

Research Article

Visual Enhancement for Sports Entertainment by Vision-Based Augmented Reality

Yuko Uematsu and Hideo Saito

School of Science for Open and Environmental Systems, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Koyoku-ku, Yokohama 223-8522, Japan

Correspondence should be addressed to Yuko Uematsu, yu-ko@ozawa.ics.keio.ac.jp

Received 1 October 2007; Revised 26 June 2008; Accepted 6 July 2008

Recommended by Adrian Cheok

This paper presents visually enhanced sports entertainment applications: AR Baseball Presentation System and Interactive AR Bowling System. We utilize vision-based augmented reality for getting immersive feeling. First application is an observation system of a virtual baseball game on the tabletop. 3D virtual players are playing a game on a real baseball field model, so that users can observe the game from favorite view points through a handheld monitor with a web camera. Second application is a bowling system which allows users to roll a real ball down a real bowling lane model on the tabletop and knock down virtual pins. The users watch the virtual pins through the monitor. The lane and the ball are also tracked by vision-based tracking. In those applications, we utilize multiple 2D markers distributed at arbitrary positions and directions. Even though the geometrical relationship among the markers is unknown, we can track the camera in very wide area.

Copyright © 2008 Y. Uematsu and H. Saito. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Augmented reality (AR) is a technique for overlaying virtual objects onto the real world. AR has recently been applied to many kinds of entertainment applications by using vision-based-tracking technique, such as [1–3]. AR can provide users with immersive feeling by allowing the interaction between the real and virtual world.

In these kinds of AR entertainment applications, virtual objects (world) generated with computer graphics are overlaid onto the real world. This means that the real 3D world is captured by a camera and the virtual objects are superimposed onto the captured images. By seeing the real world through some sort of displays, the users find that the virtual world is mixed with the real world and they can control the virtual world as well as the real world.

In such AR applications, the users carry a camera and move around the real world. Therefore, the pose and the position of the moving user's camera should be obtained so that the virtual objects can be appeared at correct position in the real world captured with the camera. Such camera tracking should also be performed in real time for interactive operations of the AR applications.

Vision-based camera tracking for such AR application is one of the popular research areas because the vision-based method does not require any special device except cameras, in contrast with sensor-based approach. For making the vision-based tracking robust and running in real time, marker-based approach is a reasonable solution, so we focus on marker-based approach. Especially, "AR-Toolkit" [4], which is using 2D square marker for the camera tracking, is a very popular tool for simple online AR applications that follow a marker-based approach. The camera's position and pose can be estimated in real time by using the 2D square markers.

This paper presents two AR applications: AR Baseball Presentation System and Interactive AR Bowling System. Users use these applications on the tabletop in the real world by using a web camera attached to a handheld monitor as shown in Figure 1.

AR Baseball Presentation System is an observation system of a virtual baseball game. Users place a real baseball field model on the tabletop and input a baseball game history (scorebook) that they want to watch into the system. Then, they can watch the game by replaying with virtual baseball

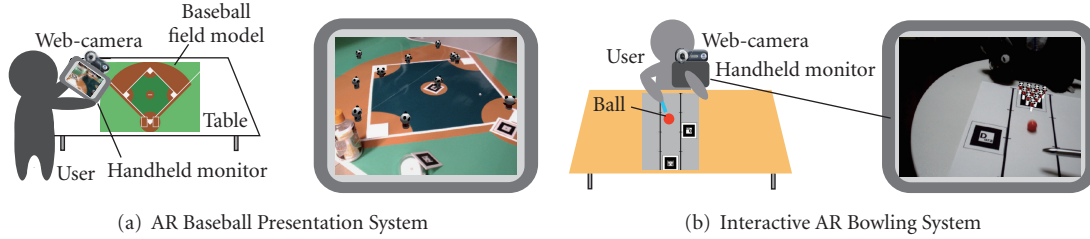


FIGURE 1: Our proposed AR applications.

players on the field model in front of them. On the field model, 2D markers are placed for registration of the virtual players. Therefore, the users can watch the game from their favorite viewpoints around the field. This system focuses on visualizing the game by using a scorebook data, so that the user can understand the main game point. Thus, the detail part of the game such as player's gesture is not replayed. In this system, the virtual players are generated as a cartoon character instead of a human player. There is a limit even if a human player is made in great detail by CG. An increase in the number of polygon in CG is also big problem for real-time processing. Therefore, we decide to use a funny and friendly character.

With interactive AR Bowling System, users can enjoy the bowling game by rolling a real ball down a real bowling lane model placed on a tabletop in the real world. On the lane model, there are virtual pins generated with CG. They knock down the virtual pins by rolling the real ball. Touching and rolling the real ball provide a sort of tangible feeling in this system. It is well known that a tangible interface enhances the reality of communication [5–8]. Because of placing some markers on the lane model, the users can watch the lane and pins from free view points.

For registration of virtual objects such as the virtual players or the virtual pins, the motion of the user's camera is estimated by multiple 2D markers. In our applications, the multiple markers can be arranged at arbitrary positions and directions in the real world. Therefore, the users can start our applications only with free-arranging the markers. These applications are based on the multiple marker-based online AR system [9].

2. Related Work of Marker-Based AR Applications

There are some related works using 2D markers for AR applications. Henrysson et al. have proposed the AR tennis application which is used on the tabletop tennis court model [10]. On the tabletop tennis court, a few 2D markers are drawn for estimating the motion of user's camera. In their application, since a ball is a virtual object and user's position does not move a lot, the 2D markers are easily detected by the user's camera. On the other hand, in our baseball application, the users move around the baseball field to watch the baseball game from favorite view points. Therefore, a lot of markers should be arranged in the real world. Moreover, in our bowling application, since the ball is a real object, the

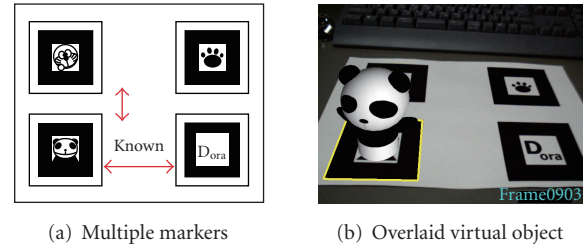


FIGURE 2: Usual multiple marker-based registration of a virtual object.

markers may be occluded by the rolling ball. Therefore, the markers should be arranged not only on the table plane but also in various directions.

Multiple markers are usually used by aligning themselves at measured intervals as shown in Figure 2 because the geometrical relationship of the multiple markers must be known [11–15]. In [15], they need the position and pose of a square marker and the position of a point marker in advance. In [14], they proposed marker-less registration method by setting up a learning process. In the learning process, however, the markers' geometrical information is required for learning the markers. In most cases, the task for the measurement of such information is implemented manually. However, this task is very time-consuming and not sufficiently accurate. Kotake et al. [16] proposed a marker-calibration method combining multiple planar markers with bundle adjustment. Although they do not require a precise measurement of the markers, they need a priori knowledge of the markers such as qualitative information to compute markers' geometrical information from a set of images by bundle adjustment, that is, multiple markers are coplanar.

In contrast, our registration method can freely distribute the multiple markers at arbitrary positions and directions. The geometrical relationship of the markers can be automatically estimated by constructing a 3D projective space which is defined by projective reconstruction of reference images. Through the projective space, then the geometrical arrangement of the marker planes is recovered in 3D. Therefore, we need not to manually measure the distance between the markers in advance. This algorithm is quite suitable for the AR applications when the users move around the real world and the markers may be occluded.

In this paper, we explain the algorithm of registration with multiple markers as described in Section 3. Then, AR Baseball Presentation System and AR Bowling System are introduced in Sections 4 and 5, respectively.

3. Registration Using Multiple Markers

In this section, we explain the algorithm of the registration method in the Multimarker-Based Online AR System [9]. This algorithm is based on [17].

