AR Baseball Presentation System with Integrating Multiple Planar Markers

Yuko Uematsu and Hideo Saito

Graduate School of Science and Technology, Keio University, Yokohama, Japan
{yu-ko, saito}@ozawa.ics.keio.ac.jp

Abstract. This paper presents "On-line AR Baseball Presentation System", which is a vision-based AR application for entertainment. In this system, a user can watch a virtual baseball game scene on a real baseball field model placed on a tabletop through a web-camera attached to a hand-held LCD monitor. The virtual baseball scene is synthesized from an input history data of an actual baseball game. Visualizing the input history data can help the user to understand the contents of the game. For aligning the coordinate of the virtual baseball game scene with the coordinate of the real field model, we use multiple planar markers manually distributed into the real field model. In contrast with most of AR approaches using multiple markers, we do not need any manual measurement of the geometrical relationship of the markers, so that the user can easily started and enjoy this system.

1 Introduction

Augmented Reality(AR)/Mixed Reality(MR) is a technique for overlaying virtual objects onto the real world. AR has recently been applied to many kinds of applications including entertainment, such as [1,2,3,4]. In this paper, we propose "AR Baseball Presentation System" which allows a user to watch a virtual baseball game at any place in the real world via a web-camera attached to a hand-held LCD-monitor with moving around the real world. For example, as shown in Fig. 1, the user can observe the virtual baseball game on the tabletop where a real baseball field model is placed. The virtual baseball game scene is synthesized with 3D CG players and overlaid on the field model in the real world. In this proposed system, the virtual game played on the field model is the replayed game of the actual game. When the history data of the actual game, such as web-site, is input, the same game is replayed on the real field model by 3D virtual players. In contrast with just reading such 2D web-site, observing the 3D game will be a big help for the user to understand the game history.

For such AR applications, registration of virtual objects with real images is a significant issue. Therefore many approaches have been studied before such as sensor-based approach, vision-based approach and hybrid approach, etc. Especially the vision-based registration does not require any special devices except cameras, so a lot of methods have been proposed. This kind of approach can be categorized into marker-based approach and using natural features approach.

Z. Pan et al. (Eds.): ICAT 2006, LNCS 4282, pp. 163–174, 2006.

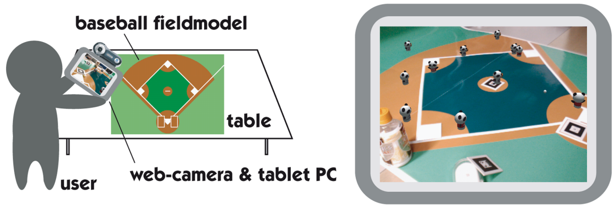


Fig. 1. AR Baseball Presentation System

Using natural features, the augmentation is naturally realized [5,6,7]. However, it is not easy to construct a robust on-line system because exact recognition of natural features in real-time is hard task and registration jitters are often caused. It is also true that only few features are available for registration in the real world. Since AR-based applications have to work in real-time and on-line, we focus on marker-based approach.

As a marker-based approach, "AR-Toolkit" [8] is very popular tool for simple implementation of an on-line AR system [9,10,11]. The simplest way is using one marker like [9], however, the camera's movable area is limited to the area where the camera (user) can see the marker in the frame. Moreover, when the marker cannot be recognized properly because of a change in its visibility, the registration of virtual objects is getting unstable.

Using multiple markers can solve such problems. In order to use multiple markers, however, it is necessary to know geometrical arrangement of the markers, such as their positions and poses, in advance [12,13,14,15,16]. In [12], they need the position and pose of a square marker and the position of a point marker in advance. In [13], they proposed marker-less registration method by setting up a learning process. In the learning process, the markers' geometrical information is required for learning the markers.

In most cases, the task for measuring of the multiple markers' arrangement is manually implemented. However such a task is very time-consuming and not sufficiently accurate. Kotake et al. [17] 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, *e.g.* multiple markers are coplanar.

