Generation of See-Through Baseball Movie from Multi-Camera Views

Takanori Hashimoto #1, Yuko Uematsu #2, Hideo Saito #3

Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
¹takanori@hvrl.ics.keio.ac.jp
²yu-ko@hvrl.ics.keio.ac.jp
³saito@hvrl.ics.keio.ac.jp

Abstract—This paper presents a method of generating new view point movie for the baseball game. One of the most interesting view point on the baseball game is looking from behind the catcher. If only one camera is placed behind the catcher, however, the view is occluded by the umpire and catcher. In this paper, we propose a method for generating a see-through movie which is captured from behind the catcher by recovering the pitcher's appearance with multiple cameras, so that we can virtually remove the obstacles (catcher and umpire) from the movie. Our method consists of three processes; recovering the pitcher's appearance by Homography, detecting obstacles by Graph Cut, projecting the ball's trajectory. For demonstrating the effectiveness of our method, in the experiment, we generate a see-through movie by applying our method to the multiple camera movies which are taken in the real baseball stadium. In the see-through movie, the pitcher can be appeared through the catcher and umpire.

I. INTRODUCTION

Since a multiple-view movie has much more information than a single-view movie, it can provide high presence and reality by generating a 3D movie, a free-viewpoint movie and so on. Especially for the sports event, lots of researches have been studied with multiple-view movies for tracking players and balls [1], [2], [3] and generating a free-viewpoint movie. A free-viewpoint movie enables viewers to change their viewpoints freely. Therefore they can enjoy watching the game from their favorite viewpoints. Soccer games and american football games have been popular research/development targets of generating free-viewpoint movies before. For example, Inamoto et al. have proposed a method of generating a freeviewpoint movie of soccer game by using the view interporation technique of four cameras [4]. The generated movie is also overlaid onto the miniature field model on the tabletop so that the users can watch the game through a HMD [5].

In this paper, we focus on baseball game that is a very popular sport as well as soccer. In Japan, the professional baseball game has been broadcasted almost everyday. One of the most exciting scene is the pitcher's throwing, however, the pitching scene is usually captured only from the viewpoint behind the pitcher with a camera placed at the center field

MMSP'10, October 4-6, 2010, Saint-Malo, France. 978-1-4244-8112-5/10/\$26.00 ©2010 IEEE. screen. In contrast, most of the famous TV games of baseball are designed so that the user's viewpoint corresponds to the catcher's viewpoint. This is because the catcher's viewpoint is very attractive for users. In the real broadcasting of a baseball game, however, it is impossible to capture the pitching from the catcher's viewpoint, because a camera can not be set up at the catcher's head or face.

For generating a movie that is virtually captured at the catcher's viewpoint, in this paper, we propose a method for generating a see-through movie using multiple cameras placed behind the catcher by synthesizing the multiple-view images with the same concept of generating a free-view point movie. Since a lot of cameras are set up in the baseball stadium, it is reasonable to use them. When capturing from just behind the catcher, the pitcher is occluded by the catcher and umpire, and not appeared in the captured image. In order to visualize the pitcher's appearance, we remove the obstacles and generate a "see-through movie".

Diminished Reality (DR) is a technique for removing a real object from an input image by replacing the object with other images [7]. Lepetit et al. have proposed a DR method which tracks and detects occlusions in the specified region and removes the occluder from a movie captured with a single camera [8]. Enomoto et al. synthesize the images captured from multiple viewpoints by planar-projection and detect occluders which are not approximated as a plane [9]. In the method of Zokai et al. [10], they reconstruct the background by creating its model. In these methods, the target scene is limited only if both of the target of recovering (background) and obstacles are static. Therefore these methods can not be applied to the scene in which both of the obstacles and target are dynamic such as a pitcher, catcher and ball in the baseball game movie.

In our method, we remove the moving catcher and umpire by recovering the moving pitcher using the property of baseball. In most cases, the baseball game is captured near the fence, so the pitcher is far from the cameras. Therefore, we can approximate the pitcher's appearance as a plane on the pitcher's plate and recover the pitcher's appearance by transforming the pitcher's region captured with a camera to the other camera's viewpoint by using Homography. In our experimental environment, there are three cameras behind the



Fig. 1. Flowchart

catcher. The pitcher's appearances captured by left and right rear cameras are transformed to the center camera's viewpoint and the transformed images are synthesized. The synthesized image represents the pitcher's appearance watched from the catcher's viewpoint. The obstacles such as the catcher and umpire exsit at the certain position during the game although they are moving. Therefore we can roughly define the region of the obstacles. In our method, we use Graph Cut algorithm [11] to detect the obstacles. The seed is assigned only once, because the uniforms of the catcher and umpire are unchanged. Finally, the synthesized pitcher's appearance is overlaid onto the region which is detected as the obstacles to generate the see-through movie.