Figure 3 shows a flowchart of the registration method. This registration method can be divided into two stages. At the first stage, the geometrical relationship of the markers is automatically estimated. For the estimation, a 3D projective space, which is a 3D virtual space, is defined by projective reconstruction of two reference images. The reference images are automatically selected from some candidate images. In our registration method, the geometrical relationship of the markers is represented as a transformation matrix called T_i^{WP} which relates each marker i and the projective space. These transformation matrices are computed once in advance.

At the second stage, a projection matrix P_i^{PI} from each marker i to the input image is computed. Those projection matrices and the transformation matrices, which are computed at the first stage, are integrated into projection matrices P_i^{PI} by (1), respectively,

$$P_i^{PI} = P_i^{WI}(T_i^{WP})^{-1}. \quad (1)$$

These projection matrices are based on the marker i and project the projective space onto the input image. Moreover, those P_i^{PI} are integrated into one projection matrix P^{PI} by least-square method. Then virtual objects described in the projective space coordinate system are overlaid onto the input image by using the integrated projection matrix. These processes of the second stage are performed at every frame.

3.1. 3D Projective Space

A 3D projective space is constructed for estimating the geometrical arrangement of multiple planes placed at arbitrary positions and poses. The projective space is defined by projective reconstruction of two images which are captured from two different view points and called reference images. As shown in Figure 4, a 3D space P - Q - R is defined as a 3D projective space, which is projected onto the reference image A and B by following equations:

$$\begin{aligned} [u_A v_A 1]^T &\simeq P_A [PQR1]^T, \\ [u_B v_B 1]^T &\simeq P_B [PQR1]^T, \end{aligned} \quad (2)$$

$$P_A = [I \mid \mathbf{0}], \quad P_B = \left[-\frac{[\mathbf{e}_B]_{\times} F_{AB}}{\|\mathbf{e}_B\|^2} \mid \mathbf{e}_B \right], \quad (3)$$

where, $[u_A, v_A, 1]^T$ and $[u_B, v_B, 1]^T$ are homogeneous coordinates of 2D points in the reference images, and $[P, Q, R, 1]^T$ is also homogeneous coordinates of a 3D point in the projective space. F_{AB} is a fundamental matrix from the image A to

B, \mathbf{e}_B is an epipole on the image B, and $[\mathbf{e}_B]_{\times}$ is the skew-symmetric matrix [18].

Since the projective space is defined by projective reconstruction of the reference images, the accuracy of F_{AB} is important and is depending on the combination of the reference images. In this system, two reference images which have most accurate F_{AB} are automatically selected. The details will be described in next section.

3.2. Automatic Selection of Reference Images

The projective space is defined by the projective reconstruction of two reference images. Therefore, the fundamental matrix between the reference images is important to construct the accurate projective space. We introduce automatic selection algorithm of the reference images. The detail is shown in Figure 5.

First, the object scene is captured for a few seconds by a moving camera. This image sequence in which all the markers should be included becomes the candidate of the reference image. When two images are selected from the candidate images, projection matrices based on the markers in the selected reference images are computed by using the algorithm of [4], where P_{A_i} and P_{B_i} are the projection matrices which project marker i to the selected reference image A and B, respectively. Using each pair of the projection matrices, a fundamental matrix based on marker i is computed as in the following equation:

$$F_{AB_i} = [\mathbf{e}_{B_i}]_{\times} P_{B_i} P_{A_i}^{-}, \quad (4)$$

where $P_{A_i}^{-}$ represents the pseudoinverse matrix of P_{A_i} [18]. Then, one fundamental matrix is selected as F_{AB} which has the smallest projection error:

$$\text{error} = \mathbf{m}_B^T F_{AB} \mathbf{m}_A, \quad (5)$$

where \mathbf{m}_A and \mathbf{m}_B are corresponding points in the selected reference images.

When a projective space is temporarily constructed by the selected F_{AB} from (3), T_i^{WP} between each marker i , and the projective space is computed. Then, P_i^{PI} are computed and integrated into one projection matrix P^{PI} . Then, we compare these two projected coordinates \mathbf{x}_i , \mathbf{x}'_i :

$$\mathbf{x}_i = P_{B_i} X_W, \quad \mathbf{x}'_i = (P^{PI} T_i^{WP}) X_W. \quad (6)$$

Although these two coordinates should be equal, if the combination of the two reference images is not reasonable, they will be different. In such a case, we return to the phase of selecting a pair of temporary reference images. We iterate these processes until every difference of \mathbf{x}_i and \mathbf{x}'_i based on plane i becomes smaller than a few pixels. In the experiments, we decide the threshold as 3 pixels.

Even if the number of markers increases, only the time of computing the transformation matrices (F_{AB_i} , P^{PI} , T_i^{WP}) is increased and the computation time is very short. Therefore, it is not a time-consuming process. The times of iteration is mainly decided by the number of the candidate reference images. When using 100 candidate reference images, the processing time of selecting reference images using 8 markers also took around 60 seconds as well as using 4 markers.

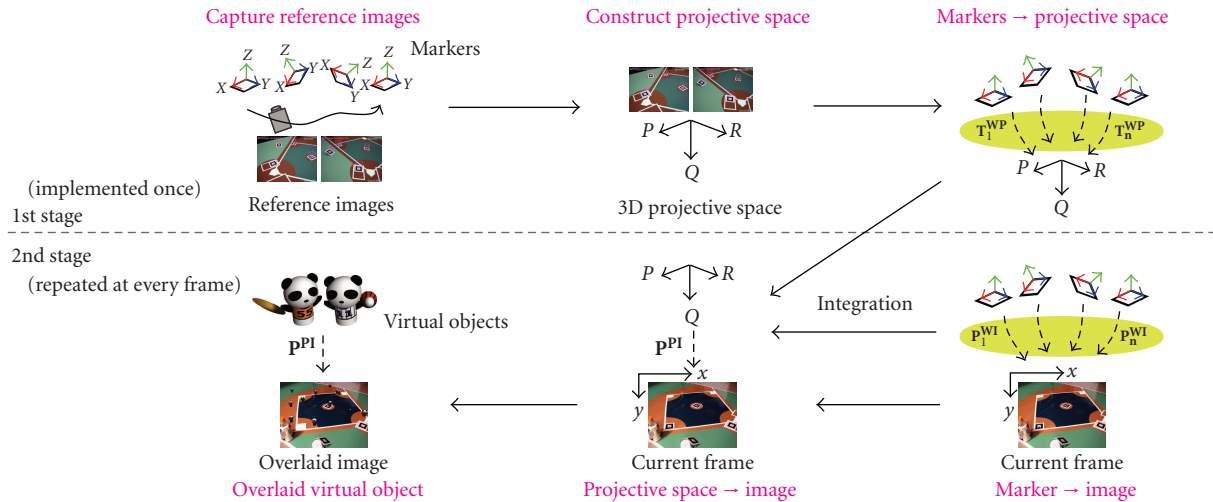


FIGURE 3: Overview of the registration method with 3D projective space.

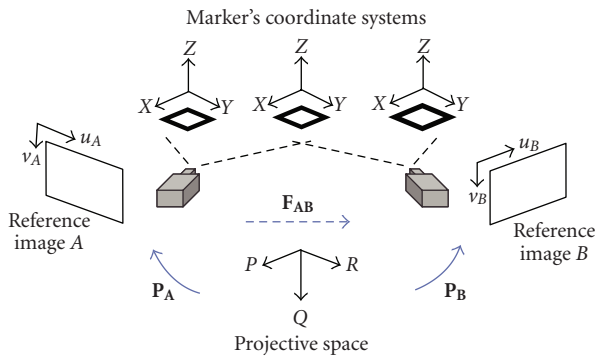


FIGURE 4: 3D projective space defined by projective reconstruction of the reference images.

