Proc. IEEE Virtual Reality Annual International Symposium
(VRAIS'98), pp.268-275, Atlanta, USA, Mar.14-18, 1998.

AR^{2}Hockey: A Case Study of Collaborative Augmented Reality

Toshikazu Ohshima, Kiyohide Satoh, Hiroyuki Yamamoto, and Hideyuki Tamura
Mixed Reality Systems Laboratory Inc.
6-145, Hanasaki-cho, Nishi-ku, Yokohama 220-0022, Japan
e-mail : {ohshima, ksato, ymmt, tamura}@mr-system.com

Abstract

This paper introduces a collaborative augmented reality (AR) system for realtime, interactive operations. AR enables us to enhance physical space with computer generated virtual space. In addition, the collaborative AR makes multiple participants to simultaneously share a physical space surrounding them and a virtual space, visually registered with the physical one. They can also communicate each other through the mixed space. This paper describes an AR^{2}Hockey (Augmented Reality AiR Hockey) system where players can share a physical game field, mallets, and a virtual puck to play an air-hockey game, as a case study of collaborative AR system. Since realtime, accurate registration between both spaces and players is crucial for the collaborations, a video-rate registration algorithm is implemented with magnetic head-trackers and video cameras attached to optical see-through HMDs. The configuration of the system and the details of registration are described. Our experimental collaborative AR system achieves higher interactivity than a totally immersive collaborative VR system.

Keywords*: Augmented reality, collaboration, registration*

1. Introduction

The augmented reality (AR) technology that seamlessly enhances a physical space with computer generated information has a wider range of applications than conventional virtual reality (VR). Instructing repair or maintenance procedures of a complex machinery [1], manufacturing , in-patient visualization of medical data [2], annotation [3] are some of the application fields in which AR can be utilized. A comprehensive survey of AR is found in [4]. These AR researches, however, have been made mainly on single-user applications so far. New application fields will appear if multiple participants can share a physical space and if we can seamlessly offer a virtual space into the shared physical space [5]. For example, it becomes possible for multiple people to collaborate to design something while exchanging their ideas through virtual objects [6].

AR has two unique features. One is that the virtual space is seamlessly merged with the physical one. The other is that the virtual space as well as the physical one reacts to the physical actions of physical participants. Based on the requirements of these *registration accuracy* and *response time* to physical actions, multi-user collaboration systems with AR are categorized into four groups.

First is the type where both requirements are not strict. This type includes applications such as collaborative information browsing [5, 7]. Secondly, there is a type that requires the accurate positional registration though it does not need high speed response. Collaborative design of mechanical parts and collaborative drawing are the examples of this kind. On the contrary, there is a type that requires the realtime response and needs not the accurate spatial registration. Finally, there exist applications in which both requirements are strict.

There is another factor, *mutual exclusion*, which characterizes the collaborative operations. In the case where participants physically touch each other and simultaneously operate a virtual object, the mutual exclusion against the virtual object operation is necessary. On the contrary, such a mechanism is not necessary if the participants alternately manipulate a virtual object.

This paper describes a collaborative AR system handling an application that requires moderate registration accuracy, realtime response, and no mutual exclusion. The application has the following characteristics.

(a) The target object to be manipulated is in a virtual space.
(b) The target object reacts and moves in virtual space in response to a physical action of participants.

(c) Each participant can watch directly the other participant's action. That is, the participants co-exist inside the area in which they are visible each other.

(d) The response time, the interval between a physical action and its reaction, is less than a threshold so that participants feel the illusion that physical and virtual objects seem to co-exist.

Such visibility of actions and fast response contribute to the feeling of co-presence.

Section 2 of this paper studies about characteristics of AR system, and finds out some problems to establish collaborative AR system. Section 3 explains about *AR²Hockey* (AR AiR Hockey) system that was made as a case study of the collaborative AR system. In this example, the optical see-through HMD augments physical world with virtual space. Section 4 studies about problems of realtime registration. The video rate registration is implemented by using Polhemus' 3D sensors and video cameras attached to the HMDs. Section 5 reports about observations on the system. Conclusion and future directions are discussed in Section 6.