II. PROPOSED METHOD

The flowchart of the proposed method is shown in Fig. 1. In our method, we use four cameras to capture the baseball game. One of them is set up behind third base. The other three cameras, which are called center, left, and right cameras respectively, are placed behind the catcher and umpire. The experimental environment is shown in Fig3.

See-through movie is generated based on the center camera by synthesizing the left and right cameras. As shown in Fig. 1, our method consists of three phases; synthesis of the pitcher's appearance, extraction of obstacles (the catcher and umpire), and visualization of ball's trajectory.

As a pre-processing, we compute Homographies of the left and center cameras, and the right and center cameras, respectively. A Trifocal Tensor among three cameras is also



Fig. 2. Specification of synthetic area. (a) for overlaying pitcher's region, (b)(c) for revising detected obstacle region by Graph Cut



Fig. 3. Experimental environment



Fig. 4. Base plane of Homography

obtained by corresponding points. After that the systthesis areas are assigned as shonw in Fig. 2. On the area of Fig. 2(a), the synthesized the pitcher's appearance is overlaid. Fig. 2(b)(c) are used for revising the obstacle area which is extracted by Graph Cut.

At the main processing, all of the cameras capture the baseball game in synchronization. The regions of the catcher and umpire are extracted by Graph Cut. Since the extracted region may not completely cover the obstacle region, the extracted area is revised by using the area of Fig. 2(b)(c). The revised obstacle region is removed from the image. The pitcher's region in the images of the left and right cameras are transformed to the center camera viewpoint by using each Homography. The ball's position is also extracted from the left and right cameras and projected to the center camera viewpoint by using the computed Trifocal Tensor. After that the transformed pitcher's area and the projected ball's appearance are overlaid onto the image of the center camera to generate the see-through movie. These processings are iterated at each frame.

A. Synthesis of Pitcher

When capturing the baseball game for the broadcasting, the cameras are set up near the fence. Since the pitcher's position is far from the cameras, we assume that the pitcher can be approximated as a plane which is perpendicular to the pitcher's plate on the ground as shown in Fig. 4. This plane is call "base plane" in this paper. By using this approximation, therefore, we synthesis the pitcher's appearance by transforming the left and right camera images to the center camera using Homography based on the base plane. The transformed images are synthesized by alpha blending as shown in Fig. 5(a).



Fig. 5. Difference of synthesis results according to the positions of pitcher

(a) using base plane (b) after taking a step

Right camera

Left camera

and planes

Base plane

After pitching

(c) after shift

Shift Right

Shift left



(a) Center camera

(b) Revised obstacle area

Excepted from Graph Cut region

Too detected area



Undetected Detected area catcher's area by Graph Cut

Added to Graph Cut region

(c) Detected region by Graph Cut

(d) Revising detected region



When the pitcher throws the ball, he takes a step from the pitcher's plate. If the pitcher moves from the pitcher's plate, the Homography based on the base plane can not be used for the transformation. For example, the synthesized pitcher's appearance becomes doubly-blurred as shown in Fig. 5(b). Since the width of the step is depending on pitchers, it is impossible and time-consuming to compute homographies based on all candidate planes (steps) in the pre-processing. Therefore we shift the transformed images to left and right according to the pitcher's step as shown in Fig. 6. After such shift processing, the pitcher's appearance can be synthesized as shown in Fig. 5(c). Only by changing the shift amount, we

can adjust every pitcher without extra pre-processing. In our method, we find out whether the pitcher is standing on the pitcher's plate or not by using the side camera, which is set up behind third base. We set a rectangle area on the pitcher's plate and check the variation in the area. If the variation is larger than the threshould value, the shift processing is employed.

B. Detection of Catcher and Umpire

The obstacle regions are extracted by Graph Cut. As described before, the uniforms of the catcher and umpire are unchanged. Therefore the seed of Graph Cut is assigned only once in advance. The detected region is shown by a red line in Fig. 7(c).