4. AR Baseball Presentation System

AR Baseball Presentation System allows users to watch a virtual baseball game on the tabletop field model in the real world via a moving web camera attached to a handheld monitor. The virtual baseball game scene is synthesized with 3D CG players. These players are overlaid on the real field model. The users can interactively change their view points as their likes by applying the algorithm described in Section 3.

This system visually replays the baseball game which was previously played in the other place by a scorebook data, in which the game history they want to know is described. In contrast with the usual way to know the game history, such as watching the captured video or reading the recorded scorebook, our AR system can provide the users with much realistic sensation as an entertainment application.

4.1. Overview of Processing

Figure 6 shows overview of the system. Multiple 2D markers are distributed inside and outside of the baseball field model which is placed on the tabletop in the real world. The

markers can be placed at arbitrary positions and poses without measuring the arrangement of them. The image of the tabletop field model is captured by a web camera attached and displayed on a handheld monitor.

This system can be divided into offline and online processes. At the offline process, first, a game history data file of a baseball game is prepared and loaded. In this file, history of game results are described play-by-play. Next, the field model is captured by the moving web camera for some seconds to automatically estimate the markers' arrangement. The detail of the algorithm is described in Section 3. These processes are executed once in advance.

At the online process, the three steps are repeated online: (1) synthesizing the baseball game scene while 1 play according the input data, (2) computing the camera's position and pose at the current frame, and (3) overlaying virtual players onto the field model. At the first step, when one line of the data file is read out, the positions of the players and the ball at every frame while 1 play are computed according to the data to render them on the field model. At the second step, the camera's rotation and translation are estimated using the markers in the current frame. At the final step, the virtual baseball scene, such as the players and the ball synthesized with CG, is overlaid onto the tabletop field model.

4.2. Input Scorebook Data File

The game played on the field model is the replayed game of the actual game which is synthesized according to input data file called "Scorebook Data File" (SDF). As shown in Figure 7, the game history of the actual game is described play-by-play in the SDF. "1 play" means the actions of the players and the ball from the moment that the pitcher throws the ball to the moment that the ball returns to the pitcher again. It is about for 15 to 30 seconds. The actions of the players and the ball in 1 play are described on one line in the SDF. The former part of the line represents the actions

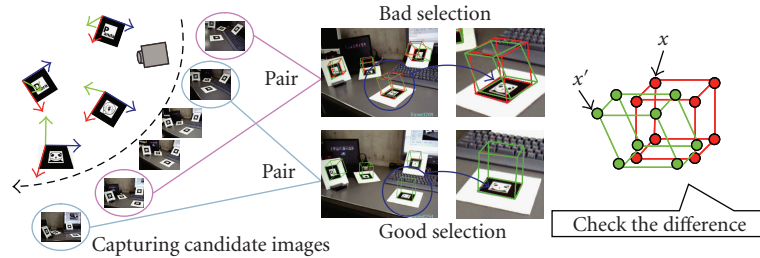


FIGURE 5: Automatic selection method of the reference images.

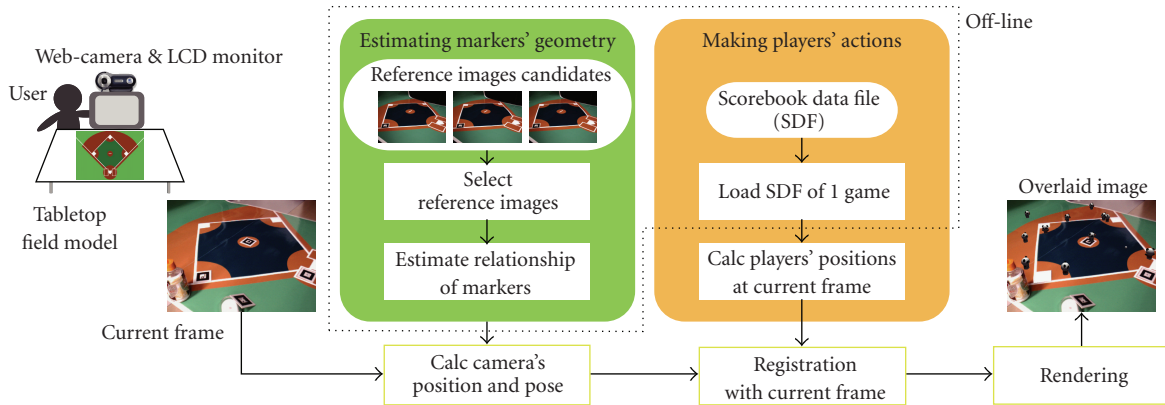


FIGURE 6: Overview of AR Baseball Presentation System.

Actions of fielders & ball	Actions of offensive players
4 4 0 B	-1 -1 -1 3 -1 -1 2 0 1
2 3 1 S	
2 3 2 H 7-9 F 8 4 1	-1 -1 0 4 -1 -1 3 1 2
3 0 0 B	
2 3 2 O -3 6 G 6 3 1	-1 -1 1 -1 -1 0 -1 2 3
	⋮

FIGURE 7: Scorebook Data File (SDF).

of the fielders and the ball, while the latter part describes the actions of the offensive players. This file is loaded in starting the system and is sequentially read out line-by-line at every 1 play. In this way, the actions of the baseball scene are described in the SDF.

4.3. Actions of Offensive Players

Offensive players indicate a batter, runners, and players who are waiting in the bench. Each player belongs to one of the six states as shown in Figure 8(a). The batter is in the batter's box, so its state is "0," third runner is "3," and the waiting players are "-1." In SDF, the destination state to which every player changes in each play is sequentially recorded. When one line of the file is read out, the destination of each player

is decided according to the data as in Figure 8(b). Then, the game scene that 3D players are moving from the present state to the destination state while 1 play is created with CG.

4.4. Actions of Fielders and Ball

In contrast to the offensive players who are just moving from present state to destination while 1 play, the fielders are doing some actions while 1 play, such as moving around the field and throwing and catching the ball, and so forth. Therefore, only the action of the ball is described in the SDF. Fielders move to catch the ball according to the action of the ball. The action of the ball while 1 play is described in Figure 9.

Fielders basically keep own positions. First, the ball is thrown by the pitcher and hit to the position which is described in part D of Figure 9. Then, the player whose position number is described in the first of part E moves to the position of part D to catch the ball. After catching the ball, the player throws the ball to the next player whose position number is described next. The next player moves to the nearest base and catches the ball. After the same iterations, finally, the ball is thrown to the pitcher.

4.5. Demonstrations

We have implemented AR Baseball Presentation System with a web camera (ELECOM UCAM-E1D30MSV) attached to a handheld monitor connected a PC (OS:Windows

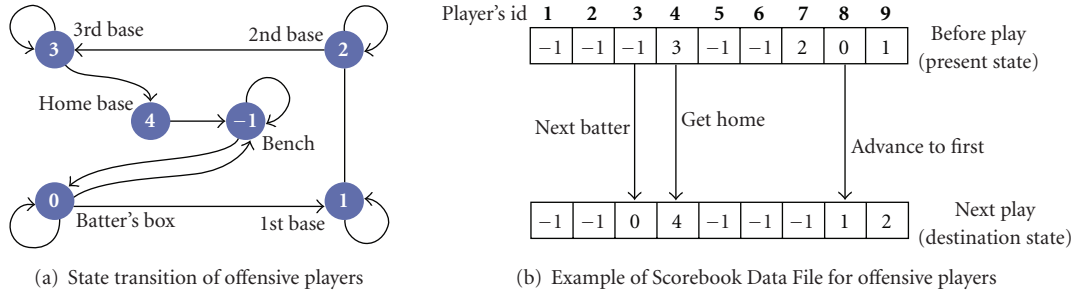


FIGURE 8: Actions of the offensive players.