2. Requirements of Collaborative AR

This section studies characteristics of conventional AR and requirements for collaborative AR. We focus on three factors here, which are augmentation, registration, and rendering.

Augmentation

Two choices are available for the augmentation of physical world with virtual one: video-based and optical based. Both approaches can be built using monitor based configuration and used in the collaborative AR. A “responsive workbench” style system works wonderfully for a collaborative AR application [8]. We omit that configuration in this paper and assume a HMD style.

Video see-through HMD works using a closed-view HMD and one or two cameras attached to the HMD. The video from the cameras is combined with computer generated images and the combined video is displayed on the HMD. This configuration is often used in the applications where accurate registration is necessary. This is because 1) digitized video images are available for additional registration methods, 2) delay, brightness, and contrast between the two spaces can be easily matched in the video see-through [4].

On the other hand, optical see-through HMD uses optical combiners so that users can see the physical world through glasses and simultaneously look at an image displayed on the HMD monitor. Since the physical world is seen directly, there is no time delay to see it while at least one frame time is delayed in the video see-through type. In addition, the resolving power of physical space is only limited by the resolution of human fovea, not the display device of HMD. The time lag, however, between physical world and virtual one is crucial.

In collaborative AR, interactions are not limited to physical to virtual and vice versa. Physical to physical interaction between participants is also important. In that case, scenes of physical space for all the participants should be synchronized and the time delays should be minimized in order to match the kinesthetic and visual systems. Thus we believe that optical see-through HMD is more suitable for collaborative AR than video see-through type.

Registration

In AR applications, it is necessary for virtual objects and physical space to be visually registered with respect each other. Virtual objects appear to be floating if the registration is not accurate enough. Such phenomena are mainly divided into three factors: 1) static error or positional misalignment, 2) dynamic error or time lag, and 3) rendering error or difference of image qualities. We discuss the static error here because it is most essential to be resolved.

In the optical see-through AR, the problem of positioning can be translated into the problem to decide the 3D position of a user's viewpoint. To decide the viewpoint, it is generally measured by 3D sensors such as magnetic, ultrasonic or gyroscopic sensors. Since these sensors can not always give us enough accuracy required, the error in these sensors causes positional misalignment. In the video see-through AR, on the other hand, the problem of positioning can be translated into the problem to seek the position of a camera, and the method developed in the computer vision research can be utilized to solve the problem. We can also utilize the method to register the images of virtual objects directly on images captured by the camera [9]. This method, however, has some problems such as lack of reliability or impossibility to present timely while it's easy to realize accurate positioning since we can directly handle the positioning error on the image.

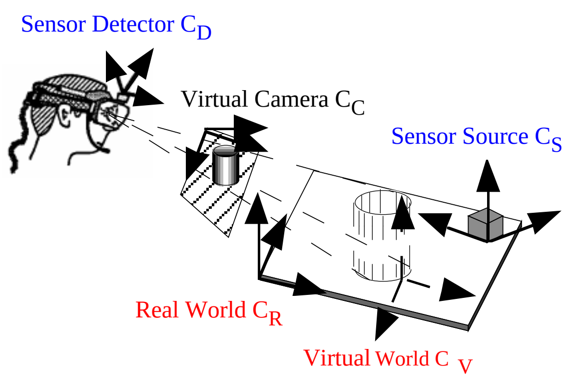
Some trials are recently reported that uses both of 3D sensors and the image information to realize accurate positioning. Bajura and Neumann [10] has suggested a method to compensate positioning error caused by errors in magnetic sensors by using image information in the video see-through AR. State et al. [11] has expanded this method and has suggested a method to compensate vagueness in position presumption from image information using sensor information.

These methods give the collaborative AR system good measures of positional registration. However, there is a problem that the registration is done in a restricted condition or a certain limited range of space. Generally, it becomes an intolerable limitation in a practical collaborative system and we believe that we have to eliminate such restrictions. Note that there is no application of the video-based registration algorithm into the optical see-through AR.

