Virtual Viewpoint Replay for a Soccer Match by View Interpolation From Multiple Cameras

Naho Inamoto and Hideo Saito, *Member, IEEE*

***Abstract*—This paper presents a novel method for virtual view synthesis that allows viewers to virtually fly through real soccer scenes, which are captured by multiple cameras in a stadium. The proposed method generates images of arbitrary viewpoints by view interpolation of real camera images near the chosen viewpoints. In this method, cameras do not need to be strongly calibrated since projective geometry between cameras is employed for the interpolation. For avoiding the complex and unreliable process of 3-D recovery, object scenes are segmented into several regions according to the geometric property of the scene. Dense correspondence between real views, which is necessary for intermediate view generation, is automatically obtained by applying projective geometry to each region. By superimposing intermediate images for all regions, virtual views for the entire soccer scene are generated. The efforts for camera calibration are reduced and correspondence matching requires no manual operation; hence, the proposed method can be easily applied to dynamic events in a large space. An application for fly-through observations of soccer match replays is introduced along with the algorithm of view synthesis and experimental results. This is a new approach for providing arbitrary views of an entire dynamic event.**

***Index Terms*—Dynamic event, multiple cameras, projective geometry, soccer match, view interpolation, virtual view synthesis.**

I. Introduction

DEVELOPMENT of information and communication technology has enabled us to enjoy viewing sporting and entertainment events from across the world. In addition to relaying broadcasts of events, present day television broadcasting also offers a variety of visual entertainment effects. An example of these effects is the "Eye Vision" system that was used for the Super Bowl XXXV broadcast by CBS. In this system, multiple video streams are captured using more than 30 cameras. The sequences of video images from different angles are then used to create a 3-D visual effect such that the viewpoint revolves around the object event at a temporally frozen moment. This system uses visual effects by simply switching the video camera images, whereas computer vision based technology can provide more attractive visual effects such as synthesizing arbitrary view images for virtual viewpoint movements.

Manuscript received September 16, 2004; revised January 24, 2007. This work was supported in part by a grant from the Japan Society for the Promotion of Science and by a grant from the 21st century Center of Excellence for Optical and Electronic Device Technology for Access Network from the Ministry of Education, Culture, Sports, Science, and Technology, Japan. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Suh-Yin Lee.

The authors are with the Department of Information and Computer Science, Keio University, Yokohama 223-8522, Japan (e-mail: nahotty@ozawa.ics.keio.ac.jp; saito@ozawa.ics.keio.ac.jp).

Digital Object Identifier 10.1109/TMM.2007.902832

Virtualized reality [1], which is a pioneering project in this field, has achieved such virtual movements for dynamic scenes using computer vision technology. Three-dimensional models of objects in the target scene are reconstructed from multiple view images; subsequently, colors in the real images are used to synthesize the texture of the 3-D models. Using conventional rendering techniques, new view images are generated from the color-textured 3-D models.

Techniques for providing visual effects for dynamic events in a large space have recently been proposed [2], [3]. They enable viewers to view a specific player on a large field from viewpoints around the player. In [2], the shape of the 3-D object is described by a set of planes in order to effectively synthesize a novel view of the object. In [3], a special 3-D coordinate system, established by epipolar geometry of cameras, is used for the reconstruction of the 3-D model without camera calibration. The target area in these methods is a certain local area in a large space, where a few players are present. Alternatively, the method for arbitrary viewpoint movement at a soccer match has been proposed [4]. In this method, players are represented by simplified 3-D models, which are reconstructed using multiple views, and then the virtual view images of the players are presented in a virtual stadium. Although viewers can watch the entire soccer scene from arbitrary viewpoints, the presented stadium is not a real stadium but a computer-generated virtual model.

Our objective is to realize the virtual view synthesis for fly-throughs in an actual sporting event held in a stadium. The entire scene including the players, field, and stadium is a reconstruction target, i.e., the object area is larger than that of the previous methods described in [1]–[3]. Moreover, the stadium at virtual viewpoint should also be synthesized using captured scenes rather than computer-generated model.

In this paper, we propose a novel approach for virtual view generation of entire soccer scenes that are captured using multiple uncalibrated cameras in a real stadium. Without using 3-D models, only the projective geometry between neighboring cameras is used to synthesize new view images [5]–[7]. View interpolation [8] can be used to reconstruct the entire soccer scene from any intermediate viewpoint among the real cameras. Firstly, the projective geometry between the neighboring cameras is obtained from image sequences. The soccer scenes are then classified into several regions and appropriate projective transformations are applied to each region to generate intermediate view images. By superimposing the intermediate images for all the regions, the global appearance of the entire soccer scene from the virtual viewpoints can be synthesized to facilitate photorealistic presentation.

In addition, we introduce the "Viewpoint on Demand System" for soccer match replays. Existing television broadcasts only deliver pre-produced contents wherein producers manually select video cameras for relaying sporting events; this is essentially a one-way communication. On the other hand, the Internet facilitates an interactive communication between the broadcasting station and the viewers, in which the contents can be interactively modified according to the viewers' demands. If the viewers can select preferred viewpoints, they will derive great enjoyment from watching the exciting scenes in these events. We demonstrate the viewpoint on demand system as an example of such interactive communication media. Using the proposed system, users can freely select their preferred viewpoints while watching a match. They can focus on a specific player in close-up view or may track the ball movement using a zoom-out virtual camera.

This paper is organized as follows. In Section II, the related work on virtual view synthesis is reviewed. Representative methods are introduced in three approaches. The overview of the proposed method is described in Section III. Section IV explains how to estimate projective geometry used for view interpolation. Subsequently, the technique for view interpolation of the entire scene in a large-scale event is proposed in Section V. Section VI shows the experimental results, and then the viewpoint on demand system is proposed in Section VII. After discussions come up in Section VIII, we finally summarize our work in Section IX.

II. Related Work

In the field of computer vision, the techniques for synthesizing virtual view images from a number of real camera images have been studied since the 1990s [9]–[11]. These techniques, termed image based rendering (IBR), can be categorized into three groups, model based approach, transfer based approach, and approach using the plenoptic function. By using the model based approach, it is possible to construct 3-D models of objects to generate the desired view. As described in the previous section, the virtualized reality [1] project at CMU, color-textured 3-D models are reconstructed for synthesizing movies at arbitrary viewpoints. Wheeler *et al.* [12] have also proposed a method for 3-D reconstruction using multiple view range images. The quality of the virtual view image generated by these methods depends on the accuracy of the 3-D model. A large number of video cameras surrounding the object or range scanners are used to construct an accurate model. Furthermore, camera calibration [13] is usually required to relate 2-D coordinates in images to their corresponding 3-D coordinates in object space. As it is essential to measure the 3-D positions of several points in the object space, calibration becomes difficult especially in a large space. For these reasons, the object area is generally limited to a few cubic meters in this approach.

On the other hand, by using the transfer based approach, it is possible to synthesize arbitrary view images without an explicit 3-D model. Seitz and Dyer [14] have used morphing techniques [15] to synthesize new viewpoints between a pair of images for a static scene. Chen and Williams [8] have also proposed the method for interpolation of intermediate views at an interactive rate by the morphing method. Avidan and Shashua [16] have employed a trifocal tensor for image transfer. In these methods, dense correspondence between the original images is required to generate intermediate views. The correspondence is often generated manually or by optical-flow; hence, almost all the targets are static images or slightly varying images such as facial expressions.

More recently, Manning and Dyer have extended view morphing [14] to rigid objects with translation, which is called dynamic view morphing [17]. Wexler and Shashua have proposed another technique to morph a dynamic view with a moving object along a straight line path from three viewpoints [18]. While the above two methods have only dealt with translation, Xiao *et al.* have extended the view morphing technique to a rotation case and applied it to non rigid objects with complicated motion [19]. All of these methods calculate motion parameters of the objects in order to interpolate the appearance of the moving objects. It is not practical to apply these methods to a scene that contains multiple objects with complicated movements such as a sporting match.

As regards approach using the plenoptic function, which describes all the radiant energy that is perceived by an observer at any point in space and time, it is possible to create novel views from a collection of sample images. This allows a user to arbitrarily pan and tilt a virtual camera and interactively explore his/her environment. In its most general forms, the plenoptic function is a seven-dimensional function. Due to its high dimensional nature, data reduction or compression of the plenoptic function is essential. The light field of Levoy and Hanrahan [20] and the lumigraph of Gortler *et al.* [21] are simplified the function with four dimensions. Recently, Shum *et al.* [22] proposed a new IBR technique called concentric mosaic for virtual reality applications. They proposed the 3-D plenoptic function and the compression algorithm of concentric mosaic. This approach provides much better image quality and lower computational requirement for rendering than the model based approach. However, it is inadequate for large-scale events because it is impossible to describe all the radiant energy.