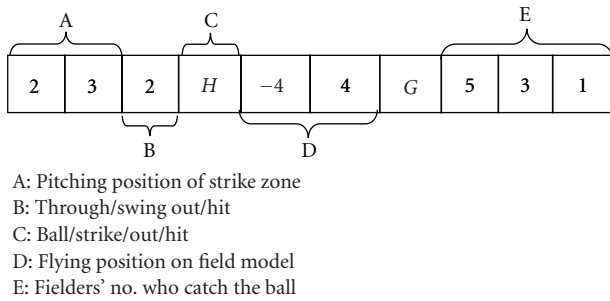


FIGURE 9: Scorebook Data File of the fielders and the ball.

XP, CPU: Intel Pentium IV 3.2 GHz. The resolution of the captured image is 640 × 480 pixels. Multiple planar markers are distributed inside and outside the field model. In this case, one of the markers must be put on one of the bases in order to determine relationship between the field model and the markers. The other markers can be placed at arbitrary positions and poses. In these experiments, we use four markers and place one of them on the third base. A Scorebook Data File of a baseball game is manually prepared in accordance with Section 4.2. 3D models of virtual objects, such as players and a ball, are rendered with OpenGL (Algorithm 1).

First, the user places the baseball field model on the tabletop and distributes the markers. Next the object scene is captured with moving around the field model for 100 frames as candidates of the reference images. Inside of the system, the best pair of the reference images is automatically selected from the candidate images. Then, the projective space is constructed by the selected reference images. The geometrical relationship of the markers is also estimated. These automatic processes take about 60 seconds. After the automatic preparations, the user inputs a Scorebook Data File and starts the system. The virtual baseball game begins on the field model and the user can watch the game from favorite view point with moving around the real world.

Figure 10 presents a baseball game: team RED versus team WHITE. In this situation, team WHITE is in the field and team RED is at bat. The bases are loaded and 4th batter of team RED is in the batter's box (frame 0–15). The pitcher throws the ball (frame 15–29). The batter hits safely to left (frame 29–35), and then all runners move up a base

(frame 50–89). In the result, team RED gets a score. In this experiment, frame rate of AR presentation is about 30 fps. Thus, user can see the baseball game at video rate.

Figure 11 shows some closeup views of the same scene. Since these images are captured from closeup view points, only a few markers can be captured in the field of view, and the captured markers are different in every frame. Even though particular markers are not continuously captured over the frames, the virtual players and the ball can be correctly registered onto the real tabletop field with the same world coordinate. This means that the consistency of the geometrical relationship between the camera and the virtual objects is kept properly although the geometrical arrangement of the markers is unknown.

In Figure 12, the angle of the camera with respect to the tabletop is too small to detect the markers lying on the tabletop plane. One marker is placed at different pose from the ground plane and the other markers are placed on the ground plane. In such a case, the markers which face to the same directions as the tabletop plane cannot be recognized because of the angle of the camera. If all the markers have to be on the same plane, it even fails recognition for most of the markers. In our registration method, however, the markers can face to various directions like Figure 12 because the markers can be placed at arbitrary positions and poses. The marker with the red cube is placed at different pose from the ground plane, so that this marker can be detected even in the case that the markers on the tabletop plane are not detected. Therefore, the registration can be stably continued even if the user moves the camera to any view point. This is a big advantage of the proposed system for applying to entertainment AR applications.

5. Interactive AR Bowling System

In the Interactive AR Bowling System, as shown in Figure 1(b), a real bowling lane model is placed on the tabletop. The users roll a real ball down the bowling lane model to knock down virtual pins generated with CG. Of course they can move around the lane model and can see the virtual bowling game scene from favorite view points by applying the algorithm presented in Section 3.

As for related work with bowling, Matysczok et al. have also proposed a bowling system using AR [19]. A user wears an HMD, in which a virtual ball, lane, and pins are

- (1) *Arrangement*
Place field model and multiple markers at arbitrary positions and poses.
- (2) *Capturing*
Capture the object scene as candidates for two reference images;
- (3) *Input*
Input Scorebook Data File;
- (4) *Observation*
Start system and observe game with moving around;

ALGORITHM 1: User's Operations.

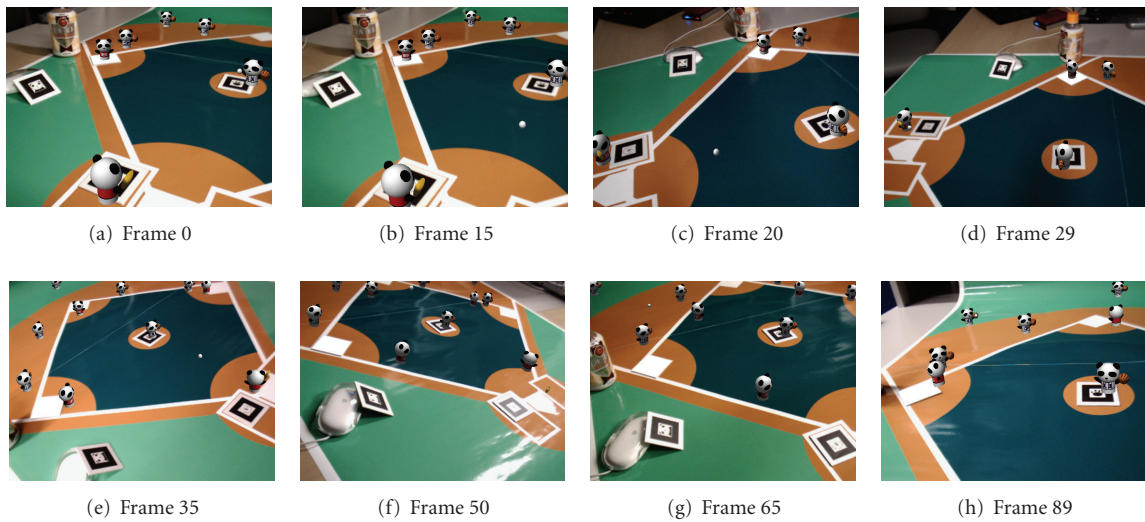


FIGURE 10: Example of play: 4th batter of team RED sends a hit to left with the bases loaded and scores a goal.

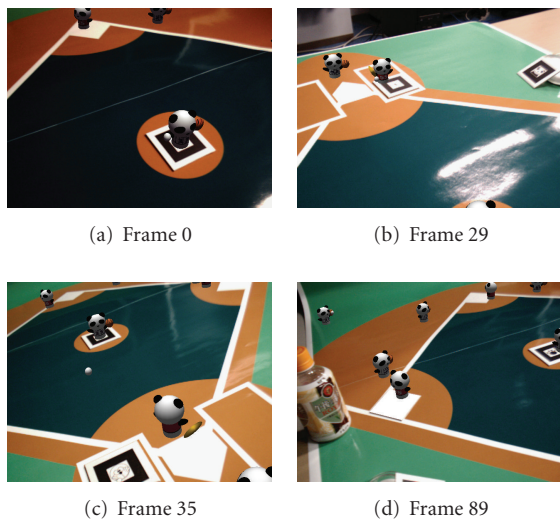


FIGURE 11: Closeup views of the same scenes as Figure 10 seen from different view points.

displayed, and interacts with the virtual ball by hand gesture. Since all objects, including the ball, are not real but virtual objects, this system just generates virtual bowling scenes

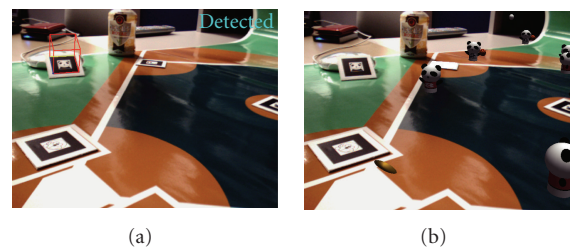


FIGURE 12: Most of the markers which face to the same directions as the tabletop cannot be detected. The marker which faces to different direction is detected successfully. (a) Marker detection: the red cube on the marker represents detected marker. (b) Augmented view: virtual objects are overlaid using the detected marker.

with the input of hand gesture from the sensors. Thus, it is unnecessary to overlay the virtual scene onto the real scene like an AR system. Moreover, the user can hardly see the real world because the virtual lane is covering the real scene. Therefore, the meaning of AR that is mixing the real world and the virtual world is lost.