We have already proposed Multi-Markers based On-line AR System [18]. In this system, multiple planar markers like AR-Toolkit are placed at arbitrary positions and poses in the real world. These markers can be used for the registration without manual measurement task because of introducing the registration method using arbitrary multiple planes [19]. In this registration method, geometrical prior-knowledge about the planes is not required because the geometrical arrangement of the planes is automatically estimated via 3D projective space defined by projective reconstruction of two images. Therefore, in this proposed system, the markers can be placed at arbitrary positions and poses on the field model. The user can freely move around wide area. Moreover the system can work at video-rate. It is suitable for AR entertainment applications.

2 System Overview

Fig. 2 shows overview of the proposed system. Multiple 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 with a web-camera attached to a LCD monitor and displayed on the monitor. The user watches the field model and the virtual baseball game the web-camera and the LCD monitor.

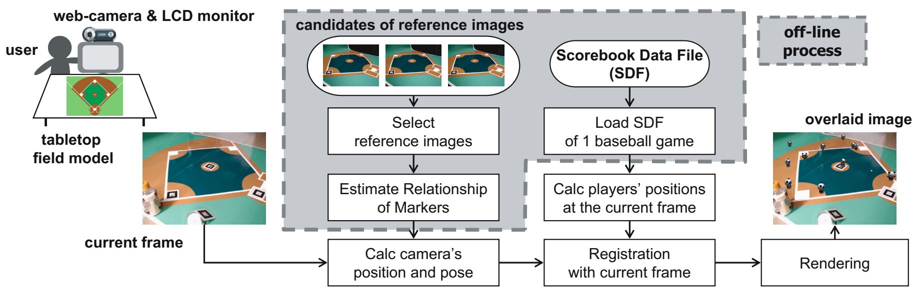


Fig. 2. Overview of Proposed System

This system can be divided into off-line and on-line process. At the off-line process, first, a game history data file of a baseball game called "Scorebook Data File" (SDF) 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 estimate the markers' arrangement automatically. The above processes are executed once in advance.

At the on-line process, the three steps are repeated on-line: (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 SDF 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.

Using this system, the user can see the virtual baseball game from favorite view points with the web-camera and LCD monitor. Because the arrangement of the markers does not need to be measured, the markers can be placed in a wide area, and the camera can also move around in wide area.

3 Replay Baseball Game by 3D Virtual Players

3.1 Input Scorebook Data File

The game played on the field model is the replayed game of the actual game which is reproduced according to input data file called "Scorebook Data File" (SDF). As shown in Fig. 3, 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 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.

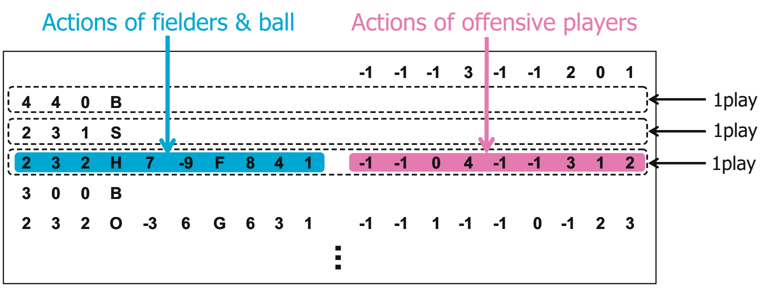
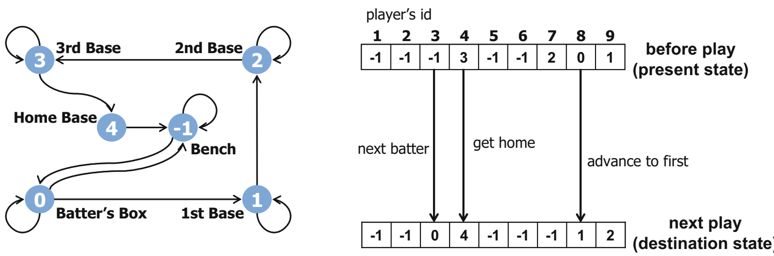


Fig. 3. Scorebook Data File (SDF)

3.2 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 Fig. 4(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 like Fig. 4(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.



(a) State transition of offensive players

(b) Example of Scorebook Data File for offensive players

Fig. 4. Actions of the offensive players

3.3 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, etc. 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 as shown in Fig. 5

Fielders basically stay own positions. First, the ball is thrown by the pitcher and hit to the position which is described in part D of Fig. 5. Then the player whose position number is described in the fist 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.

