Free-viewpoint Video Synthesis of Soccer Match using Multiple Camera Sources

Chaowei Fang, Takatsugu Hirayama, Kenji Mase
Graduate School of Information Science, Nagoya University
Nagoya, Japan

***Abstract*—**We developed a new three-dimensional (3D) free-viewpoint viewer system with a streaming video-texture billboard method. To synthesize object images continuously and efficiently use a the billboard texture, our method clips objects such as soccer players from multi-viewpoint videos and saves such objects to a video database in a format called Video Texture Catalogue (VTC). The system streams the appropriate VTC over the network and applies texture mapping to billboard at the player's 3D position in each frame of the VTC frame in in real time. Because only moveable objects are streamed, the system can achieve extremely low bit-rate performance online. Our results indicate that object animation can be played smoothly in our system using the Unity™ framework.

Keywords: *free-viewpoint, virtual reality*

I. INTRODUCTION

Videos which multiple cameras capture the same object from different perspectives are called multi-viewpoint videos. Multi-viewpoint videos have been used widely in computer science to analyze or reconstruct three-dimensional (3D) scene such as outdoor sports events[1].

Our research is to realize virtual view synthesis at good speeds that will be suitable for online broadcasting, and allow users to explore the entire scene freely. In this paper, we propose a novel 3D viewer system for virtual view generation using a billboard and multi-viewpoint videos. The main difference between our system and traditional view reconstruction systems is that our system does not construct 3D models from video or image contents; instead, our system simply uses a streaming video-texture billboard to reduce the size of online streaming data.

The proposed system provides several methods to help users explore the final synthesized 3D scene. In order to obtain high performance from real-time synthesis we introduce a new method called Video Texture Catalogue (VTC) method. We apply to compose parts of the synthesis data offline and to reconstruct the scene in real time. Background objects, such as a soccer stadium, are constructed as 3D objects in advance. The movement of objects such as players and the ball is restored by playing back their position at the specific time recorded by the laser sensors during the event. The system allows users to view the 3D environment freely in order to get experience a free-viewpoint viewer system.

The remainder of this paper is organized as follows. Related work on billboard synthesis methods is reviewed in Section II. Three methods are introduced in the related work. In Section III, the proposed method is described, including a record of multi-viewpoint videos, the method for our synthesis, and a method for building 3D environments. Finally, the conclusion and summary are given in Section IV.

II. RELATED WORK

Research on virtual view synthesis has been conducted since the 1990s [2-3]. Shai Avidan et al. performed view synthesis via billboard method. In that research, photos of an object taken from 3 angle were used. They applied texture mapping to billboard in order to synthesize one object.

Some researches on virtual view synthesis from multi-viewpoint videos has been proposed [4] [5]. Methods for improving the video compression efficiency of multi-viewpoint video were proposed. J. Starck et al. proposed a method that can achieve a good performance on multi-viewpoint video compression via H.264/AVC [6].

The mentioned researches proposed mainly for indoor environment. It is more difficult for some outdoor events view synthesis because the objects might not have enough resolution for the huge scene and the background might be more complex. Some view synthesis method and view system for outdoor events were proposed [7] [8].

Jarusirisawad et al. [7] calibrated the video contents by applying a homography and used billboard to synthesize players with multiple cameras. They synthesized the players, the stadium and other objects for "a fly-through" scene of a soccer match held in a particular stadium using 4 multi-viewpoint cameras.

In the research of 3D viewer system, Horiuchi et al. proposed a mobile interactive viewer based on free-viewpoint video technology [9]. Users can watch a music video from various viewpoint controlled by touch screen.

III. OVERVIEW OF THE PROPOSED METHOD

A multi-camera video source for soccer matches is utilized to synthesize movable objects in a 3D virtual scene in order to realize real-time, free-viewpoint 3D viewing of the soccer match. Horiuchi et al. synthesized such scene from video frames, but when the system is online and provided as a viewer to the end user, there is still a lot of problem need to be solved such as the performance of network streaming and the needs of a proper user interface. Our proposed system synthesizes the scene continuously in real time, placing each object on its exact