In our system, in contrast, since the ball is a real object, the virtual scenes are generated according to the ball's motion in the real scene. Therefore, it is effective to be an AR

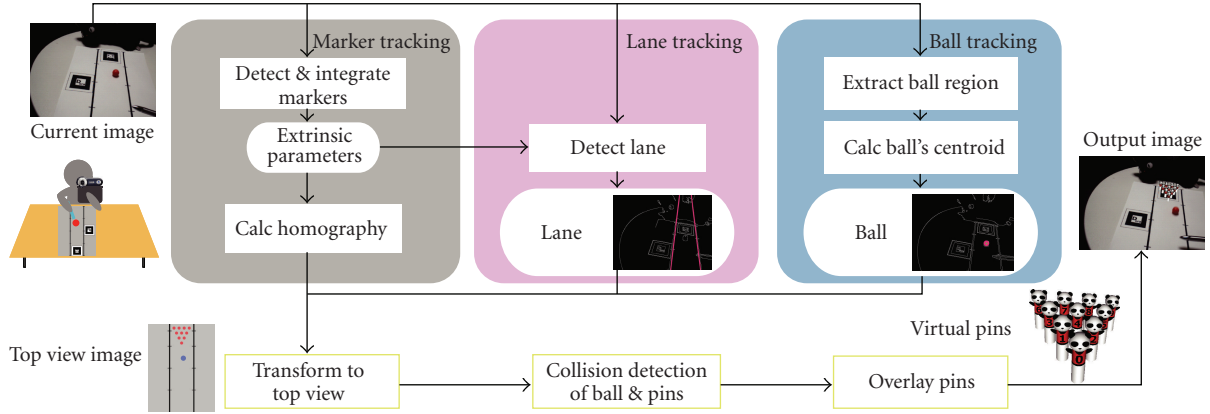


FIGURE 13: Overview of Interactive AR Bowling System.

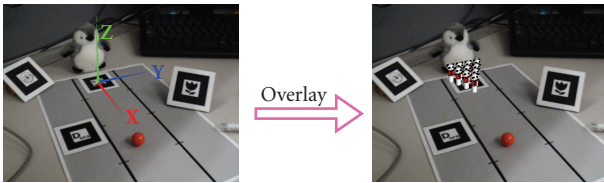


FIGURE 14: 3D coordinate system defined on the real lane model. Virtual pins are overlaid in this coordinate system.

system with overlaying the virtual pins onto the real lane model. Moreover, the user can touch the real ball, so our system achieves a real bowling style. In Matysczok’s system, special gloves with physical sensors are also required for user’s interaction, however, our system needs only a camera and a PC and use a real ball and lane model.

To realize this kind of bowling system, we have to perform the following tasks. There are two lines and 2D markers on the bowling lane model. These two lines define the lane, which means that the ball should be rolling between the two lines. In case the ball goes out of the lane, it is considered as “gutter.” Therefore, the lane and the ball have to be detected and tracked at every frame.

When the ball hits any virtual pins, the pins are knocked down. For generating such virtual pins according to the ball, the geometrical relationship between the real ball and the virtual pins has to be computed interactively. In our method, the ball’s position on the input image is transformed onto a top view image, which is the input image seen from top view, to obtain the ball’s position to the pins.

Finally, the virtual pins are overlaid onto the input image according to the camera’s position and pose, which are corresponding to extrinsic parameters of the camera. The extrinsic parameters are estimated by multiple 2D markers.

5.1. Overview of Processing

Figure 13 shows a flowchart of our proposed system. First, the images captured by the web camera are applied to three kinds of processing; marker tracking, lane tracking,

and ball tracking. During the marker tracking process, AR-Toolkit [4] detects 2D markers placed around the lane model. Then, a 3D coordinate system where the virtual objects should be overlaid is defined on the lane model. Since the relationship of the lane with respect to the marker is fixed by the lane model, the two lines which consist of the lane can be detected by marker detection process. During the ball tracking process, a region of the ball is detected. We assume that the centroid of the region is the ball’s position.

After the tracking of the markers, the lane, and the ball, the ball’s position is transformed to the top view image to compute the geometrical relationship between the ball and the virtual pins. Then, collisions between the ball and the virtual pins are computed according to their relative position. Finally, the pins are overlaid onto the input image by using the extrinsic parameters computed at the marker tracking process.

5.2. Marker Tracking

The multiple markers placed around the lane model are detected as same as in our baseball system. The geometrical relationship of the markers are also estimated by the registration method described in Section 3. Then, a 3D coordinate system, where the virtual objects should be overlaid, is defined on the lane model as shown in Figure 14. In our system, to track the trajectory of the ball on the lane, we use a top view image. The top view image is an image which is the input image transformed to the top view point. By this transformation, the ball’s 2D motion becomes understandable. Therefore, we compute a homography H [20] to transform the input image to a top view image. The H is the planar projection matrix which relates the real-lane model and the lane model area in the input image and can be computed from the corresponding points on the lane model in the real world and the input image. It will be used in Section 5.4.

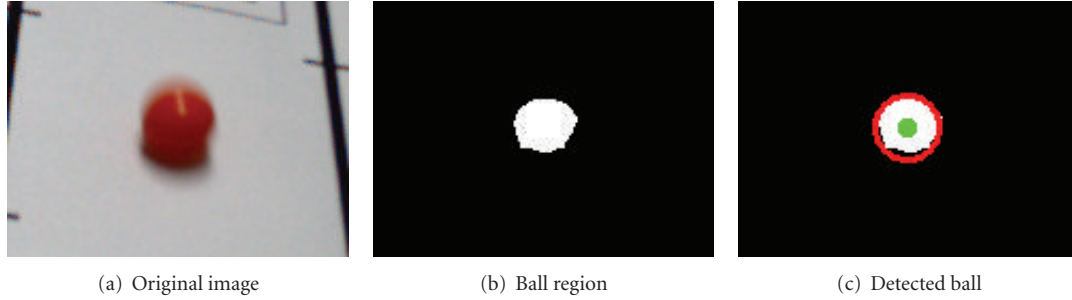


FIGURE 15: Ball detection.

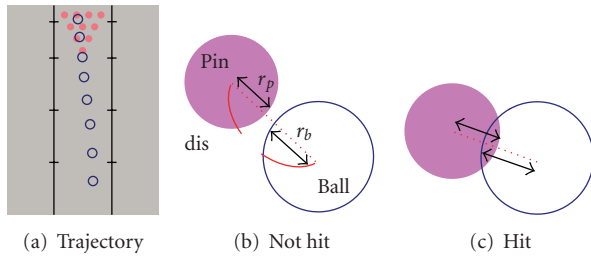


FIGURE 16: Collision detection by trajectory of the ball.

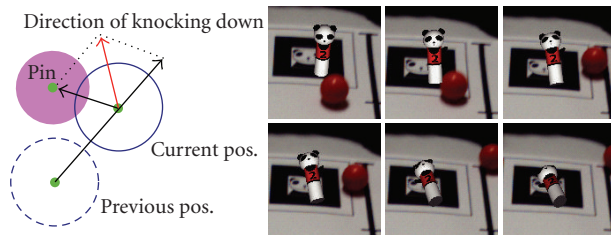


FIGURE 17: Direction of knocking down.