In a related approach, Connor *et al.* have proposed a method using layered representation for dynamic view synthesis between a pair of images [23]. Foreground objects are represented as multiple layers with one background. Subsequently, by estimating the parameters of the layered model, the new view image is generated. In this method, an approximate selection of the corresponding regions in the initial frame is necessary for layer representation. The number of layers does not vary over time; hence, it cannot be applied to a long image-sequence. On the other hand, by applying our proposed method, we can automatically synthesize the virtual view image for dynamic regions and represent the entire soccer scene in each frame. We apply the method to several image sequences spanning a few minutes. Furthermore, this paper performs view interpolation among three views.

III. Overview of the Proposed Method

Images of arbitrary viewpoints are generated by view interpolation among real camera images. Two or three cameras near the virtual viewpoint chosen by a user are selected from multiple cameras. The virtual viewpoint image is generated through correspondence among the selected cameras. As our target is dynamic events in a large space, we divide the object scene into

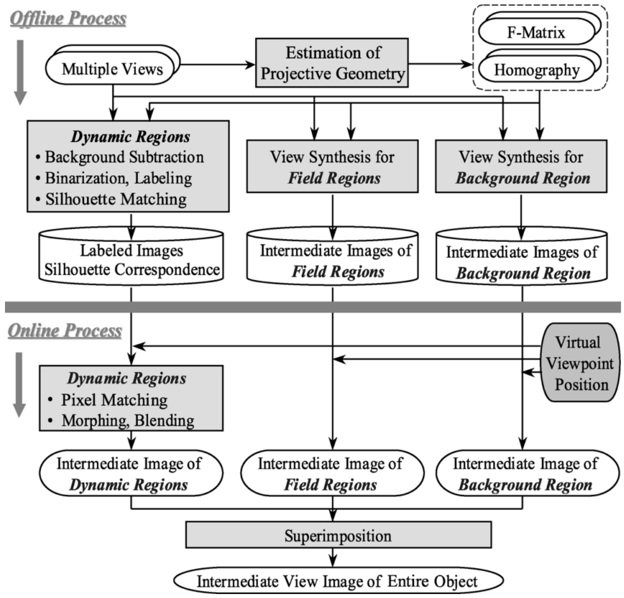


Fig. 1. Overview of the proposed method. Offline process begins with the estimation of projective geometry and then intermediate view images for field regions and a background region are synthesized. A part of the process for dynamic regions is also implemented offline. Online process proceeds based on the virtual viewpoint position chosen by viewers. After view synthesis for dynamic regions, superimposition completes the generation of intermediate view images of entire object.

dynamic regions and static regions. View interpolation is then performed for each region independently.

Fig. 1 describes an overview of the proposed method. Our approach is as follows. Firstly, the projective geometry between multiple views, which are fundamental matrices [24] and homographic matrices [24], are estimated in a certain frame selected in the image sequences. The soccer scene is then classified into dynamic and static regions by background subtraction for performing view interpolation.

According to the properties of soccer scenes, the static regions can also be divided into several plane regions. One is the background region, which can be approximated as an infinitely distant plane. The others are field regions such as the ground plane and the goal, which can be approximated as sets of planes. Intermediate view images are synthesized through homographic transformations of the each plane region. Since background region and field regions are considered as stable, they are detected manually, and virtual view images are generated for all intermediate viewpoints only once, in advance. Although all of the audiences' movements may not be captured, we do not regard this as a problem because audience movement is not essential to soccer scene representation. If the captured scene has variations in lighting, the background image is generated for every lighting condition of the captured sequence as explained in Section V-B.

As regards the dynamic regions, view interpolation at each frame is necessary because the shape or position of an object changes over time. Our method, however, combines offline and online processes in order to efficiently render the scene. In the offline process, every player region is segmented and labeled automatically. The labeled regions of the same player in the neighboring views are corresponded through homographic transformations of the ground plane among the views. In the online process, applying fundamental matrix obtains dense correspondence for every labeled region and the ball region. Morphing technique synthesizes the intermediate view images from the reference camera images. If the captured scene has shadows of players and ball, the intermediate images for shadows are also generated. Finally, by superimposing the intermediate images in the order of background region, field regions, and dynamic regions, we complete the entire virtual view of the soccer scene at the viewpoint chosen by the user.

IV. Estimation of Projective Geometry

A. Fundamental Matrix

The epipolar geometry between two cameras is represented by the fundamental matrix (denoted as F-matrix below) $\boldsymbol{F}$, which is a 3×3 matrix. If a point P in 3-D space is projected to a point p_1 in the first view and a point p_2 in the second, the corresponding image points satisfy the following equation:

$$\tilde{\boldsymbol{p}_2}^\top \boldsymbol{F} \tilde{\boldsymbol{p}_1} = 0 \quad (1)$$

where $\tilde{\boldsymbol{p}_1}$ and $\tilde{\boldsymbol{p}_2}$ are the homogeneous coordinates of p_1 and p_2, respectively. $\boldsymbol{F}$ is a rank 2 homogeneous matrix with 7 degrees of freedom; hence, it can be computed nonlinearly by at least seven correspondences in the two views. Considering the search for corresponding points in stereo matching, the search area can be reduced by this geometry. Assuming that a point x is known in the first view, the corresponding point in the second view must lie on the epipolar line l obtained by

$$\tilde{\boldsymbol{l}} = \boldsymbol{F} \tilde{\boldsymbol{x}} \quad (2)$$

where $\tilde{\boldsymbol{l}}$ and $\tilde{\boldsymbol{x}}$ represent the homogeneous coordinates of l and x, respectively. Therefore, the search does not need to cover the entire image plane and can be restricted to the epipolar line. In the proposed method, the F-matrix is employed for obtaining a dense correspondence for the dynamic regions.

B. Homography

Image points on a plane in the first view are related to their corresponding image points in the second view using a homographic matrix $\boldsymbol{H}$, induced by a world plane, as

$$s\tilde{\boldsymbol{p}_2} = \boldsymbol{H} \tilde{\boldsymbol{p}_1} \quad (3)$$

where $\tilde{\boldsymbol{p}_1}$ and $\tilde{\boldsymbol{p}_2}$ are the homogeneous coordinates of the corresponding image points and s is the scale factor. $\boldsymbol{H}$ is a 3×3 matrix with 8 degrees of freedom; hence it can be computed by at least four correspondences in the two views. Through a homographic transformation, a point in one view determines a point in the other. The proposed method employs homographic transformations for obtaining dense correspondences in the static regions.

V. View Interpolation

A. Static Regions

The method for view interpolation in each region is described below. For simplicity, we consider the case of interpolation between two views. This method can be also applied in the case of three views (see Section VI).

As the static regions are considered to undergo little or no changes over time, view interpolation is implemented only once in the selected frame, where neither players nor the ball is present. If such an image is not contained in the captured image sequence, it can be constructed by setting the mode value of the image sequence to each pixel. The image that does not include dynamic objects is thus generated for each camera. If the captured scenes have variations in lighting, the background image needs to be generated for every lighting condition in the sequence. In our experiment, we synthesized the background image every 150 frames for the image sequence beforehand.

The static regions are then manually classified into field regions and a background region. This manual segmentation is necessary only once because the cameras are fixed. In the field regions, including the ground and the goal, a dense correspondence is obtained by applying a homographic transformation to each plane. Their positions are transferred by image morphing for generating a new view. For the background region, including spectator seats, partial area images are extracted from the panoramic image compounded from the background of multiple views. As the intermediate viewpoint position is defined by reference cameras and the interpolating weight, by changing the weights gradually, we can synthesize the virtual view images of the static regions for every possible viewpoint.