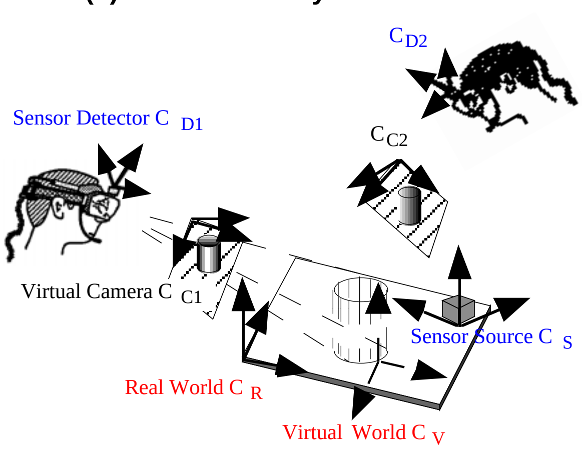
Figure 1 (a) shows the typical coordinate systems used in simple AR. The registration is the process that transforms the viewing matrix C_C. In collaborative AR, the physical space and virtual space are shared by all the participants. Thus the coordinate system C_R and C_V exist in the system and shared by the participants. On the other hand, the coordinate system C_C and C_D that relate to the viewing transformations exist for each participant. This situation is illustrated in Fig. 1 (b). Thus the registration can be implemented independently for each participant.

Rendering

For AR applications, rendering should be fast enough so that the dynamic error, the lag between the physical space and virtual one, is minimized. In addition, the rendered images of virtual object with high photo-reality are preferable in order that rendering error may be



(a) Coordinate systems in AR



(b) Coordinate systems in collaborative AR

Figure 1: Coordinate systems

negligible. Since such a rendering system is not likely to appear anytime soon, there is a little attention to the rendering in AR research. Thus rendering engines specialized for realtime graphics are normally used.

Because the participants share the physical space around them and gather together in a small area in the collaborative AR, the problem of synchronization must be considered as another factor. In order to interact each other through the mixed space, it is ideal to see the two spaces totally synchronously between the participants. If the optical see-through HMDs are used, the physical space are completely synchronized in that sense. However, there should be another mechanism to synchronize the images of virtual space. This is not the case of collaborative VR or networked reality in which participants and rendering engines are placed at physically separated locations.

3. System Configuration

This section describes the configuration of a collaborative AR system, called AR^2Hockey. Air hockey is a game in which two players hit a puck with mallets on a table and shoot it into goals. In our AR^2Hockey, the puck is in a virtual space. This simple application challenges to the following problems of the collaborative AR. Firstly, more than two persons share a single physical and a virtual space. Secondly, since the puck moves fast, the response time becomes severe and the synchronization problem should be solved. Thirdly, since the virtual puck is hit by an physical hand, the positioning error must be minimized. Finally, since a player can see his opponent's most subtle movements through the HMD, it becomes more than a simple VR.

Figure 2 (a) shows the scene of playing AR^2Hockey and (b) is an image seen through the HMD while the system is operating.

3.1. Hardware components

According to the discussion in Section 2, we adopt the system configuration described below.

HMD

This system uses optical see-through HMDs shown in Fig. 3 [12]. The HMD contains an LCD of 180 thousands pixels and two prisms for each eye. One prism is used to lead images displayed on the LCD to the eye. This prism has two off-axial reflective surfaces. To correct the off-axial aberrations, the aspherical surface without rotational symmetry is used in this prism. Attaching the compensation prism to the outside of the prism, good see-through view is achieved. This HMD gives us 34 degrees of horizontal view angle and 22.5 degrees of vertical one.

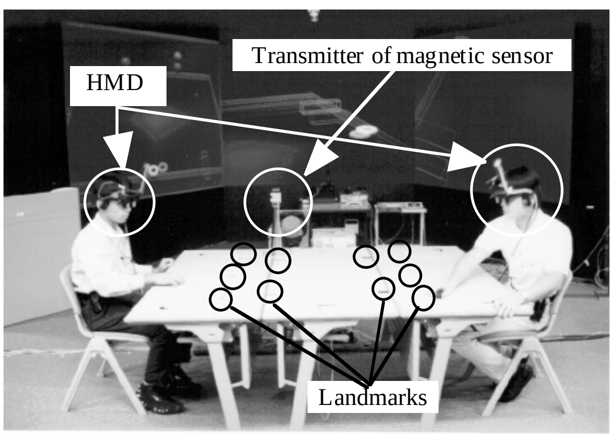
Trackers

The HMD uses a magnetic sensor (Polhemus' Fastrak) to measure the player's viewpoint. Since this positional and orientational sensor dose not have enough accuracy to produce images without notable displacement, we have placed a small color CCD camera (ELMO) having 45 degrees of view angle near the right eye position of HMD. This camera detects markers in the physical space in order to compensate the error of magnetic sensor. See Section 4 for the details of registration algorithm.