5.3. Ball Tracking

In this system, we assume that the color of the ball should be quite different from the lane model. In this paper, we use a red ball on a gray lane model as shown in Figure 15(a). For detection of the ball, first, red regions are detected from the input image by dividing it into R , G , B channel images. Figure 15(b) shows the image after dilation and erosion a few times. Finding the minimal circumscribed circle (contour) for the detected region, the center of the circle is considered as the 2D ball's position in the input image as shown in Figure 15(c).

5.4. Transformation to Top View Image

Using homography \mathbf{H} computed at the marker tracking process, the ball's position on the input image is transformed onto the top view image that provides a geometrical relationship between the ball and the pins on the lane model.

As shown in Figure 16(a), the trajectory of the ball can be obtained. This trajectory is used to detect the collision

between the ball and the pins, and compute the directions in which the pins are knocked down.

5.5. Collision Detection of Ball and Pins

We assume that radii of the ball and the pins are r_b and r_p , respectively, and define the distance between the ball and each pin as dis . For detecting a collision between the ball and the pins, the distance dis is computed from the top view image at every frame. The collision is detected by comparing distance and radius as in the following equation, and as in Figures 16(b) and 16(c):

$$\{(r_b + r_p) \leq dis \Rightarrow \text{Not hit}, (r_b + r_p) > dis \Rightarrow \text{Hit}. \quad (7)$$

5.6. Overlay Virtual Pins

After the collision detection, the pins are generated with CG and overlaid onto the image. If the collision is detected, the pins are gradually inclined and knocked down. The direction of knocking down is defined by trajectory of the ball. As shown in Figure 17, the direction is computed by a motion vector of the ball, which is decided by ball's positions in previous and current frames, and a vector from the ball to each pin.

The generated pins are superimposed onto the image by the extrinsic parameters computed by 2D markers. The user can see the virtual pins according to the motion of the camera and the rolling ball.

5.7. Demonstrations

In this experiment, four 2D markers on the lane model are placed on the tabletop to estimate the camera motion (extrinsic parameters). Some of them are placed on the same plane as the tabletop; the others are aligned in various directions. The geometrical relationship between every marker is automatically computed by the method in Section 3. One of the markers, which lies between the lines, is also used for defining the 3D coordinate system. The resolution of the captured image is 640×480 pixels. The virtual pins are rendered with OpenGL library.

Figure 18 shows the detected lane and ball's trajectory. Both of the lane and the ball can be correctly detected and tracked over all frames by our tracking method, according

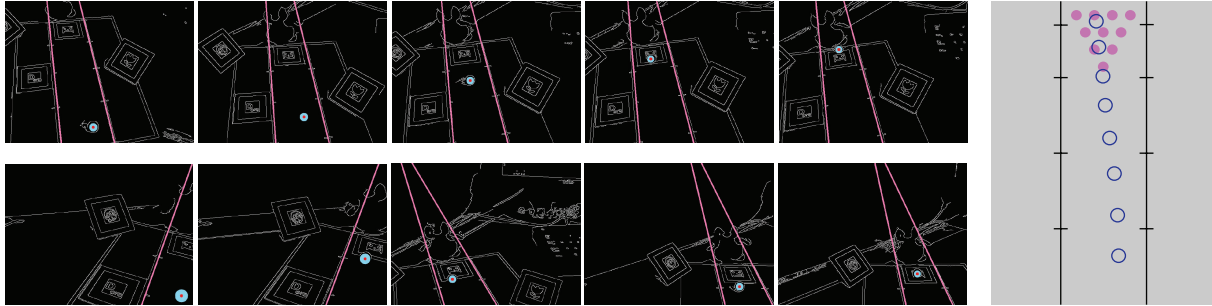


FIGURE 18: Detected lane and ball and trajectory of ball.

to the camera motion. The ball's position is also successfully transformed onto the top view image by the homography computed by 2D markers.

Figure 19 shows example scenes where the virtual pins are overlaid according to the camera motion. If we use only one marker for overlaying the pins, the registration becomes impossible when the ball is rolling over the marker because the marker cannot be detected. Therefore, we have to use multiple markers. Even though particular markers are not continuously captured over the frames, the virtual pins can be correctly registered on the lane model because of our registration algorithm which can estimate the geometrical relationship of the markers.

Moreover, since the collision of the real ball and the virtual pins are successfully detected, some pins are knocked down by hitting of the ball. The pins existing behind the hit pins are also knocked down as a chain reaction of the front pins by computing the direction of knocking down from the trajectory. This system runs 30 fps, so the user can enjoy the bowling game at video rate.

6. User Study

6.1. Baseball System

AR Baseball Presentation System visually replays the baseball game which was previously played in the other place. In contrast with the usual way to know the game history, such as watching the captured video or reading the recorded scorebook, our AR system can provide the users with much realistic sensation as an entertainment application.

We design our system as AR system which can overlay the CG scene on the real field model in front of the user as well as visualize the recorded baseball game by 3D CG. Using this system, the user can watch the game from the favorite view point by just moving around the field model. Such simple way for watching the CG-represented event using the AR system provides more immersive feeling than using usual CG viewers, in which a mouse or a keyboard is used for changing view points [21].

In this user study, we intend to evaluate how the AR system is effective to enhance the quality of entertainment. There are a lot of factors that may enhance the quality of entertainment such as usability, interactivity, visual effects. Those factors are evaluated by studying "how quickly," "how

easily," and "how intuitively" the user can change their view point. The same factors are also evaluated for the usual CG viewer. Then, both of results are compared to evaluate the effectiveness of designing our system as AR system.

In this evaluation experiment, we prepared two kinds of baseball observation systems as shown in Figure 20, one is our AR Baseball Presentation System, another one is created by only CG. In this CG system, the user watches the baseball game with changing the view point by using a keyboard of PC. The rotation and translation about X - Y - Z axis are assigned to each key. In the AR system, they just move around the field model with a handheld monitor. Then, we asked 15 examinees to use these systems, and measured the time each examinee spent on moving the view point to the specified view points as shown in Figures 21(a)–21(d).

Figure 21 shows the average time which the examinees spent to change their view points. We can find that the CG system took much longer time than the AR system. In this experiment, every examinee spent triple to ten longer times to change the view point in CG system than AR system. Because the users only have to move around the field model with a handheld monitor to their favorite positions, our AR system can quickly change the view points.

This result can also be found in Figure 22, which shows answers of questionnaire about changing view points. We asked four questions (a)–(d), and then the examinees rated on a scale of 1 to 5. In the same way as the actual measurement time in Figure 21, most of the examinees felt that our AR system was easier than the CG system to change their view points to their desired positions quickly and intuitively. The questionnaire also asked whether they could change the view point with watching the game. As a result, most of examinees felt that the AR system was easier to change the view point with watching the game. This is because the view point corresponding to the user's own view point, while the view point of the CG system is the virtual camera position. Therefore, designing our system as the AR system is quite helpful for any user to handle this kind of digital contents because the operation is very easy and intuitive.