1) Field Regions: In a soccer scene, the ground and the goal can be considered as a single plane and a set of planes, respectively. We then apply homography to the planes to obtain the correspondences required for the generation of intermediate view images. Equation (3) yields the pixel-wise correspondence for two views of a plane. The homographic matrices of the plane that represents the ground and goal provide the dense correspondence within these regions. We first generate the two interpolated images at the same virtual viewpoint using the two directed correspondences, from view 1 to view 2 and from view 2 to view 1, separately. Then, the two warped images are blended into a single image. In order to warp the image, the position and the value of the pixel are transferred by image morphing as described by the following equations:

$$\acute{\boldsymbol{p}} = (1-\alpha)\boldsymbol{p}_1 + \alpha \boldsymbol{p}_2 \tag{4}$$

and

$$I(\acute{\boldsymbol{p}}) = (1-\alpha)I(\boldsymbol{p}_1) + \alpha I(\boldsymbol{p}_2) \tag{5}$$

where $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are the coordinates of the corresponding points in images I_1 and I_2, respectively, and $I(\boldsymbol{p}_1)$ and $I(\boldsymbol{p}_2)$ are the pixel values of the corresponding points in images I_1 and I_2, respectively. $\acute{\boldsymbol{p}}$ represents the interpolated coordinates and $I(\acute{\boldsymbol{p}})$ represents the interpolated pixel value. α defines the interpolating weight assigned to the respective actual viewpoints as shown in Fig. 2. After two warped images are generated using above process, they are blended into a single image, which is the target image at the intermediate viewpoint. In blending the two images, if the color of a pixel differs between these images, the corresponding pixel in the virtual view is rendered with the average of the colors; otherwise, the rendered color is taken from either of the actual images. Fig. 3 presents examples of generated intermediate images for the field regions. Fig. 3(a) and (d) shows the real camera images, and (b) and (c) shows the interpolated images from (a) and (d). The interpolating weight of the virtual view to the real views is 4 to 6 in (b) and 6 to 4 in (c).

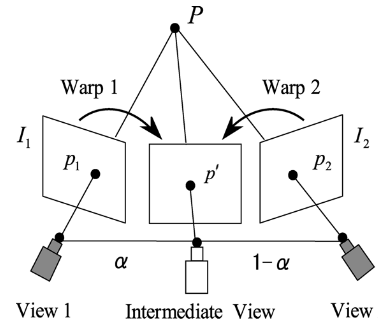


Fig. 2. Image morphing for transfer of the correspondence. The coordinates and pixel value of point $\acute{\boldsymbol{p}}$ are determined by those of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ with interpolating weight α. Warp from view 1 and view 2 results in the transfer of the correspondence to the intermediate view.

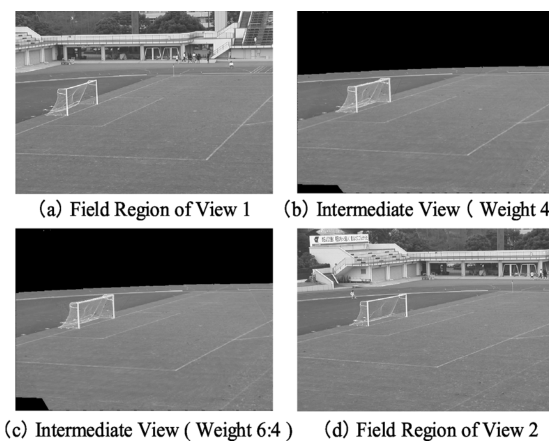


Fig. 3. Examples of the intermediate images for the field regions. (a), (d) Real camera images. (b), (c) Interpolated images from (a) and (d), where the relative weight of the virtual view to the real view is 4 to 6 in (b) and 6 to 4 in (c).

2) Background Region: The background is placed in the regions that are at a distance from the viewpoint positions of the cameras such that it can be considered as a single, infinitely distant plane. We compose images from each of the two real viewpoints in order to generate mosaics, which are the respective panoramic images of the background. Here, we assume that the backgrounds of the neighboring viewpoints have an overlapping region. Intermediate view images are extracted from these panoramic images.

We begin the composition by integrating the coordinate systems of the two views through the homographic matrix $\boldsymbol{H}_b$, which represents transformation from the first view to the second view, for the background. Next, we blend the pixel values of the overlapping area so that the pixel colors at the junction areas can smoothly connect the two backgrounds. The pixel value in the mosaic image is given by the following equation:

$$\acute{v} = \begin{cases} v_1 & (x < x_1) \\ (1-\beta)v_1 + \beta v_2 & (x_1 \le x \le x_2), \\ v_2 & (x > x_2) \end{cases} \tag{6}$$

where

$$\beta = \frac{x - x_1}{x_2 - x_1},$$

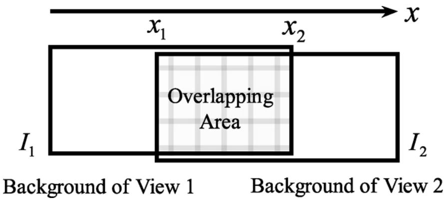


Fig. 4. Image mosaicing. Pixel value of overlap area is determined by blending original pixel value of view 1 and view 2 in accordance with the distance from the boundaries, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

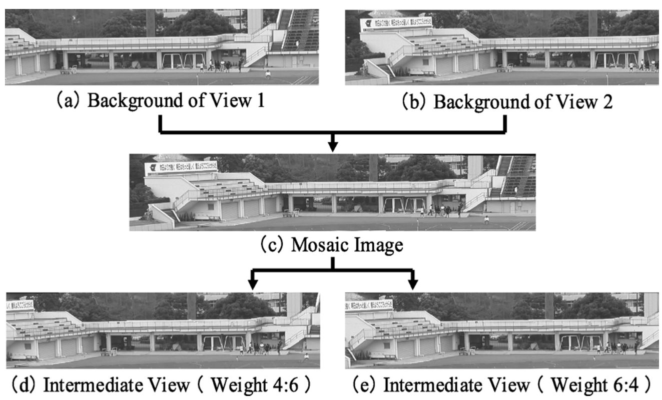


Fig. 5. Examples of the intermediate images for the background. (a), (b) Real camera images. (c) Mosaic image composed of (a) and (b). (d), (e) Intermediate images, whose interpolating weight is 4 to 6 in (d) and 6 to 4 in (e).

v_1 and v_2 are the pixel values of I_1 and I_2, and x_1 and x_2 are the x-coordinates of the left hand side and the right hand side of the overlapping area, respectively (as shown in Fig. 4). The partial area that is necessary for each virtual view is then extracted from the panoramic image. The following homographic matrix $\acute{\boldsymbol{H}}_b$ is then used in the transformation of coordinates to complete the intermediate view of the background region.

$$\acute{\boldsymbol{H}}_b = (1 - \alpha)\boldsymbol{E} + \alpha \boldsymbol{H}_b \, , \tag{7}$$

where α is the interpolating weight and $\boldsymbol{E}$ is a 3×3 unit matrix. Fig. 5(a) and (b) illustrate the examples of background regions in real camera images, and (c) shows a mosaic image composed of (a) and (b). Fig. 5(d) and (e) present intermediate images for the background region, whose interpolating weight is 4 to 6 in (d) and 6 to 4 in (e).

B. Dynamic Regions

The method of view interpolation for the dynamic regions is explained below. In these regions, as the shapes or the positions change over time, view interpolation is implemented for each frame. The process is categorized into offline and online processes for effectivity. In the offline process, all the dynamic regions in every frame are extracted by subtracting the background from the original image. The image where neither the players nor the ball exists is used as the background of each camera. If view interpolation is applied to the sequence that has variations in lighting, we select a background with the same light condition. The segmentation of dynamic regions and static regions is sometimes difficult. Therefore, we extract dynamic regions by background subtraction using not only intensity but also color vector, which has three components: red, green, and blue. They are considered to be identical in pixel assigned to the static region between current frame image and background image while they vary in pixel of the dynamic region. Fig. 6 shows the result of background subtraction. The dynamic regions are greatly extracted by the above method. Although this segmentation is necessary for applying view-interpolation, we do not address the issue directly. That is because the main objective in this paper is to produce video effect of virtual viewpoint replay on the condition that dynamic regions are correctly extracted.

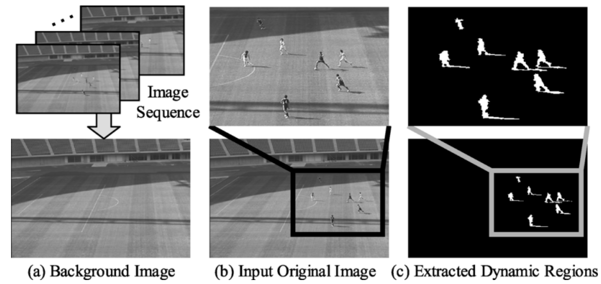


Fig. 6. Extraction of the dynamic regions. (a) Background image that are synthesized from the image sequence. (b) Input original image (bottom) and the close-up view (top). (c) Extracted dynamic regions (bottom) and the close-up view (top).