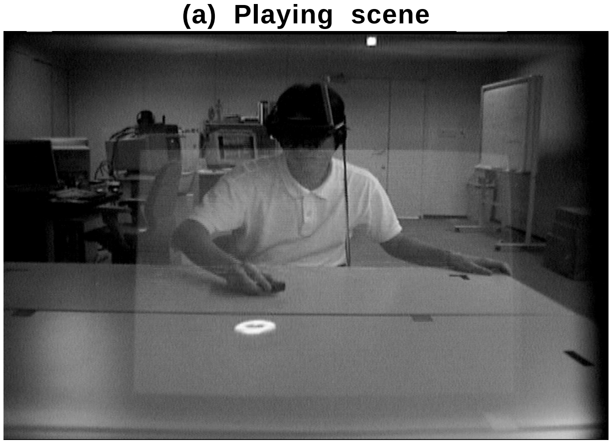
Each player holds a mallet as a physical device to hit a virtual puck. The mallet is a simple device having infrared LEDs. The position of the mallet is tracked on the image captured by a CCD camera set directly above the table. In our AR^{2}Hockey, movement of the mallets is constrained on two dimensional plane, but 3D tracking may be required depending on applications.

Computers

The AR^{2}Hockey system uses three SGI O2s for processing two video images from cameras attached to HMDs and a mallet tracking process. In addition, one SGI ONYX2 (with eight CPUs and three InfiniteReality



(a) Playing scene



(b) Player's view

Figure 2: Playing scene of AR^{2}Hockey

Figure 3: Optical See-through HMD

graphic pipelines) handles head position tracking, image and sound rendering, and the total system control. All the computers in the system proceed processings while communicating with each other over the Ethernet network. See Section 3.2 for the flow of the process.

By making only one super-graphic workstation to process all the rendering, images given to the participants can completely be synchronized and solve the synchronization problem described in Section 2. Currently this system distributes other processes to three computers over the Ethernet, however, it is ideal to do all processes by only one graphics workstation since the time lag on the network communication may be a problem.

3.2. Process flow

Figure 4 shows the process flow of this system. As shown in the figure, the process is composed of six types of sub processes and one master process. Three sub processes, marker tracking, registration and rendering are invoked for each player.

The four tracking processes that drive input devices work asynchronously. This means that they proceed independently from the master process and parallel to each other. On the other hand, the registration, the space management and the rendering processes are synchronous to the master process. By configuring the system in this way, it becomes possible to reduce the effect caused by the difference of the sampling rates of 3D sensors, video capturing rate and the rendering rate. This effect directly influences the time lag of the system.

Head-tracking Process

Positions of two players' heads are measured by magnetic sensors attached to the HMDs. The head positions are measured at up to 50 Hz depending on the characteristics of Fastrak system. Note that two sensors for two HMDs are used in the system. A corresponding subprocess tracks a head position at this rate and sends the data to the registration process.

Marker-tracking Process

An image from a player's viewpoint is taken from the CCD camera mounted on the HMD. From the image, the system extracts a position of marker on the table. The marker position is sent to the registration process at about 30 Hz. However, there is a latency of about 40 ms to capture and process an image, the data sent to the registration process delays at that interval.

Ten small square-shaped markers are placed on the table as shown in Fig. 2 (a). Each marker has one of two colors, red or green, and the color specifies which participants of the two the marker is used for.

A maker is extracted by a simple image processing. The process decides that a point is a marker when there is a point near the marker position of the previous frame and the point has an intensity of the predefined color over a certain threshold. If there is not such a point, the system scans the entire image and detects the point having the highest color intensity as a marker.