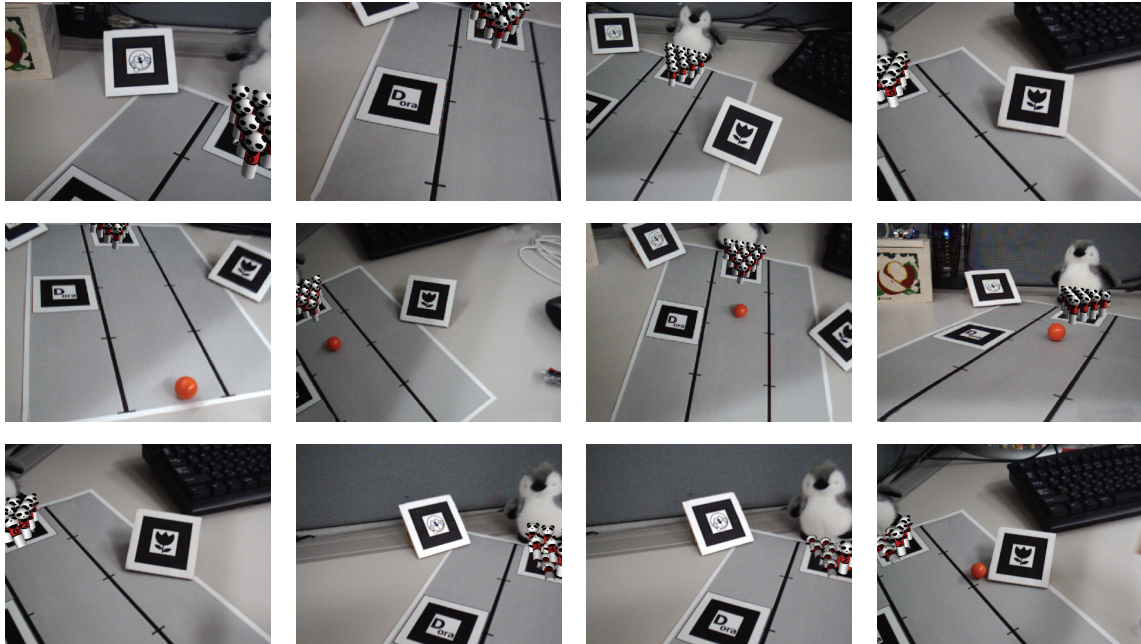


FIGURE 19: Resulting images on which virtual pins are overlaid.

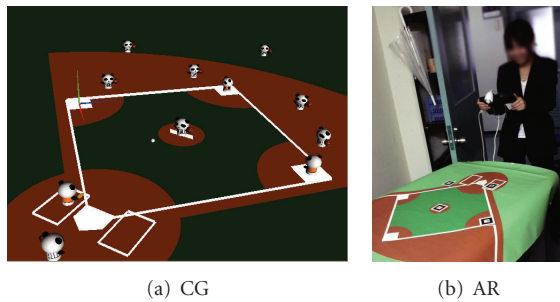


FIGURE 20: Prepared baseball system.

6.2. Bowling System

AR Bowling System consists of real and virtual objects, such as the real ball, the real lane model, and the virtual pins. In this system, the users physically touch and roll the real ball on the real lane. Such physical communication provides the users with much reality as a tangible interface [5–8].

Therefore, we focus on the effectiveness of such tangibility of our AR system as the evaluation point of the AR Bowling System. The AR system can interact with the virtual world by rolling the real ball with a hand unlike a CG system which entirely consists of virtual objects. This means that we should evaluate how effective the direct touch to the ball for the bowling game as an entertainment application. For evaluating this point, we evaluate “how naturally the users can control the ball,” “whether they actually feel that the ball is controlled by themselves,” and “whether the game is challenging.”

In the same way as the baseball system, we prepared two kinds of bowling system, our AR Bowling System, a CG

bowling system, and a real toy bowling game as shown in Figure 23. We also asked the examinees to play the real toy bowling game before using the CG and AR bowling systems. In the CG system, when the user drags the virtual ball on the display by a mouse, the ball starts rolling. The direction of the ball is defined according to user’s dragging. The speed of the ball is also defined as the length of dragging of the mouse. In the AR system, the user rolls the real ball on the lane model with a hand and watches the proceeding through the handheld monitor. After playing all the systems, they answered questionnaire by rating on a scale of 1 to 5. The questionnaire items and the results are shown in Figure 24.

Although they could only decide the direction and speed of the ball in the CG system with a mouse, the users could freely control the ball in the AR system; and also they could really touch the ball. As a result, they actually felt rolling the ball by themselves in the AR system. Since the ball is virtual object in the CG bowling system, the users can roll the ball only linearly. Therefore, some users said that they wanted to roll the curve-ball. In order to achieve such a curve-ball in CG system, some random elements have to be included. Since such randomness cannot be handled by the users, however, such system is unacceptable as computer games. On the other hand, the users can roll any kind of ball depending on their skills because the ball of the AR system is real object. Therefore, most of the users felt that the ball’s motion of the AR system was more natural like the real bowling game than the CG system. Because there are various ways to roll the ball in the AR system, the game is not too simple to complete. For example, some users sloped the lane; other users used a pen to roll the ball instead of their hands. Therefore, they felt that the AR system was more challenging than the CG system.

By the way, when playing the real bowling game, we asked the examinees to raise and reset the fallen pins by themselves.

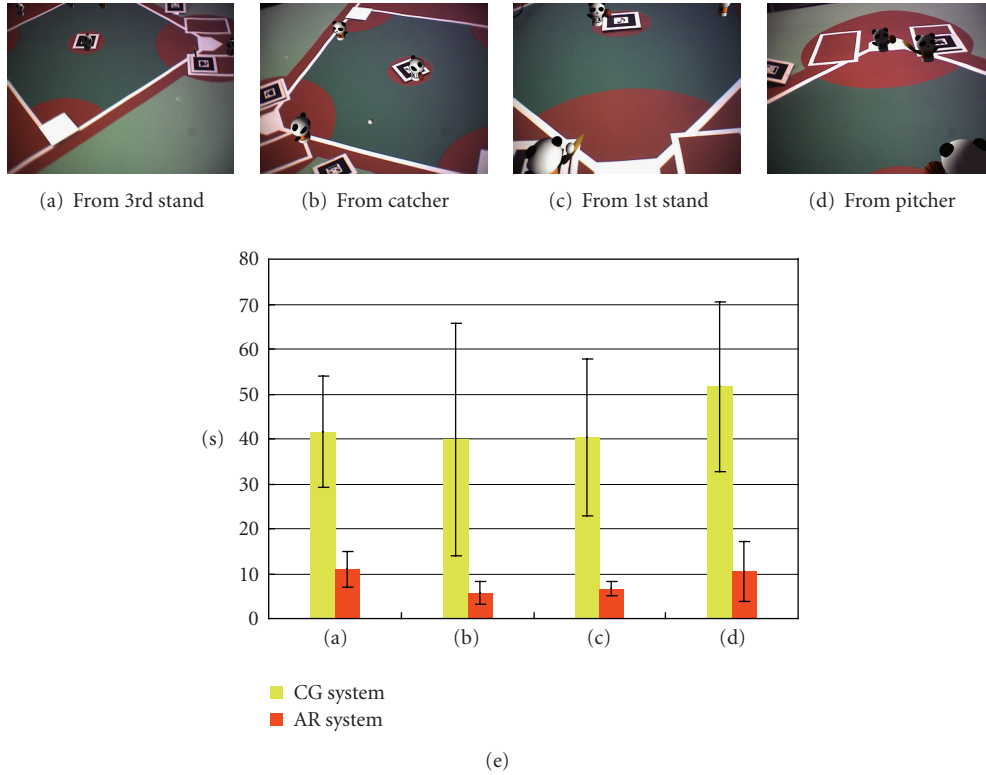


FIGURE 21: Average time the examinees took to change their view points to specified view points.

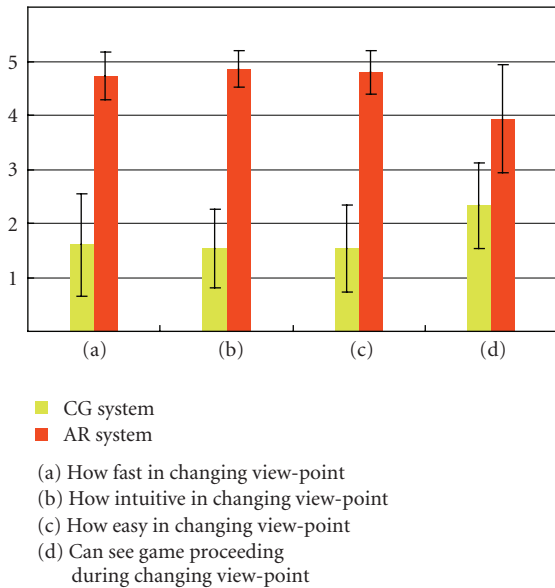


FIGURE 22: Answer of questionnaire about changing view points.