The extracted region may not completely cover the obstacle region. The region above the umpire's head should not have been extracted, and the left part of the catcher should have been extracted as shown by a green and blue lines in Fig. 7(c). Therefore, it is necessary to revise the detected regions by using the area of Fig. 2(b)(c).

The purple area in Fig. 7(d) is excepted from the detected Graph Cut region because the area represents not the obstacle area but the center field screen and the batter's head. The yellow area in Fig. 7(d) is added to the detected Graph Cut region, because the catcher often appears in this area when catching the ball. In particular, the appeared catcher's part is detected by background subtraction between the backgound image and the center camera image about this yellow area. The revised obstacle region is shown in Fig. 7(b).

C. Estimation of Ball's Trajectory

Center camera's ball position is projected by using Trifocal tensor with the right and left camera as shown in Fig. 9(a). Ball positions are extracted from the right and left camera images as shown in Fig. 8. We could use some automated technologies for ball extraction and tracking such as [6], but we simply use a manual operation in the experiments in this paper. In this method, we estimate the trajectory of the ball only from the center camera video. Since the optical axis of the center camera is almost horizontal and parallel with the line between pitcher position and the catcher position, we can assume that the horizontal component movement can be estimated from the ball movement in the x direction of the input video, while the vertical component movement can be estimated from the ball movement of the y axis. In this estimation, we assume that

Fig. 6. Image synthesis by right and left shift



(a) Left camera

(b) Right camera

Fig. 8. Ball position of right and left cameras





(a) Before Estimation

(b) After Estimation

Fig. 9. Correction of ball's trajectory

the velocity of the horizontal component is constant, while the acceleration of the vertical component is constant. Based on this assumption, we fit the position of the x component of the ball position to a line, while we fit the position of the y component to two quadrics as shown in Fig. 10. Fig. 9(b) shows the effectiveness of the fitting process of the ball positions.

D. Generation of See-through movie

By using the output obtained with the methods described in II-A and II-C, we generate the see-through movie. The flow of image processing is shown in Fig. 11. First, the obstacle area is overlaid by background image onto the image of the center camera as shown in Fig. 11(b). Second, the pitcher's area is



Fig. 10. Approximation and interpolation of y axis



Fig. 11. Flow of generating a see-through image

overlaid by pitcher's image as shown in Fig. 11(c). Finally, the ball 's appearance are projected as shown in Fig. 11(d). These processing are iterated at each frame.

III. EXPERIMENTAL RESULTS

For demonstrating the effectiveness of our method, in the experiment, we generate a see-through movie by applying our method to the multiple camera movies which are taken in the real baseball stadium. The camera and PC's spec which is used to this experiment is as follows.

- Camera : Canon XHG1(1440 \times 1080)
- CPU : Intel Core 2 Duo 2.80 GHz
- Memory : 3.2GB
- OS : Windows XP Professional

A. Experimental Environment

The experimental environment at Seibu dome is shown in Fig12. The four cameras are set up near the fence.

B. Results

For example, the input image is shown in Fig. 13.

The view of the center camera is interrupted by the umpire and catcher as shown in Fig. 13(b). However, the view of the right and left cameras can see the pitcher's appearance as shown in Fig. 13(a)(c).



Fig. 12. Experimental environment at Seibu dome



Fig. 13. Example of input images

The examples of generated see-through images are shown in Fig. 14, 15 and 16.

By the generating see-through image at the view of center camera using the proposed method, we can see the pitcher appearance which is actually occluded by the catcher and the umpire. Even if the pitcher moves from the pitcher's plate, we can synthesize pitcher by changing the shift amount as described in II-A. We can also adjust the shift amount for every pitcher by changing shift amount dynamically, so that we can synthesize the pitcher image with higher quality.

C. Processing Time

The total processing time for the processes presented from II-A to II-D is approximately 50ms, so that we may achieve 16fps. However, it takes about 1 second for the Graph Cut process. Besides, ball's position extraction from the left and right images is performed by manual operation in this experiment.

IV. DISCUSSION

It is necessary to attack a number of problems for the practical use in baseball broadcast. First, the processing time is a issue. Because we expect that the generated see-thorough movie can be generated a several after the real movie for presenting such as a reply movie that is generally used in the practical broadcasting. However, the processing speed in the experiment is not sufficiently fast for the requirement of the broadcasting. Therefore is it necessary to speed up this method.