Registration Process

This subprocess makes registration in response to the request from the master process based on the stored head tracking and marker tracking data and sends corrected head position and orientation data to the master process.

As described, the latest data stored in this process does not correspond to the current status of the physical world. There is some time delay. Thus the registration process records the timestamp that indicates when the data is updated by the tracking process. Based on the timestamp and data itself, this process first predicts the current head position, orientation, and marker position using the second order prediction algorithm described in Eq. (1).

$$\hat{p} = p_t + \frac{1}{2} a_t \Delta t^2 + v_t \Delta t \qquad (1)$$

where $\hat{p}$ is the predicted value,
p_t is the latest recorded data,
v_t, is the velocity at the time p_t is recorded,
a_t is the acceleration at the time p_t is recorded,
Δt is the elapsed time from the moment p_t is recorded.

Since the second order algorithm considers the velocity and acceleration of the head and marker, this process keeps track of three successive data, that is p_t, p_{t-1}, and p_{t-2}.

The registration algorithm is applied to the predicted data. See the next section for the registration algorithm.

Mallet-tracking Process

This process measures 2D positions of mallets on the table. It is implemented by a simple image processing as the marker tracker and sends mallet position data to the space management process at about 30 Hz with a latency of about 40 ms.

Space Management Process

This process manages the state of the game such as puck position, speed and a score and updates the game

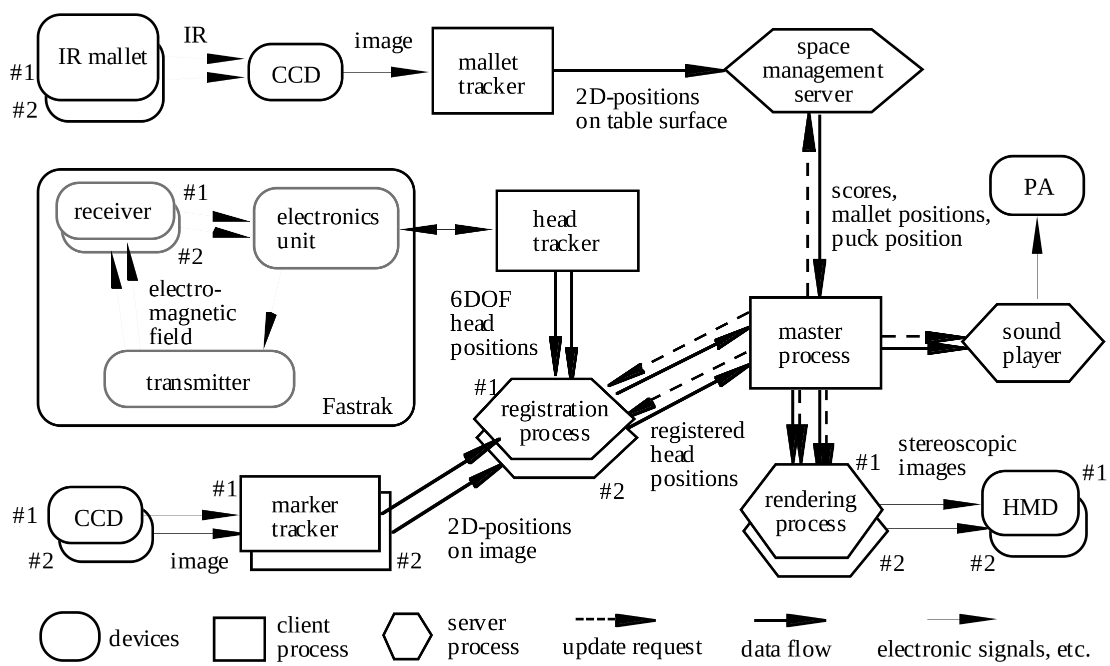


Figure 4: Process diagram

status by predicting recent mallet positions based on the stored mallet data in response to the request from the master process. This process also generates sound effects when it updates game status.

Rendering Process

This process is synchronous to the master process and generates a set of stereoscopic images. The process obtains the required data from the master process and displays rendered images to the HMD.