As a result, they felt that it's troublesome to reset the pins every time. As described before, physical communication is very effective for the young users, however, we can afraid that little children cannot arrange the pins very well. On the other hand, virtual bowling game (both of AR system and CG system) do not require such a task because the users only

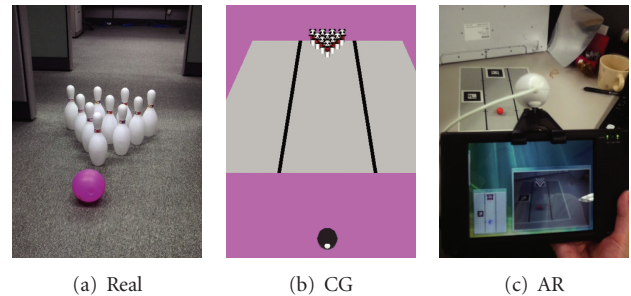


FIGURE 23: Prepared bowling system.

have to press 1 button to reset the fallen pins. Therefore, the concept of the AR Bowling is very helpful for any user by allowing the physical communication without troublesome task.

7. Conclusions

In this paper, we have presented two AR applications using vision-based tracking method: AR Baseball Presentation System and Interactive AR Bowling System. Both of the applications can be enjoyed on the tabletop in the real 3D world only with a web camera and a handheld monitor connected to a PC. It is a big advantage for home users that our applications do not require any special device such as positioning sensors or a high-performance PC.

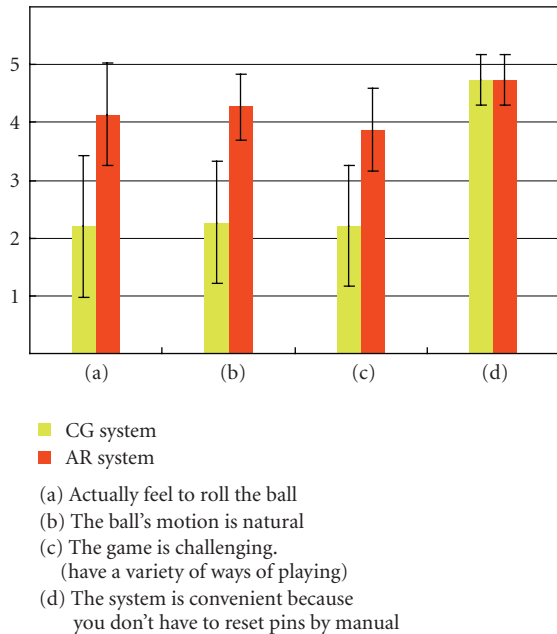


FIGURE 24: Answer of questionnaire about the reality of playing bowling.

Users can interactively change their view points by moving around the tabletop because of multiple 2D markers. In usual AR applications using multiple 2D markers, users have to measure the distance between the markers. Of course, such extra task is unnecessary in our applications by applying the registration method with the 3D projective space. In contrast with usual CG viewers in which a mouse or a keyboard is used for changing view points, changing view points by moving the users is very intuitive and easy way. Such a facility is very important specially for children.

Using the baseball application, the users can watch a 3D virtual baseball game in front of themselves. It can be a future-oriented 3D game which is represented in movies or animations. The bowling application can interest children because their actions in the real world affect the virtual world.

For the future work, we have to consider that sound is very important element. Since sound can directly interest people, sound is very effective as response of the user's interaction in AR. For example, if the baseball system downloads ambient sound data in the actual baseball stadium with the scorebook data and play the sound according to the game, more realistic sensation will be given to the users. In the bowling system, sound effect of collision between the real ball and the virtual pins is also effective and interesting. So we would like to adopt sound elements in the future.

References

- [1] M. Haller, M. Billinghurst, J. L. Leitner, and D. Seifried, "Coeno-enhancing face-to-face collaboration," in *Proceedings of the International Conference on Augmented Tele-Existence (ICAT '05)*, pp. 40–47, Christchurch, New Zealand, December 2005.
- [2] A. Henrysson, M. Billinghurst, and M. Ollila, "Virtual object manipulation using a mobile phone," in *Proceedings of the International Conference on Augmented Tele-Existence (ICAT '05)*, pp. 164–171, Christchurch, New Zealand, December 2005.
- [3] G. Klein and T. Drummond, "Sensor fusion and occlusion refinement for tablet-based AR," in *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*, pp. 38–47, Arlington, Va, USA, November 2004.
- [4] M. Billinghurst, S. Cambell, I. Poupyrev, et al., "Magic book: exploring transitions in collaborative ar interfaces," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, p. 87, New Orleans, La, USA, July 2000.
- [5] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso, "Curlybot: designing a new class of computational toys," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*, pp. 394–401, Pittsburgh, Pa, USA, May 1999.
- [6] B. Piper and H. Ishii, "PegBlocks: a learning aid for the elementary classroom," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, pp. 686–687, Minneapolis, Minn, USA, April 2002.
- [7] K. Ryokai, S. Marti, and H. Ishii, "Designing the world as your palette," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, pp. 1037–1049, Portland, Ore, USA, April 2005.
- [8] J. Zigelbaum, A. Millner, B. Desai, and H. Ishii, "Bodybeats: wholebody, musical interfaces for children," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, pp. 1595–1600, Montreal, Canada, April 2006.
- [9] Y. Uematsu and H. Saito, "AR registration by merging multiple planar markers at arbitrary positions and poses via projective space," in *Proceedings of the International Conference on Augmented Tele-Existence (ICAT '05)*, pp. 48–55, Christchurch, New Zealand, December 2005.
- [10] A. Henrysson, M. Billinghurst, and M. Ollila, "AR tennis," in *Proceedings of the 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH '06)*, Boston, Mass, USA, July-August 2006.
- [11] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington, "FlightTracker: a novel optical/inertial tracker for cockpit enhanced vision," in *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*, pp. 212–221, Arlington, Va, USA, November 2004.
- [12] E. Foxlin and L. Naimark, "Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker," in *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)*, pp. 151–160, Tokyo, Japan, October 2003.
- [13] E. Foxlin and L. Naimark, "Vis-traker: a wearable vision-inertial selftracker," in *Proceedings of the IEEE Virtual Reality Conference (VR '03)*, pp. 199–206, Los Angeles, Calif, USA, March 2003.
- [14] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab, "Marker-less tracking for ar: a learning-based approach," in *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '02)*, pp. 295–304, Darmstadt, Germany, September-October 2002.
- [15] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top ar

- environment,” in *Proceedings of IEEE and ACM International Symposium on Augmented Reality (ISAR '00)*, pp. 111–119, Munich, Germany, October 2000.
- [16] D. Kotake, S. Uchiyama, and H. Yamamoto, “A marker calibration method utilizing a priori knowledge on marker arrangement,” in *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*, pp. 89–98, Arlington, Va, USA, November 2004.
- [17] Y. Uematsu and H. Saito, “Vision-based registration for augmented reality with integration of arbitrary multiple planes,” in *Proceedings of the 13th International Conference on Image Analysis and Processing (ICIAP '05)*, vol. 3617 of *Lecture Notes in Computer Science*, pp. 155–162, Cagliari, Italy, September 2005.
- [18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.
- [19] C. Matyszczok, R. Radkowski, and J. Berssenbruegge, “AR-bowling: immersive and realistic game play in real environments using augmented reality,” in *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE '04)*, pp. 269–274, Singapore, June 2004.
- [20] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2003.
- [21] GLT ZPR, <http://www.nigels.com/glt/gltzpr/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