One of possible solutions is speeding up of Graph Cut process. By limiting the area of Graph-Cut segmentation into the small area around the umpire and the batter, we can expect that the computational cost can be reduced.

About the quality of the generated see-through movie, we should consider some issues. First issue is improving the accuracy in estimating the ball trajectory. We consider that using a high-speed camera only for ball tracking is one solution. After estimating accurate trajectory with the help of a high-speed camera, we can virtually render the ball image



Fig. 14. Center camera images and see-through images (Middle)



Fig. 15. Center camera images and see-through images (Inside pitch)



Fig. 16. Center camera images and see-through images (Outside pitch)

based on the accurate trajectory. Even in the input video, the ball is not always captured clearly, so such rendering based on the other high-speed camera will enhance the visual quality of the generated see-through video.

Next issue is rendering of the batter. In the experiment, a foot and the bat of a batter is sometimes erased in the obstacle area. We need to recover them. About the bat, we think that using high speed camera is also effectice, since the bat cannot be clearly captured in the input video because of it moves very fast.

Considering about the future extension of the proposed method, we can increase the number of cameras for generating the see-through movie. Some cameras that captures the scene from the upper viewpoint will be improve the quality of the output video. Other possibility will be applying a sort of motion capturing algorithm of the player. The captured motion will improve the quality in rendering the different viewpoint image of the player.

V. CONCLUSIONS

This paper presents a generation method of see-through movie from behind the catcher by using four cameras for using in baseball broadcast. In this method, we extract the occluded area of the pitcher by the catcher and the umpire using Graph-Cut segmentation in the center camera. Then we apply homograpies for rendering the occluded pitcher from the other cameras (right and left cameras). We use the different homographies for the different positions of the pitcher. Furthermore, we estimate the ball trajectory for clearly present the ball positions in the output video.

We demonstrate that the see-through movie in real environment can be successfully generated by using our method. As a future research issues, our method need to be upgraded by reducing processing time and restoring ball/ batter/ foot.

ACKNOWLEDGMENT

This research was supported by National Institute of Information and Communications Technology (NICT).

References

- W. Du, J.-B. Hayet, J. Piater, and J. Verly, "Collaborative multi-camera tracking of athletes in team sports," in Workshop on Computer Vision Based Analysis in Sport Environments, 2006, pp. 2.13.
- [2] M. Xu, J. Orwell, L. Lowey, "Architecture and algorithms for tracking football players with multiple cameras," in Proc. IEE Workshop on Intelligent Distributed Surveillance Systems, 2004, pp. 51.56
- [3] G. Pingali, Y. Jean and A. Opalach, "Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts," in 15th Int. Conference on Pattern Recognition, 2000.
- [4] N. Inamoto, H. Saito, "Intermediate View Generation of Soccer Scene from Multiple Videos," International Conference on Pattern Recognition (ICPR2002), Vol.2, pp.713-716, Aug. 2002.
- [5] N. Inamoto, H. Saito, "Immersive Observation of Virtualized Soccer Match at Real Stadium Model," The Second International Symposium on Mixed and Augmented Reality (ISMAR03), pp.188-197, Oct.2003.
- [6] M. Takahashi, T. Misu, M. Tadenuma, N. Yagi, "Real-Time Ball Trajectory Visualisation using Object Extraction," CVMP2005, 2005, pp. 62-69.
- [7] J. Wang, E.Adelsen, "Representing moving images with layers," IEEE Transactions on Image Processing Special Issue, Image Sequence Compression, September 1994.

- [8] V. Lepetit and M.-O. Berger, "A semi-automatic method for resolving occlusion in augmented reality," In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), June 2000.
- [9] A. Enomoto, H. Saito, "Diminished Reality using Multiple Handheld Cameras," Proc. ACCV'07 Workshop on Multi-dimensional and Multiview Image Processing, pp.130-135, 2007.
- [10] S. Zokai, J. Esteve, Y. Genc and N. Navab, "Multiview Paraperspective Projection Model for Diminished Reality," Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003).
- [11] Z. Garrett and H. Saito, "Real-Time Online Video Object Silhouette Extraction Using Graph Cuts on the GPU," 15th International Conference Image Analysis and Processing, September 8-11, 2009, Lecture Notes in Computer Science, Volume 5716/2009, pp. 985-994.