Master Process

This is the process which coordinates, controls the other processes and creates the system highly cohesive. The process requests corrected position and orientation data from the registration process, and the game state from the space management process at exact rate of 36 Hz. Then it sends these data to the rendering process for rendering.

4. Static Registration Algorithm

This section describes the registration algorithm implemented in the system. This is the vision-based registration algorithm and corrects the value of the sensor output based on the difference between the marker position captured by the camera and the marker position calculated from the sensor output.

4.1. Vision-based registration

Correcting positioning error using only one marker or landmark is a simple and effective method for the registration. Thus we chose this method and implemented a registration algorithm based on the registration proposed by Bajura [10].

Figure 5 shows the basic theory to correct positioning error using one landmark. The discussion below assumes that all the inner camera parameters are already known and an image is captured by an ideal capturing system without any distortion.

In the figure, let C, I, and Q be the camera position, the image plane, and the landmark position in the physical 3D space, respectively. For those C, I, and Q, the projected landmark position on the image Q' is determined as the point at which the line l_Q connecting C and Q intersects the image plane I. On the other hand, The landmark position P in the camera coordinate system and the corresponding position on the image P' are calculated based on the 3D sensor data. These P and P' can be thought as the landmark in the virtual space and its corresponding image position. Ideally, point Q and P coincide in the 3D space. That is , the projected position Q' and P' coincide on the image plane. This is, however, usually not true because of the 3D sensor error.

The correction is done by translating the virtual space coordinates so that the corrected predicted observed coordinate of the landmark P' coincides with the point Q'. This is done by translating objects in the virtual space by

$$v = n(v_1 - v_2) \qquad (2)$$

where n is a scale factor derived by

$$n = \frac{|CP|}{|CP'|} \ . \qquad (3)$$

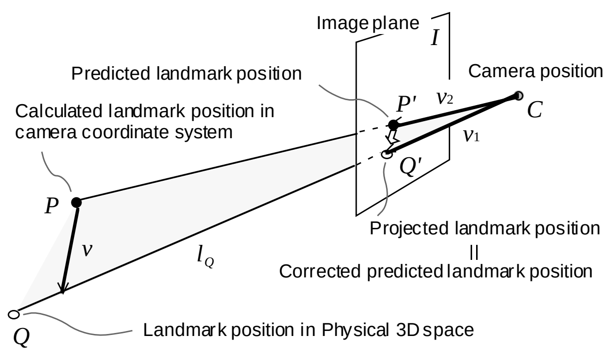


Figure 5: Registration with one landmark

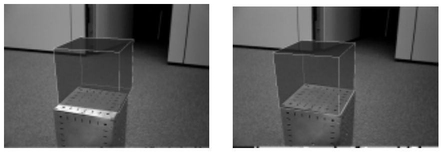
Since this method only registers the positions on 2D image plane, the three dimensional positions may not coincide correctly even after the registration. This method, however, is still effective if the sensor error is not so much. Moreover, the calculation cost of this method is inexpensive and suitable to the realtime process required to the system such as our collaborative AR.

4.2. Widening registration space

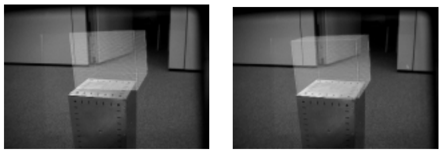
Since the method described above requires a marker always reside in the captured image, the registration is limited to only a small area of the physical space, while the collaborative operation requires broader registration area. Thus, we have expanded the method as shown below.

The orientation of the camera can roughly be measured from the 3D sensor output. By using this information as a guide to identify the marker to be used among multiple markers placed in the physical space, it becomes possible to establish merged process for broader area.

Suppose multiple markers are placed in the physical space and the three dimensional positions of these markers P_i $(i = 1...N)$ are already known. Here N is the number of markers. For P_i, let the point P_i' denotes the calculated coordinate of the marker on the image derived from the sensory output. When a marker is detected at a coordinate



(a) Video see-through



(b) Optical see-through

Figure 6: Registration results. (The lefts show merged images without registration. The rights are the registered results.)

of Q', we have to decide which marker is detected among P_i. As the target marker, we choose the marker P_K having the minimum value of the following evaluation e_i:

$$e_i = \left|P_i'Q'\right| \quad (i = 1 \ldots N) \tag{4}$$

After the marker is decided in this way, the same method as described above can be used as it is.