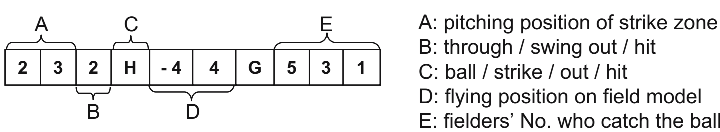


Fig. 5. Scorebook Data File of the fielders and the ball

4 Multi-markers Based Registration

In this section, we explain the algorithm of registration method in the Multi-Markers Based On-line AR System [18]. This algorithm is based on [19].

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

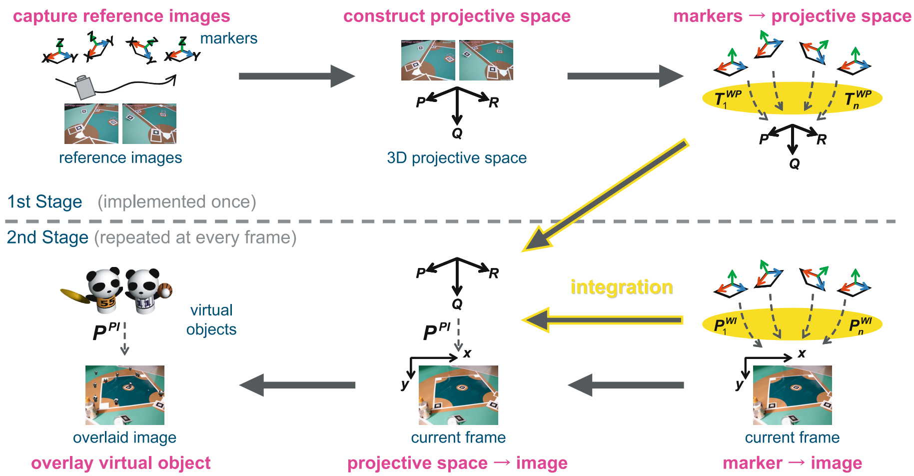


Fig. 6. Overview of the Registration Method

projective reconstruction of two reference images. The reference images are automatically selected from some candidate images. The geometrical relationship of the markers is represented as a transformation matrix called $\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{WP}}$ between each marker i and the projective space. These transformation matrices are computed once in advance. At the second stage, a projection matrix $\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{WI}}$ from each marker i to the input image. Those projection matrices and the transformation matrices which are computed at the first stage are integrated into a projection matrix $\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{PI}}$ by eq. (1) respectively, which is based on each marker i and projects the projective space to the input image.

$$\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{PI}} = \boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{WI}} \left(\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{WP}}\right)^{-1} \tag{1}$$

Moreover those $\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{PI}}$ are integrated into one projection matrix $\boldsymbol{P}^{\boldsymbol{PI}}$ by least-square method. Then virtual objects described in the projective space coordinate system are overlaid onto the input image by using the projection matrix. These processes of the second stage are performed at every frame.

4.1 3D Projective Space

A 3D projective space is used for estimation of 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 Fig. 7, a 3D space P-Q-R is defined as a 3D projective space, which is projected to the reference image A and B by following equations.

$$\left[\, u_A v_A 1 \,\right]^{\top} \simeq \boldsymbol{P_A} \left[\, P\ Q\ R\ 1 \,\right]^{\top}, \qquad \left[\, u_B v_B 1 \,\right]^{\top} \simeq \boldsymbol{P_B} \left[\, P\ Q\ R\ 1 \,\right]^{\top} \tag{2}$$

$$P_A = [\mathbf{I} \,|\, \mathbf{0}], \qquad P_B = \left[-\frac{[e_B]_\times F_{AB}}{\|e_B\|^2} \,\middle|\, e_B \right] \tag{3}$$

where, $[u_A, v_A, 1]^\top$ and $[u_B, v_B, 1]^\top$ are homogeneous coordinates of 2D points in the reference images, and $[P, Q, R, 1]^\top$ is also homogeneous coordinates of a 3D point in the projective space. F_{AB} is a F-matrix from the image A to B, e_B is an epipole on the image B, and $[e_B]_\times$ is the skew-symmetric matrix [20].