As the extracted regions usually contain several players and a ball, and possibly shadows, we deal with these dynamic objects separately. If shadows are included in the object scene, we first segment the shadow regions and the player/ball regions. Both the geometric information and the color information are used for this segmentation. It is assumed that the shadow is usually projected on the ground in a soccer scene. We detect a candidate for shadow regions by applying homography of the ground plane to all the extracted dynamic regions in neighboring two view images. This detection based on the homography often includes a part of player's foot. Therefore, we also use the pixel color for shadow extraction by applying HSI transform to the candidate in each view image. The hue of the pixel is almost identical in the shadow regions between the current frame image and the background image, while it is different in the player/ball regions. Fig. 7 exhibits the segmentation results, where the above method, which is the combined method of geometric transform; homography transform and color transform; HSI transform, is compared with the method using only homographic transform or HSI transform. It is evident that the combined method is better than the independent methods at segmenting the dynamic regions into shadows and players/ball.

After segmentation, view interpolation is applied to the shadow and the player/ball regions, respectively. Using the classical method, it is possible to synthesize shadows in another view by estimating the light sources in an environment; however, this is performed at a high cost of calculation. Alternatively, in the proposed method, we can project shadows on the intermediate view image by transferring the shadow regions from the reference images using projective geometry. Since the shadows are considered to be projected on the ground, homographic transformation is applied to the shadow regions as well as the field regions. The intermediate view images for

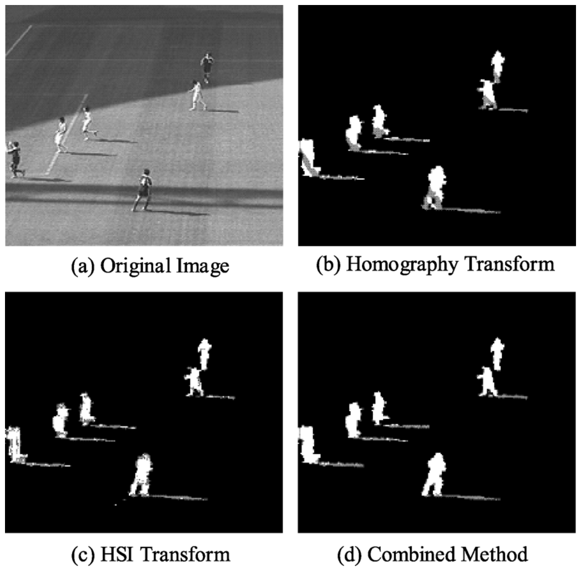


Fig. 7. Comparison of segmentation results for dynamic regions. (a) Original image. (b)–(d) Segmented results. Gray indicates the shadow regions and white indicates the players/ball regions. (b) Result using homography transform. (c) Result using HSI transform. (d) Result using combined method.

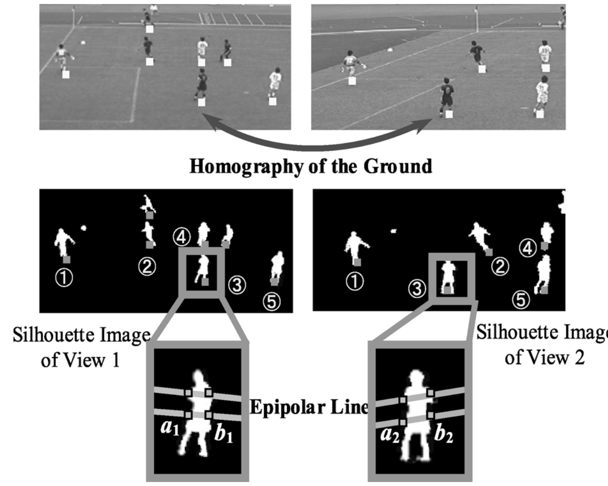


Fig. 8. Correspondence for the dynamic regions. Homography of the ground plane results in silhouette correspondence, and epipolar lines drawn by F-matrix facilitate the computation of dense correspondence within the silhouette.

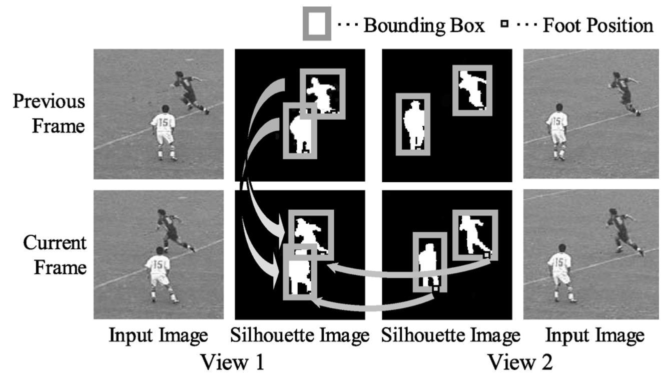


Fig. 9. Region correspondence in case of occlusion. The foot position of the occluded player is calculated by homography of the ground plane from neighboring view. The player region is projected from the previous frame image from the same viewpoint.

shadow regions are synthesized using the homography of the ground plane as explained in Section V-A.

Next, we generate the virtual view image for the player/ball regions. The labeling process is used to segment each player and the ball. Subsequently, the corresponding silhouettes are obtained using the homography of the ground plane as shown in Fig. 8. This is based on the assumption that one foot of a player is always in contact with the ground. Even if a player jumps, the error caused by the jump is sufficiently small; therefore, the homographic matrix of the plane that represents the ground can still locate the corresponding silhouettes. Some players, however, may not have one to one correspondence due to occlusion. In such a case, the segmented silhouettes in the previous frame are used for the segmentation of the players. As shown in Fig. 9, the foot position of the occluded player is calculated by the homography of the ground plane from the neighboring view. The bounding box (rectangle surrounding the segmented player) is then projected onto the current frame from the previous frame. Thus, the occluded player can also have a correct correspondence. If the occlusion is detected in both the views, the players are treated as one large object. For the online process, both the labeled images and the silhouette correspondence are stored at every two neighboring viewpoints. This completes the offline process.

The online process proceeds using stored information such as labeled images and silhouette correspondence regarding the two reference viewpoints near the chosen virtual viewpoint. By drawing epipolar lines in two different views, view 1 and view 2, using a F-matrix, we obtain the pixel correspondence within the silhouettes. On each epipolar line, the correspondences of intersections with boundaries, such as a_1 and a_2, and b_1 and b_2 of Fig. 8, are obtained first. The correspondences within the silhouette are obtained by linear interpolation of the intersection points. After a dense correspondence for the entire silhouette is obtained, the pixel positions and values are transferred from the source images of view 1 and view 2 to the target image by image morphing in the same manner as in the field regions. However, view interpolation only generates intermediate view images, where the zoom ratio is identical to that of real cameras. In order to provide zooming effects in free-viewpoint observation, it is necessary to control the 3-D position of the virtual camera or it's focal length. As the proposed method, which is based on view interpolation, cannot directly deal with the extrinsic and intrinsic parameters, we deal with a zooming feature for expanding or contracting images. View interpolation is modified as given by the following equation, instead of (4):

$$\acute{\boldsymbol{p}} = (1-\alpha)\left\{(\boldsymbol{p}_1 - \boldsymbol{c}_1)\frac{\acute{f}}{f_1} + \boldsymbol{c}_1\right\} + \alpha\left\{(\boldsymbol{p}_2 - \boldsymbol{c}_2)\frac{\acute{f}}{f_2} + \boldsymbol{c}_2\right\} \tag{8}$$

where $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are the coordinates of the principal points in images I_1 and I_2, respectively, and f_1 and f_2 are the focal lengths of cameras 1 and 2, respectively. $\acute{f}$ represents the focal length of the virtual camera. This equation enables zooming in or out approximately by expansion and contraction using the ratio of the focal length of the real camera to the focal length of the virtual camera. The pixel value is transferred using (5). Virtual

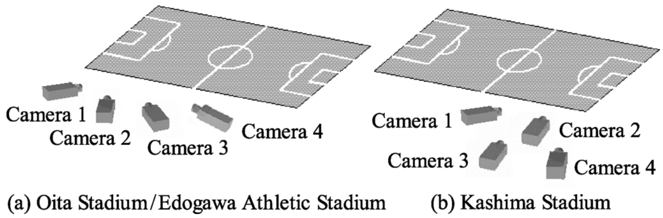


Fig. 10. Camera configuration at the stadiums. (a) Oita Stadium/Edogawa Athletic Stadium; all cameras were set at the same height. (b) Kashima Stadium, cameras 1 and 2 were set at positions higher than those of cameras 3 and 4.

views are generated by blending the two warped images. The above algorithm is applied to every pair of silhouettes. After synthesizing them in order of distance from the viewpoint, all player/ball regions are overlaid onto the shadow regions. This concludes view interpolation for dynamic regions. Finally, superimposition of the images, in the order of background region, field regions, and dynamic regions, completes the virtual view image of the entire scene for the chosen viewpoint.