In the marker tracking process, only one marker is tracked even when more than one marker are in the image and another marker in the image is only used when the process fails to track or the marker tracked goes out of the image.

By switching markers, displayed virtual objects may shift abruptly. This is because the abrupt change of the corrected value occurs between the previous frame and the current frame. To avoid this kind of abrupt change, the following linear interpolation method is incorporated in our system.

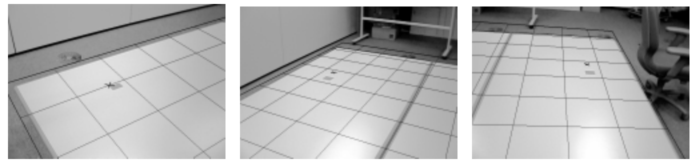
Let v_t and $\hat{v}_t$ denote the value of Eq. (2) and the new recalculated value at the current frame t. The following equation recalculates the value.

$$\hat{v}_t = \alpha\hat{v}_{t-1} + (1-\alpha)v_t \tag{5}$$

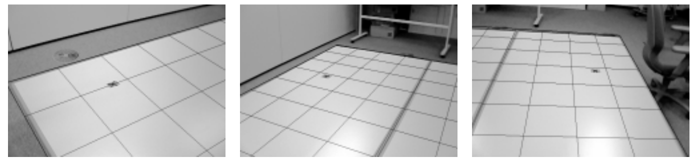
Here α $(0 \le \alpha < 1)$ is a constant that denotes influence of the past information. Past information is not used when the α equals to 0. The abrupt shift of the virtual objects can be avoided by setting the value of α to an appropriate value.

4.3. Applying to optical see-through AR

In order to apply the above method to our optical see-through HMD, it is ideal to nullify the parallax between the camera attached to the HMD and the eyes seeing images in the HMD. We are currently considering an optical see-through HMD with parallax free video capturing function. Since such kind of HMD is not



(a) Without registration



(b) With registration

Figure 7: Effects of registration

realized yet, the registration method described above is used as a second best.

In the registration method, parameters for rendering virtual space are corrected so that the marker in the virtual space is placed on the line between the center of the camera attached to the HMD and the marker in the physical space. Therefore, the position of the physical marker seen through the HMD is not always coincides the position of the virtual marker projected onto the HMD using this correction value. This is because there is some distance, parallax, between the center of the camera and the eye looking the HMD. Though it is, we can expect the effect of the correction if the distance is small enough comparing to the distance between the eye and the marker.

4.4. Effects of registration

Figure 6 shows results with this registration algorithm. In this experiment, we use a special setup in order to evaluate quantitatively the effect. In the figure, a marker is placed at the left lower corner of the upper plane of a physical cube. Then virtual cube that is the same size as the physical one is overlaid. In the case of video see-through, the mean positional difference between the physical cube corners and virtual ones on the image of 640 $\times$ 480 pixels is 42.1 pixels without registration. This value decreases to 9.8 pixels with the registration method above. In the optical see-through case, we have similar results as shown in the figure.

Figure 7 shows the effects for a wide registration area. The registration area is the game filed on the table with the size of 120 cm $\times$ 150 cm, and the landmarks are placed as the Fig. 2 (a) so that at least one landmark is captured by the camera attached to HMD while participants play game. Figure 7 (a) and (b) shows scenes augmented by the virtual game fields without and with the registration. In the figures, the virtual game fields are shown as wire-frames. Through the experiments, we confirmed that our static registration algorithm gives us enough accuracy and speed for the game of AR^{2}Hockey.

As regards speed, the quantitative evaluation is not available. People can play the game naturally as described in the next section. Thus the response time seems to be kept within a satisfactory range.

5. Observations

Over 250 people played this game system. Most participants reported that they played the game in much the same way as the actual game except for the head movement. Whereas human beings have over 100 degrees of visual field, the current HMD has only 34 degrees of horizontal view angle. Thus participants have to move their head so that they can track the puck in sight. It is also reported by some people that the virtual puck sometimes floats around. This is a case when a player moves his/her head so fast that the prediction Eq. (1) fails.