Since the projective space is defined by projective reconstruction of the reference images, the accuracy of F_{AB} is important. F_{AB} is computed from the projection matrix which is computed from the markers by using the algorithm of [8]. Therefore the accuracy of F_{AB} is depending on the accuracy of marker detection. In this system, two reference images which have most accurate F_{AB} are automatically selected. The details will be described in next section.

4.2 Automatic Selection of Reference Images

The projective space is defined by the projective reconstruction of two reference images. Therefore the F-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 Fig. 8.

First, the object scene is captured for a few seconds by a moving camera. This image sequence becomes the candidates of the reference image. When two images are selected from the candidate images, projection matrices based on the markers which exist in the selected reference images are computed by using the algorithm of [8]. Here, 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 F-matrix based on marker i is computed as following equation,

$$F_{AB_i} = [e_{B_i}]_\times P_{B_i} P_{A_i}^- \tag{4}$$

where, $P_{A_i}^-$ represents the pseudo inverse matrix of P_{A_i} [20]. Then one F-matrix is selected as F_{AB} which has the smallest projection error:

$$error = {m_B}^\top F_{AB_i} m_A \tag{5}$$

where, m_A and m_B are corresponding points in the selected reference images.

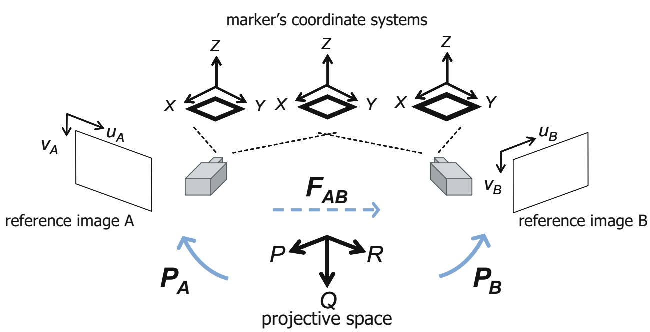


Fig. 7. Projective 3D space defined by projective geometry of the reference images

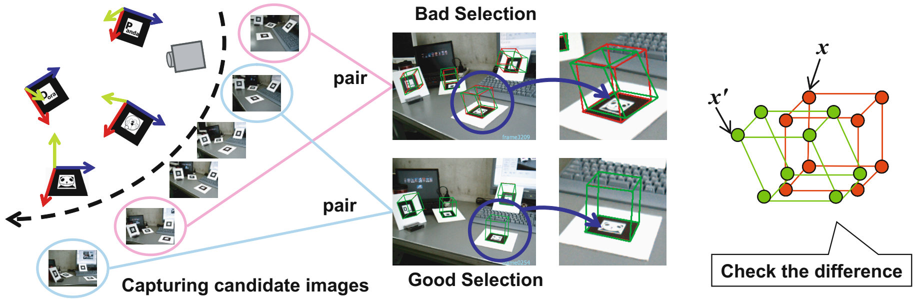


Fig. 8. Automatic selection of the reference images

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

$$\boldsymbol{x_i} = \boldsymbol{P_{B_i} X_W}, \qquad \boldsymbol{x_i'} = \left(\boldsymbol{P^{PI} T_i^{WP}}\right) \boldsymbol{X_W} \tag{6}$$

Although these two coordinates should be equal, if the combination of the two reference images is not reasonable, they will be different. In such case, we return to the phase of selecting a pair of temporary reference images. We iterate these processes until every difference of $\boldsymbol{x_i}$ and $\boldsymbol{x_i'}$ based on plane i becomes smaller than a few pixels. In the experiments in Sec.5, we decide the threshold as 3 pixels.

5 Experimental Results

We have implemented AR Baseball Presentation System. A user watches a tabletop baseball field model thorough a web-camera (ELECOM UCAM-E1D30MSV) attached to a LCD monitor. 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 sec. 3.1. 3D models of virtual objects, such as players and a ball, are renderd with OpenGL.