VI. Experimental Results

We have applied the proposed method to several image sequences of actual soccer matches captured using multiple video cameras at three kinds of soccer stadiums; the Edogawa Athletics Stadium in Tokyo, the Kashima Stadium in Chiba, and the Oita Stadium in Oita, Japan. As shown in Fig. 10, a set of four fixed cameras was placed on one side of the soccer field in all three stadiums in order to capture the penalty area. Neighboring cameras require an overlapping region of the background for image mosaicing. The captured videos were converted to BMP format image sequences, composed of 720×480 pixel, 24-bit-RGB color images, and were then used for virtual view synthesis.

The fundamental matrices between the viewpoints of the cameras and the homographic matrices between the planes in the neighboring views were computed using the corresponding points. In this experiment, we manually selected about 50 corresponding points, whose 3-D positions varied in object space, for fundamental matrices and 20 points on each plane for homographic matrices in the image sequence between neighboring views.

Fig. 11 presents some results of the generated intermediate view images for soccer scene at the Edogawa Athletic Stadium. Fig. 11(a)–(d) presents images captured using real cameras and the others present virtual view images generated by the proposed method. The position of players and the location of the background gradually change depending on the angle of the virtual viewpoint, which is determined by the interpolating weight between two real camera viewpoints. For example, the virtual viewpoint of (e) is located at a position whose relative weight is 2:8 between cameras 1 and 2. Although our method involves the rendering of separated regions, the synthesized images appear very realistic due to which the boundaries between the regions are not visible. Fig. 12 presents the reconstruction of the player from different angles. Not only the global appearance of the entire scene but also the local appearance of the player can be represented to a great extent.

We also have experimental results for evaluation. As seen in Fig. 13, the proposed method is applied to two view computer-generated images drawn by OpenGL, where four cuboids are placed on one plane. Fig. 13(c) shows the synthesized images generated by the proposed method from (a) and (b) with an interpolating weight value of 0.5. This result is synthesized by superimposing the virtual view image for the cubical region on the virtual view image for the plane region. Fig. 13(d) shows the image drawn by OpenGL from the same viewpoint as (c). The color difference between (c) and (d) is presented in (e). Although errors can be seen on the edges of the objects, most of the areas in the synthesized image are almost identical in appearance. This result indicates that the proposed method represents the objects at the correct positions in the intermediate view image with certain color differences. The pixel correspondence error is responsible for a significant part of the color differences.

Fig. 14 compares the virtual and real camera images for a soccer scene captured using cameras at the Edogawa Athletic Stadium. Fig. 14(a) and (c) shows the virtual view image generated from the real view images captured using cameras 2 and 4. Fig. 14(b) and (d) shows the real view image captured using camera 3. The virtual camera is placed at the position where the interpolating weight is 5 to 5 between cameras 2 and 4, which is close to but does not coincide with the position of camera 3. By comparing the virtual and real views, we see that the realistic image is obtained without distortion or holes. The player regions and the field regions captured by the two real cameras have been correctly reconstructed in the virtual view image. Slight differences in the position of the players arise from the difference in the viewpoint positions of the virtual camera and real camera.

Next, we have applied the proposed method to three view images captured at the Kashima Stadium. In the case of view interpolation among the three views, the viewpoint position is determined by weight α and weight β, as shown in Fig. 15. The virtual view image is synthesized from three real view images by morphing as in the case of two views. The following equations are used instead of (4) and (5):

$$\acute{\boldsymbol{p}} = (1-\alpha)(1-\beta)\boldsymbol{p}_1 + \alpha(1-\beta)\boldsymbol{p}_2 + \beta\boldsymbol{p}_3 \quad (9)$$

and

$$I(\acute{\boldsymbol{p}}) = (1-\alpha)(1-\beta)I(\boldsymbol{p}_1) + \alpha(1-\beta)I(\boldsymbol{p}_2) + \beta I(\boldsymbol{p}_3) \quad (10)$$

where $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$ are the coordinates of the corresponding points in images I_1, I_2, and I_3, respectively and $I(\boldsymbol{p}_1)$, $I(\boldsymbol{p}_2)$, and $I(\boldsymbol{p}_3)$ are the pixel values of the corresponding points in images I_1, I_2, and I_3, respectively. When the number of reference camera is three, blending the color of the reference images for all points may blur the virtual view image. We then use the pixel value of the nearest camera for the edge points. Fig. 15 presents the results of synthesized images from three view images. The soccer scene including the shadows from the real three viewpoints is well represented from the virtual viewpoints.

We have also obtained results for other scenes including shadows captured at the Oita Stadium, where view interpolation is performed between two views (see Fig. 16). Fig. 16(c) shows the resultant image when view interpolation is applied to shadow and player/ball regions after segmentation, while