During the trials, we changed the augmentation level as follows:

Immersive VR: Only the cyberspace images are displayed to players. The cyberspace consists of the puck, mallets, and game field. Players cannot see any physical space in this case.

AR#1: The puck and mallets are displayed as virtual objects over the physical space. The virtual mallets can guide the users so that they can guess the relationship between the physical hands and virtual mallets.

AR#2: In cyberspace, there is a puck. The other information is perceived through the physical space in this case. This is the final goal of our collaborative AR.

Through the experiments, we understand that players often failed to hit the puck when the virtual mallet goes out of their visual field in the case of immersive VR and AR#1. In the AR#2, such mistake is observed less frequently. It is natural to consider that people feel difficulty in moving their hands when the displayed mallet moves out of the visual field since the mallet acts as a bridge between the physical space and the cyberspace. In AR#2, there is no such conflict.

6. Conclusions and Future Studies

In this paper, we have introduced a collaborative augmented reality system. The collaborative AR makes multiple participants to simultaneously share physical space surrounding them and virtual space, communicate each other through mixed spaces. We have implemented an AR^{2}Hockey system where two players can share physical game field, mallets, and a virtual puck to play air-hockey game. Since realtime, accurate registration between both spaces is necessary, we have developed a video-rate static registration algorithm with magnetic head-trackers and video cameras attached to optical see-through HMDs.

The experiments we have carried out show that the system is stable and participants can play naturally as the actual game.

Since the registration algorithm implemented is simple and primitive, we are studying more sophisticated and fast registration methods using this system. Time critical system configuration and the performance evaluation of collaboration work are among other aspects to be investigated further.

Acknowledgments

The authors would like to thank the people of Mixed Reality Systems Laboratory Inc. for their cooperative works and useful discussions.

References

[1] S. Feiner, B. MacIntyre, and D. Seligmann, “Knowledge-based augmented reality,” *Comm. ACM*, Vol.36, No.7, pp.52-62 (1993).

[2] M. Bajura, H. Fuchs, and R. Ohbuchi, “Merging virtual reality with the real world: Seeing ultrasound imagery within the patient,” *Computer Graphics*, Vol. 26, No.2, pp.203-210 (1992).

[3] E. Rose et al., “Annotating real world objects using augmented reality,” *Proc. Computer Graphics International '95*, pp.357-370 (1995).

[4] R. Azuma, “A survey of augmented reality,” *Presence*, Vol.6, No.4, pp.355-385 (1997).

[5] M. Billinghurst, S. Baldis, E. Miller, and S. Weghorst, “Shared space: Collaborative information spaces,” *Proc. HCI International '97*, pp.7-10 (1997).

[6] J. Rekimoto, “TransVision: A hand-held augmented reality system for collaborative design,” *Proc. VSMM '96*, pp.85-90 (1996).

[7] A. Fuhrmann, H. Loffelmann, and D. Schmalstieg, “Collaborative Augmented Reality: Exploring Dynamical Systems,” *Proc. Visualization '97*, pp.459-462 (1997).

[8] M. Agrawala et al., “The two-user responsive workbench: Support for collaboration through individual views of a shared space,” *Proc. SIGGRAPH '97*, pp.327-332 (1997).

[9] K.N. Kutulakos and J.R. Vallino, “Affine object representations for calibration-free augmented reality,” *Proc. VRAIS '96*, pp.25-36 (1996).

[10] M. Bajura and U. Neumann, “Dynamic registration correction in video-based augmented reality systems,” *Computer Graphics and Applications*, Vol.15, No.5, pp.52-60 (1995).

[11] A. State et.al., “Superior augmented reality registration by integrating landmark tracking and magnetic tracking,” *Proc. SIGGRAPH '96*, pp.429-438 (1996).

[12] H. Hoshi et al., “Off-axial HMD optical system consisting of aspherical surfaces without rotational symmetry,” *SPIE*, Vol.2653, pp.234-242 (1996).