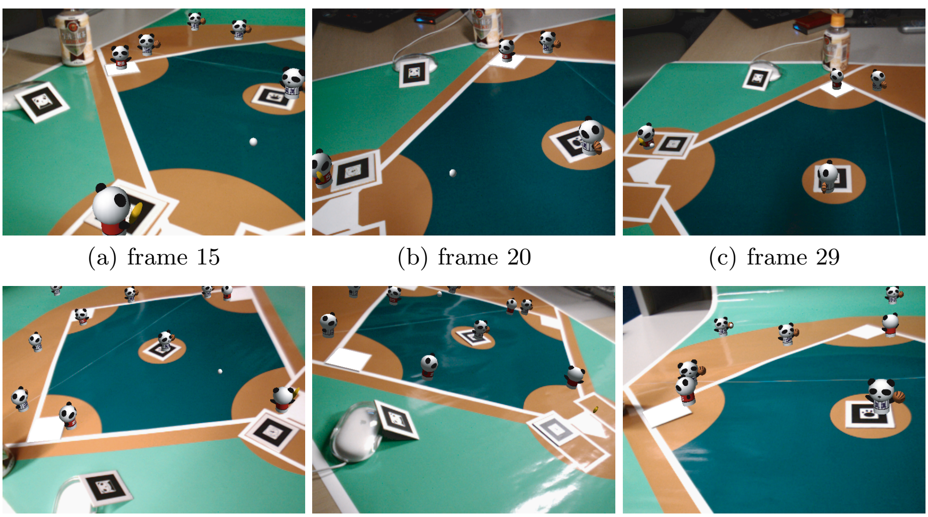
First, the user places the baseball field model on the tabletop and distributes the markers. Next the object scene is captured with moving around the field model for 100 frames as candidates of the reference images. Inside of the system, the best pair of the reference images is automatically selected 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

User's Operations

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;

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.

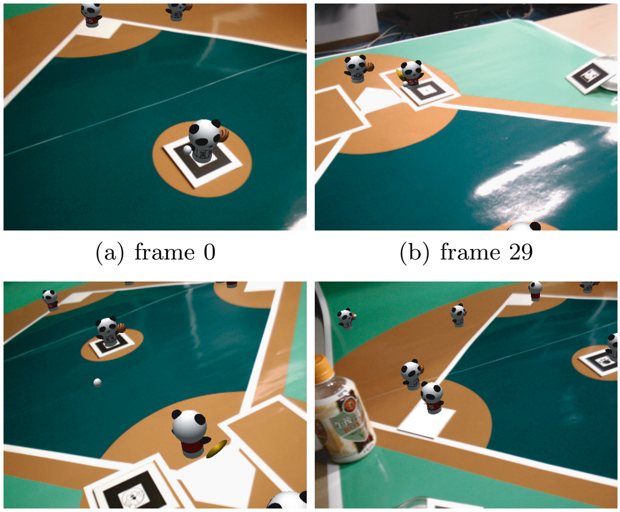
Fig. 9 presents a baseball game: team RED vs. team WHITE. In this situation, team WHITE is in the field and team RED is at bat. The bases are loaded and 4th batter of team RED is in the batter's box. The batter hits safely to left, and then all runners move up a base. Team RED gets a score. In this experiment, frame rate of AR presentation is about 10 fps with a desktop PC (OS:Windows XP, CPU:Intel Pentium Ⅳ 3.6GHz). Thus user can see the baseball game at video-rate without feeling any discomfort.



(a) frame 15 (b) frame 20 (c) frame 29

(d) frame 35 (e) frame 50 (f) frame 89

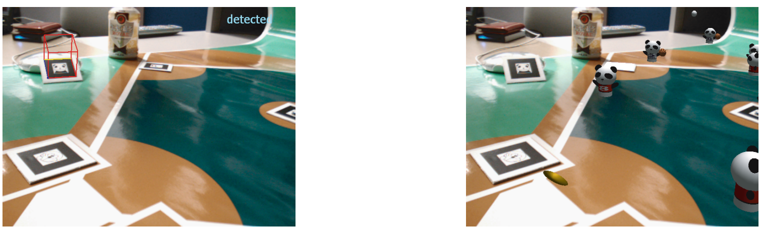
Fig. 9. Example of play: 4th batter of team RED sends a hit to left with the bases loaded and scores a goal



(a) frame 0 (b) frame 29

(c) frame 35 (d) frame 89

Fig. 10. Closeup views of the same scenes as Fig. 9 from different view points



(a) (b)

Fig. 11. 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.

Fig. 10 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 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 Fig. 11, the angle of the camera relative 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 Fig. 11 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.