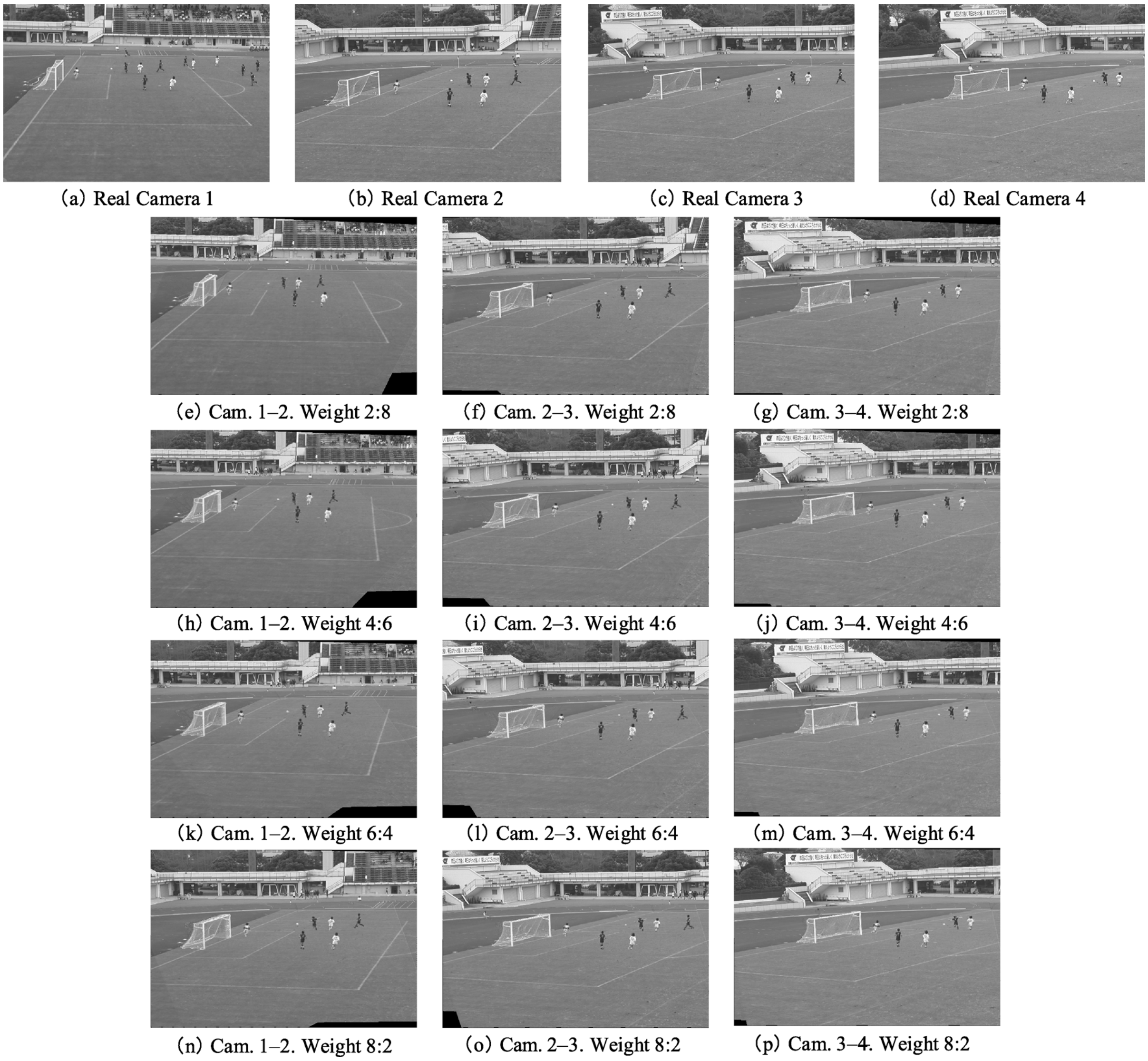


Fig. 11. Synthesized virtual view images for the entire soccer scene from one frame captured at the Edogawa Athletic Stadium. (a)–(d) Real camera images. (e), (h), (k), and (n) Interpolated images between cameras 1 and 2. (f), (i), (l), and (o) Interpolated images between cameras 2 and 3. (g), (j), (m), and (p) Interpolated images between cameras 3 and 4. The interpolating weight to the reference cameras is shown under each image.

Fig. 12. Reconstruction of the player from different angles.

(d) shows the result without segmentation. Although Fig. 16(d) lacks part of or the entire shadows of the players, all shadows are projected correctly in (c). This shows that by warping the shadow regions, we can successfully represent them in another viewpoint.

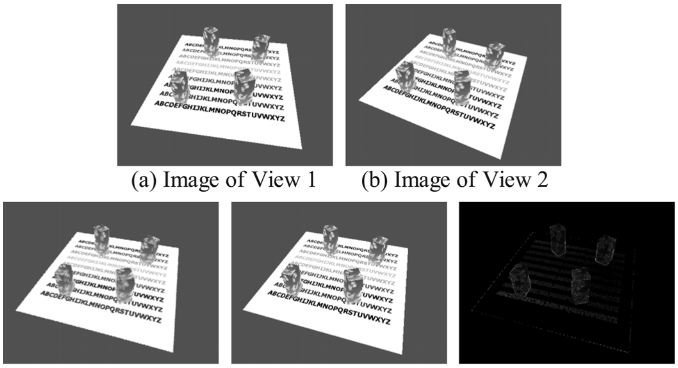
Fig. 17 presents an example of the image sequence in a virtual viewpoint replay. Free-viewpoint observation is implemented by selecting the reference cameras and the interpolating weight for every frame. Frames 1462 and 1468 contain some occlusions, but the occluded players are constantly tracked and their appearance is well synthesized by using both the neighboring camera information and the previous frame information.

Finally, we have produced a video[1] that gives viewers the impression of flying through the soccer field or playing in the soccer match by changing the position of the viewpoint according to the ball movement. An other example is a video that creates a 3-D effect of walking around an action scene as the movie "The Matrix." We have created two videos to compare the proposed system and the "Eye Vision" system. This comparison indicates that rotating the virtual camera by interpolating intermediate viewpoints make the video much more effective than just switching real cameras.

VII. Viewpoint on Demand System

As an application of the proposed method, we have developed a system termed the "Viewpoint on Demand System," which allows viewers to watch soccer match replays from their favorite

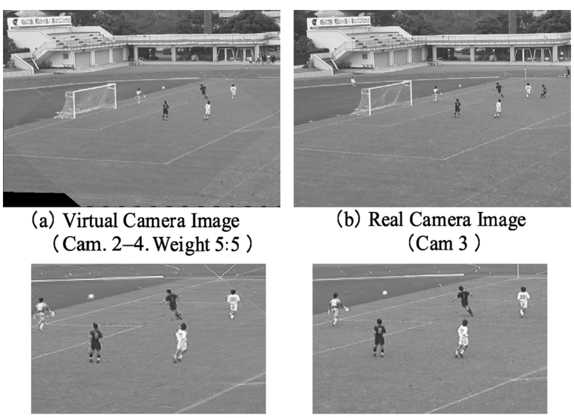
[1]The fly-through view videos are available at the following web site. http://www.ozawa.ics.keio.ac.jp/~nahotty/research.html



(a) Image of View 1 (b) Image of View 2

(c) Synthesized by proposed method (d) Drawn by OpenGL (e) Difference between (c) and (d)

Fig. 13. Evaluation of the proposed method. (c) Image synthesized by the proposed method from (a) and (b) with an interpolating weight value of 0.5. (d) Image drawn by OpenGL from the same viewpoint as (c). (e) Color difference between (c) and (d).



(a) Virtual Camera Image (Cam. 2–4. Weight 5:5) (b) Real Camera Image (Cam 3)

(c) Close-up View of (a) (d) Close-up View of (b)

Fig. 14. Comparison between virtual view image and real view image for a soccer scene captured at the Edogawa Athletic Stadium. (a), (c) Virtual camera images. (b), (d) Real camera images.

viewpoints. Fig. 18 presents the interface of the system. The generated virtual view images are displayed in the center of the window, according to the position and the zoom ratio of the virtual camera. The position of the virtual camera, which is given by the interpolating weight α in (8), is determined by the horizontal slide bar at the bottom of the window. The zoom ratio, which is given by $\acute{f}/f_1$ and $\acute{f}/f_2$ in (8), is determined by the vertical slide bar on the right of the window. Once users select their favorite scenes, rendering of the virtual view, whose position and zoom ratio have been initially defined, begins. While watching the video, the users can change the viewpoint at any time using the two slide bars.

We successfully accomplished fly-through observations for several sequences spanning a few minutes from the beginning of an attack to the end. Fig. 19 presents examples of the images shown on the window of the system. We virtually moved the camera from right to left while zooming in. For example, Fig. 19(a) shows the scene of frame number 322 where the virtual viewpoint is placed at the interpolating weight 4 to 6 between cameras 3 and 4, and the zoom ratio of the virtual camera to the real camera is 0.8.

To evaluate the performance of the system, the processing time was measured by using the desktop PC (CPU: Pentium 4 3.2 GHz, Memory: 2 GB, Graphic Card: ATI Radeon 9800). The system ran at 3.7 fps on an average. The processing time has turned out to be linear in the number of dynamic objects in the output image. This is because the process for virtual view synthesis is performed sequentially for every player and ball. Parallel processing using a PC cluster can increase the processing speed so that the computational cost does not depend on the complexity of the scene.

The proposed system provides an intuitive interface for soccer match observation. Even first-time users can easily control his/her viewpoint via two slide bars because the viewpoint control is very similar to the way of scrolling a display screen on a PC. This application offers a new framework for presenting a sporting match on demand. Along with the broadcast digitizing and convergence of communication and broadcasting, video on-demand systems are getting more attractive applications. A practical system for interactive visualization in sporting broadcast can be constructed based on the proposed viewpoint on demand system.

VIII. Discussions

As the proposed method contains some manual work, we clear up them in this section. One of them is to give corresponding points for estimating projective geometry between cameras, that is fundamental matrix and homographic matrices. It can be easily implemented by just clicking feature points on GUI. The other is to specify the background region and the field region on the captured image in each camera. This process can be easily performed by generating mask images. The manual work mentioned above is required only once because the cameras are fixed. After that, the other offline processes are implemented automatically. In the experiment in the Oita Stadium, we successfully demonstrated the viewpoint on demand system on the next day of image acquisition. The manual work took about an hour in the case of four cameras. Considering producing special effect videos by utilizing the proposed method, it is appropriate for both live broadcast and postproduction. Replays of the exciting scenes of the first half of a match can be provided to audiences in the halftime or in rerun of the match on the next day. If the computational performance is improved, presentation of special effects becomes possible just after the play.

