. Conclusions

In this paper we have described a new Augmented Reality conferencing application and the computer vision techniques used in the application. Our computer vision methods give good results when the markers are close to the user, but accuracy decreases the further the cards are from the camera. Also our HMD calibration method which does not require a non-moving user give good results without user's patience. In future, we will improve this AR conferencing prototype and execute user testing for its evaluation as a communication system.

References

[1] A. Wexelblat, "The Reality of Cooperation: Virtual Reality and CSCW", Virtual Reality: Applications and Explorations. Edited by A. Wexelblat. Boston, Academic Publishers, 1993.

[2] C. Carlson, and O. Hagsand, "DIVE - A Platform for Multi-User Virtual Environments", Computers and Graphics, Nov/Dec 1993, Vol. 17(6), pp. 663-669.

[3] J. Mandeville, J. Davidson, D. Campbell, A. Dahl, P. Schwartz, and T. Furness, "A Shared Virtual Environment for Architectural Design Review", CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.

[4] J. Grudin, "Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces", Proceedings of CSCW '88, Portland, Oregon, 1988, New York: ACM Press, pp. 85-93.

[5] H. Ishii, M. Kobayashi, K. Arita, "Iterative Design of Seamless Collaboration Media", Communications of the ACM, Vol 37, No. 8, August 1994, pp. 83-97.

[6] D. Schmalsteig, A. Fuhrmann, Z. Szalavari, M. Gervautz, "Studierstube - An Environment for Collaboration in Augmented Reality", CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.

[7] J. Rekimoto, "Transvision: A Hand-held Augmented

Reality System for Collaborative Design", Proceeding of Virtual Systems and Multimedia '96 (VSMM '96), Gifu, Japan, 18-20 Sept., 1996.

[8] T. Ohshima, K. Sato, H. Yamamoto, H. Tamura, "AR2Hockey:A case study of collaborative augmented reality", Proceedings of VRAIS'98, pp.268-295 1998.

[9] M. Billinghurst, S. Weghorst, T. Furness, "Shared Space: An Augmented Reality Approach for Computer Supported Cooperative Work", Virtual Reality Vol. 3(1), 1998, pp. 25-36.

[10] A. Sellen, "Speech Patterns in Video-Mediated Conversations", Proceedings CHI '92, May 3-7, 1992, ACM: New York , pp. 49-59.

[11] R. Azuma, "SIGGRAPH95 Course Notes: A Survey of Augmented Reality", Los Angeles, Association for Computing Machinery, 1995.

[12] A. State, G. Hirota, D. T. Chen, W. F. Garrett, M. A. Livingston, "Superior Augmented Reality Registration by Integrating Landmark Tracking and magnetic Tracking", Proceedings of SIGGRAPH96, pp.429-446, 1996.

[13] U. Neumann, S. You, Y. Cho, J. Lee, J. Park, "Augmented Reality Tracking in Natural Environments", Mixed Reality - Merging Real and Virtual Worlds (Ed. by Y. Ohta and H. Tamura), Ohmsha and Springer-Verlag, pp.101-130, 1999.

[14] J. Rekimoto, "Matrix: A Realtime Object Identification and Registration Method for Augmented Reality", Proceedings of Asia Pacific Computer Human Interaction 1998 (APCHI'98), Japan, Jul. 15-17, 1998.

[15] G. Klinker, D. Stricker, D. Reiners, "Augmented Reality: A Balancing Act Between High Quality and Real-Time Constraints", Proceedings of ISMR '99, 1999, pp.325-346.