6 Conclusion

We have presented AR Baseball Presentation System based on multiple planar markers. On-line AR System using multiple markers placed at arbitrary positions and poses is extended to the AR application so that a baseball game can be presented on the tabletop field model in the real world according to an input game history data of the game. Multiple markers can be placed anywhere without knowledge about their geometrical arrangement, so extra effort, such as measuring geometrical relationship of the markers, is not necessary in advance. Since such measurement is not needed, the directions of the markers' faces are also free. Making the markers face to various directions allows keeping the registration accurate. Moreover the proposed system can be performed at video-rate.

References

1. Cheok, A.D., Fong, S.W., Goh, K.H., Yang, X., Liu, W., Farbiz, F.: Human pacman:a mobile entertainment system with ubiquitous computingand tangible interaction over a wide outdoor area. Personal and Ubiquitous Computing **8**(2) (2004) 71–81
2. Klein, K., Drummond, T.: Sensor fusion and occlusion refinement for tablet-based ar. In: Proc. of the ISMAR. (2004) 38–47
3. Henrysson, A., Billinghurst, M., Ollila, M.: Virtualobject manipulation using a mobile phone. In: Proc. of the ICAT. (2005) 164–171
4. Haller, M., Mark Billinghurst, J.L., Leitner, D., Seifried, T.: Coeno-enhancing face-to-face collaboration. In: Proc. of the ICAT. (2005) 40–47
5. Neumann, U., You, S.: Natural feature tracking for augmented reality. IEEE Trans. on Multimadia **1**(1) (1999) 53–64
6. Simon, G., Fitzgibbon, A., Zisserman, A.: Markerless tracking using planar structures in the scene. In: Proc. of the ISAR. (2000) 120–128
7. Chia, K.W., Cheok, A., Prince, S.J.D.: Online 6 DOF augmented reality registration from natural features. In: Proc. of the ISMAR. (2002) 305–313
8. Billinghurst, M., Cambell, S., Poupyrev, I., Takahashi, K., Kato, H., Chinthammit, W., Hendrickson, D.: Magic book: Exploring transitions in collaborative ar interfaces. Proc. of SIGGRAPH 2000 (2000) 87

9. Prince, S., Cheok, A.D., Farbiz, F., Williamson, T., Johnson, N.., Billinghurst, M., Kato, H.: 3d live: Real time captured content for mixed reality. In: Proc. of the ISMAR. (2002) 7–13
10. E.J.Umlauf, Piringer, H., Reitmayr, G., Schmalstieg, D.: ARLib: The augmented library. In: Proc. of the ART02. (2002) TR–188–2–2002–10
11. Claus, D., Fizgibbon, A.W.: Reliable automatic calibration of a marker-based position tracking system. In: Proc. of the WACV. (2005) 300–305
12. Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K.: Virtual object manipulation on a table-top ar environment. In: Proc. of the ISAR. (2000) 111–119
13. Genc, Y., Riedel, S., Souvannavong, F., Akinlar, C., navab, N.: Marker-less tracking for ar: A learning-based approach. In: Proc. of the ISMAR. (2002) 295–304
14. Foxlin, E., Naimark, L.: Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker. In: Proc. of the ISMAR. (2003) 151–160
15. Foxlin, E., Naimark, L.: Vis-traker: A wearable vision-inertial self-tracker. In: Proc. of Virtual Reality. (2003) 199–206
16. Foxlin, E., Altshuler, Y., Naimark, L., Harrington, M.: Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In: Proc. of the ISMAR. (2004) 212–221
17. Kotake, D., Uchiyama, S., Yamamoto, H.: A marker calibration method utilizing a priori knowledge on marker arrangement. In: Proc. of the ISMAR. (2004) 89–98
18. Uematsu, Y., Saito, H.: Ar registration by merging multiple planar markers at arbitrary positions and poses via projective space. In: Proc. of ICAT2005. (2005) 48–55
19. Uematsu, Y., Saito, H.: Vision-based registration for augmented reality with integration of arbitrary multiple planes. In: 13th International Conference on Image Analysis and Processing (ICIAP2005), LNCS 3617. (2005) 155–162
20. Hartley, R., Zisserman, A.: Multiple View Geometry in computer vision. CAMBRIDGE UNIVERSITY PRESS (2000)