Subsequently, we take up camera configuration. We assume that all the cameras capture the same target area and that variations in lighting and scale across cameras are negligible small. In addition, the neighboring cameras require an overlapping region of the background. In the experiment presented in this paper, we manually adjusted the brightness and the focus of the multiple video cameras so that the size of players and the overall colors in the captured scene can be almost identical across the cameras. At the camera configuration as shown in Fig. 10(a), four cameras were set at a distant of about 10 m. This setup appears to be adequate for covering the penalty area. If more cameras are used, the quality of the synthesized image may be improved. The proposed method has no strong limitations in the color of the players' uniform. The only restricting condition is that the

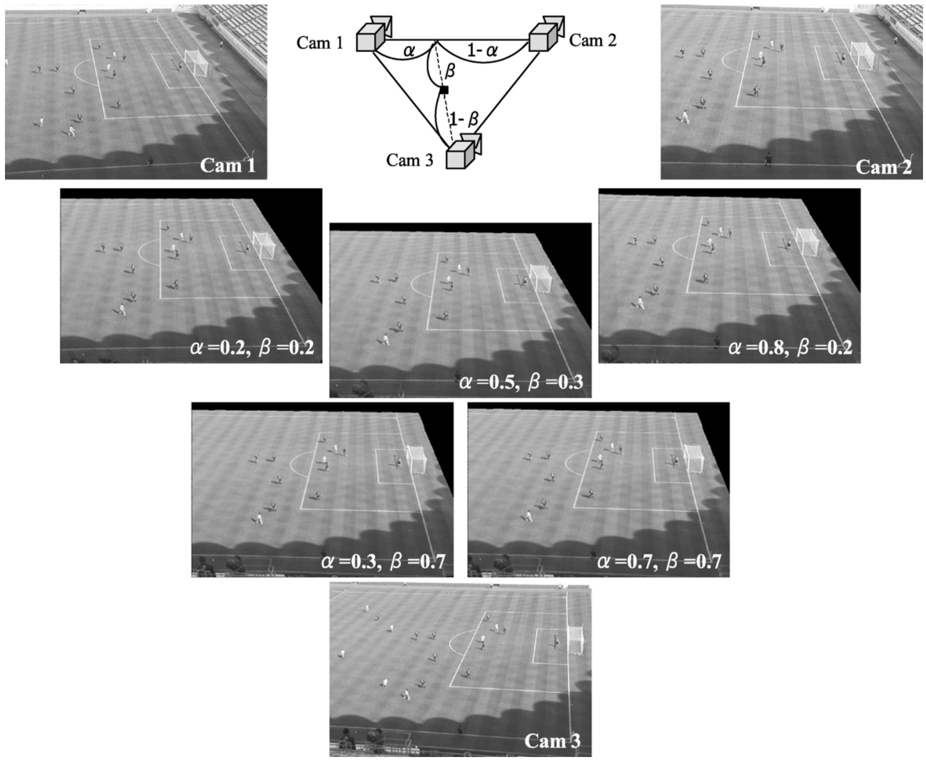


Fig. 15. View interpolation among three views for a soccer scene captured at the Kashima Stadium.

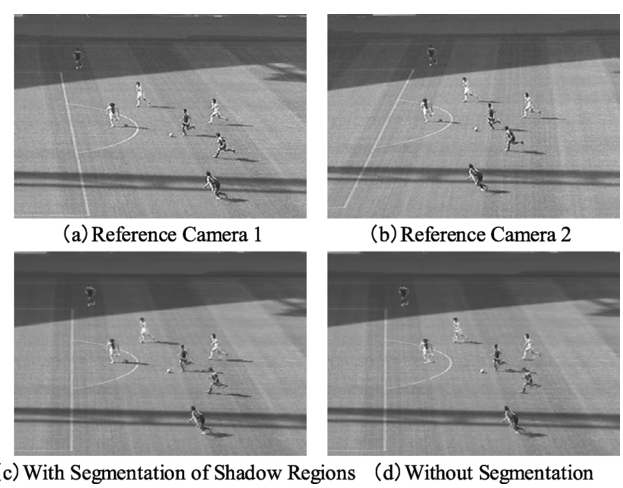


Fig. 16. Comparison of the synthesized image for a scene including shadows at the Oita Stadium. (a), (b) Reference camera images. (c), (d) Virtual view images. (c) is synthesized after segmentation of player/ball region and shadow regions. (d) is synthesized without segmentation. Presentation of shadow reginos in (c) is better than that in (d).

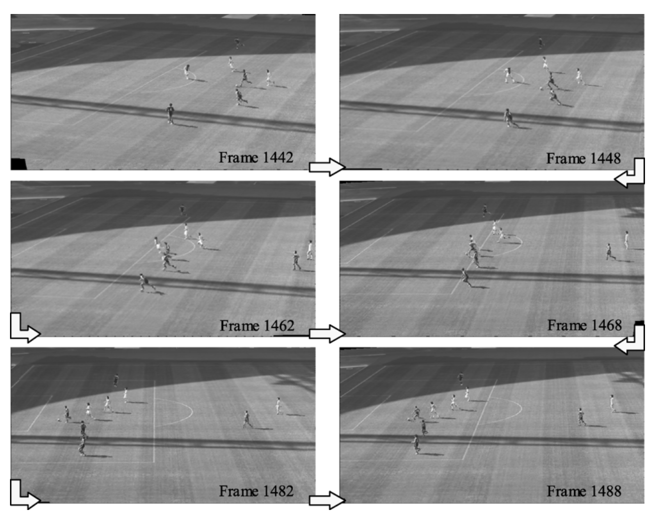


Fig. 17. Example of free-viewpoint video for the sequence including shadows at the Oita Stadium.

colors of the uniform and the ball should differ from that of the ground. As soccer matches usually satisfy this condition, this method can be applied to other soccer matches at other stadiums.

Next, critical situation of the proposed method is considered. Certain errors have been observed in the current approach. A player suddenly disappears from the image sequence because the player has not been captured in two reference camera images. This error occurs when the reference cameras are switched from two cameras, both of which capture the player, to two cameras, one of which does not capture the player. Furthermore, the segmentation/correspondence fails when more than four or five players overlap; hence, a set play may be difficult situation for view synthesis. It is essential to improve the proposed method for such cases.

IX. Conclusion

This paper presents a novel method of arbitrary view synthesis for virtual viewpoint observation of soccer matches. In this method, virtual view images are generated based on view interpolation of two or three real camera images near the virtual viewpoint chosen by a user. Soccer scenes are classified

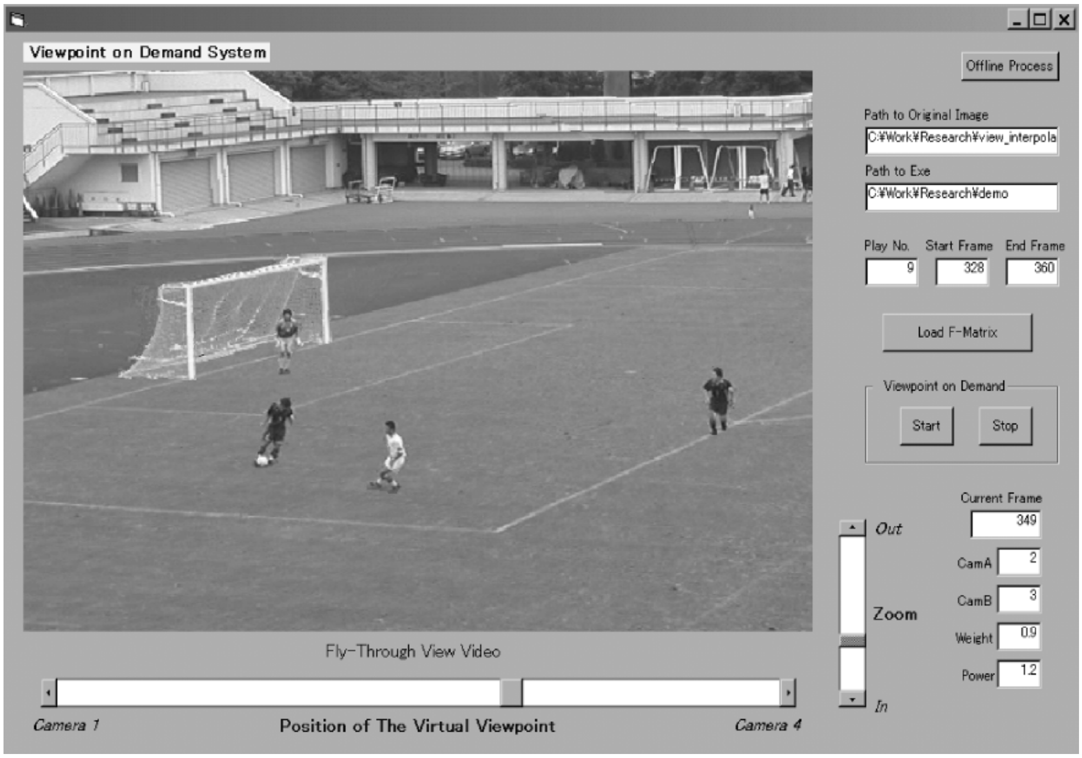


Fig. 18. Interface of the "Viewpoint on Demand System." The horizontal slide bar at the bottom of the window determines the position of the virtual viewpoint. The vertical slide bar on the right of the window determines the zoom ratio of the virtual camera to the real camera.

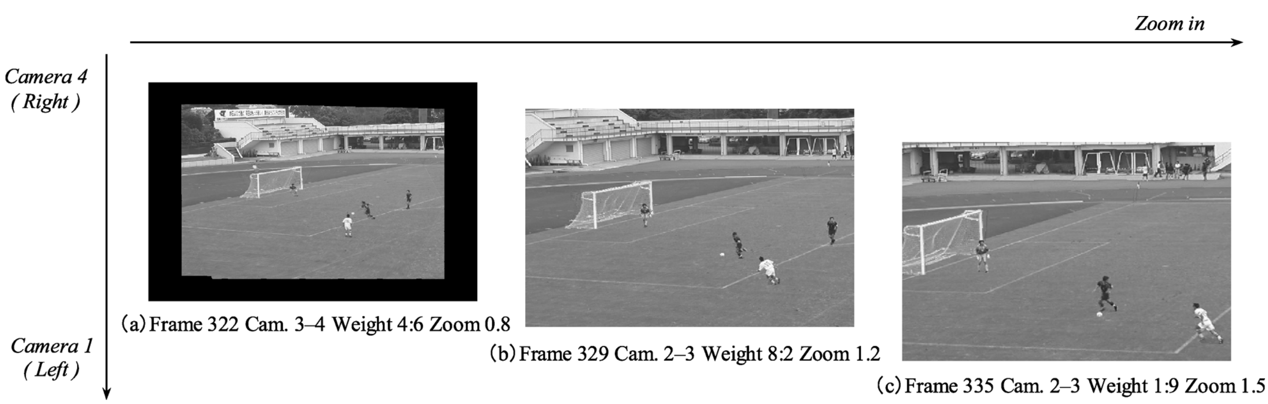


Fig. 19. Examples of the image window of the "Viewpoint on Demand System.".

into three or four regions according to the property of the scene. Appropriate projective geometry is employed for view interpolation of each region. The shadows as well as the players/ball are represented in the virtual view image. By separating the offline process from the online process, we can render an entire soccer scene effectively. The projective geometry between cameras enables us to reduce the difficulties of camera calibration for such large-scale events without the reconstruction of 3-D models. This accomplishes free-viewpoint video synthesis that targets entire dynamic events in a large space such as a soccer stadium.

In addition to the techniques of view synthesis, the application for virtual viewpoint replays of soccer matches has been introduced. The "Viewpoint on Demand System," enables audience to view a soccer match from their favorite angle with the preferred zoom ratio, and allowing them to change these settings at any point of the match. This framework will lead to the creation of completely new and enjoyable ways to present or view entertainment and sporting events including soccer games.

References

[1] T. Kanade, P. W. Rander, and P. J. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE Multimedia*, vol. 4, no. 1, pp. 34–37, Jan. 1997.

[2] I. Kitahara and Y. Ohta, "Scalable 3D representation for 3D video display in a large-scale space," in *Proc. IEEE Virtual Reality 2003*, Mar. 2003, pp. 45–52.

[3] S. Yaguchi and H. Saito, "Arbitrary viewpoint video synthesis from multiple uncalibrated cameras," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 430–439, Feb. 2004.

[4] T. Koyama, I. Kitahara, and Y. Ohta, "Live mixed-reality 3D video in soccer stadium," in *Proc. Int. Symp. Mixed and Augmented Reality (ISMAR2003)*, Oct. 2003, pp. 178–187.

[5] N. Inamoto and H. Saito, "Fly through view video generation of soccer scene," in *Int. Workshop on Entertainment Computing (IWEC2002)*, May 2002, pp. 94–101.

[6] N. Inamoto and H. Saito, "Intermediate view generation of soccer scene from multiple videos," in *Proc. Int. Conf. Pattern Recognition (ICPR2002)*, Aug. 2002, vol. 2, pp. 713–716.

[7] N. Inamoto and H. Saito, "Fly-through viewpoint video system for multi-view soccer movie using viewpoint interpolation," in *Proc. SPIE*, Jul. 2003, vol. 5150, Visual Communications and Image Processing 2003, pp. 1143–1151.

[8] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. SIGGRAPH '93*, 1993, pp. 279–288.

[9] S. Pollard, M. Pilu, S. Hayes, and A. Lorusso, "View synthesis by trinocular edge matching and transfer," *Image Vis. Comput.*, vol. 18, pp. 749–757, 2000.

[10] H. Saito, S. Baba, and T. Kanade, "Appearance-based virtual view generation from multicamera videos captured in the 3-D room," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 303–316, Sep. 2003.

[11] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *Proc. Computer Vision and Pattern Recognition (CVPR1997)*, 1997, pp. 1067–1073.

[12] M. D. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple range images," in *Proc. DARPA Image Understanding Workshop*, May 1997, pp. 911–920.

[13] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-Shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.

[14] S. M. Seitz and C. R. Dyer, "View morphing," in *Proc. SIGGRAPH '96*, 1996, pp. 21–30.

[15] T. Beier and S. Neely, "Feature-based image metamorphosis," in *Proc. SIGGRAPH '92*, 1992, pp. 35–42.

[16] S. Avidan and A. Shashua, "Novel view synthesis by cascading trilinear tensors," *IEEE Trans. Visualiz. Comput. Graph.*, vol. 4, no. 4, pp. 293–306, Oct.-Dec. 1998.

[17] R. A. Manning and C. R. Dyera, "Interpolating view and scene motion by dynamic view morphing," in *Proc. Computer Vision and Pattern Recognition (CVPR 1999)*, 1999, pp. 1388–1394.

[18] Y. Wexler and A. Shashua, "On the synthesis of dynamic scenes from reference views," in *Proc. Computer Vision and Pattern Recognition (CVPR 2000)*, 2000, pp. 1576–1581.

[19] J. Xiao, C. Rao, and M. Sha, "View interpolation for dynamic scenes," in *Proc. Eurographics 2002*, 2002.

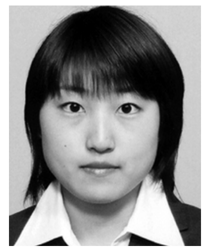
[20] M. Levoy and P. Hanrahan, "Right field rendering," in *Proc. SIGGRAPH '96*, Aug. 1996, pp. 31–42.

[21] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH '96*, Aug. 1996, pp. 43–54.

[22] H. Y. Shum, K. T. Ng, and S. C. Chan, "A virtual reality system using the concentric mosaic: Construction, rendering, and data compression," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 85–95, Jan. 2005.

[23] K. Connor and I. Reid, "Multiple view layered representation for dynamic novel view synthesis," in *Proc. British Machine Vision Conf. (BMVC 2003)*, 2003.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.



Naho Inamoto received the B.E., M.E., and Ph.D. degrees in information and computer science from Keio University, Japan, in 2002, 2003, and 2006, respectively.

She was a Visiting Researcher at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, from 2004 to 2005. She was also a Visiting Researcher of the Institute for Creative Technologies, University of Southern California, Los Angeles, from 2006 to 2007. She served as a Research Assistant for the 21st Century Center of Excellence for Optical and Electronic Device Technology for Access Network from the Ministry of Education, Culture, Sports, Science, and Technology in Japan from 2003 to 2006. She was a Research Fellow of the Japan Society for the Promotion of Science from 2004 to 2007. She has been engaged in the research areas of computer vision and augmented reality.

Hideo Saito received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1987, 1989, and 1992, respectively.

He has been on the faculty of Department of Electrical Engineering, Keio University, since 1992. From 1997 to 1999, he was a Visiting Researcher at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, where he joined the Virtualized Reality project. From 2000 to 2003, he was also a Researcher with PRESTO, JST. Since 2006, he has been a Professor in the Department of Information and Computer Science, Keio University. He has been engaging in the research areas including computer vision, image processing, augmented reality, and human-computer interaction.

Dr. Saito is a member of IEICE, IPSJ, VRST, ITE, IIEEJ, and SICE.