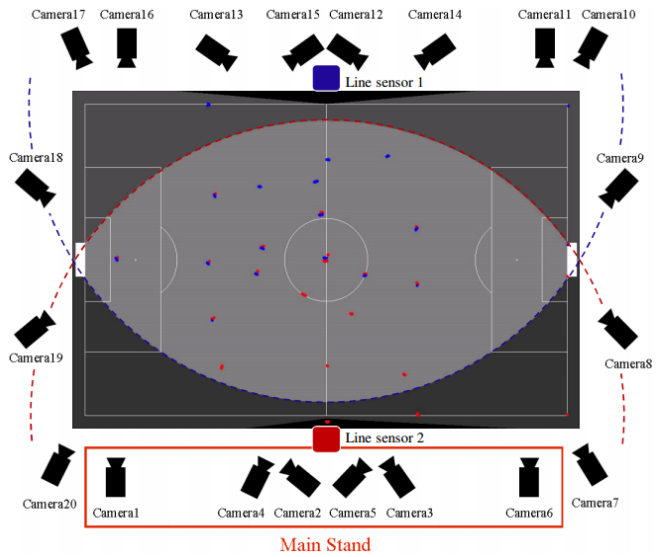

Figure 1. Video capture settings

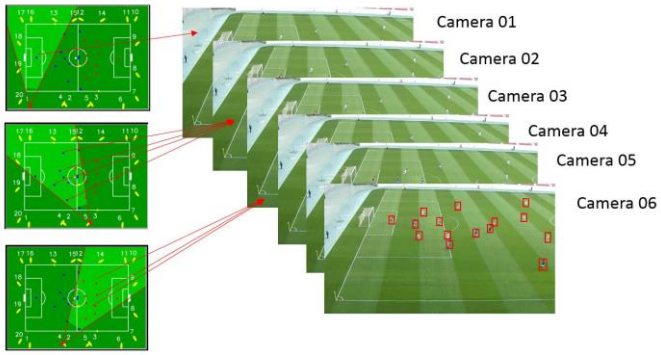

Figure 2. Clipped player rectangle from multi-viewpoint videos.

position in the 3D soccer field. In this section, we describe the basic idea of VTC and the system interface.

A. Multi-viewpoint video preparation

We captured the scene of a soccer match in a standard soccer stadium. Twenty cameras (CANON EX-F1) were set close to the audience seats throughout the stadium. The location settings of the cameras are shown in Figure 1. We recorded the match simultaneously with all the cameras. With this set of cameras, we obtained a sequence of videos for one match from multiple perspectives. All the videos have a resolution of 1920 × 1080 pixels recorded at a speed of 30 frames per second (fps). The format is a 24-bit RGB color image. In addition, we used two line sensors that were placed on the center of each side line in the soccer field. With the sensors, we obtained the position of all players and referees on the ground in 25 Hz.

B. VTC generation

In order to synthesize objects in a 3D scene, we compose a video file that contains a series of images of the target objects. This video file is comprised of multi-viewpoint videos. We call such video file VTC. In this section, we describe the method for generating a VTC generation.

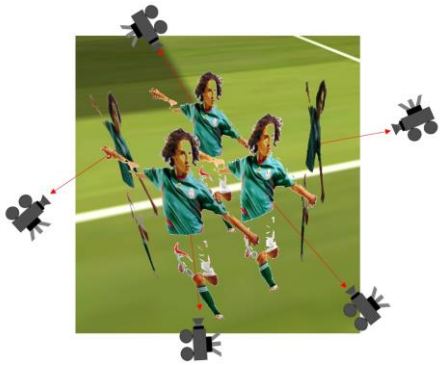

Figure 3. Image of clipped rectangle from videos.

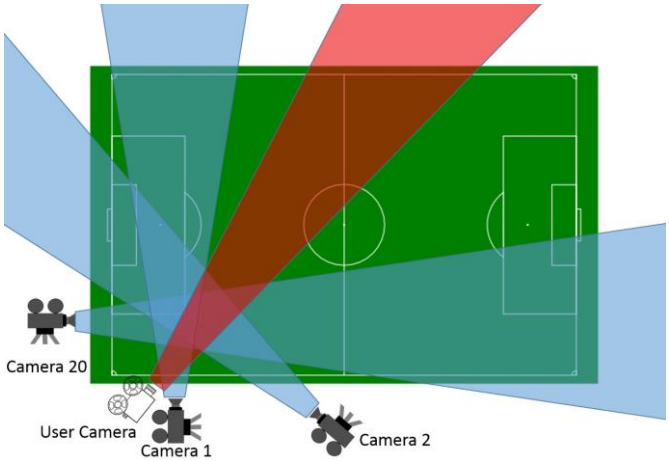

Figure 4. Camera selection strategy in generating VTC. Red space: the perspective of user camera; Blue space: the perspective of video camera

Figure 5. VTC results.

We use the positions of moveable objects from line sensors and a homography matrix of the soccer field plane viewing from each camera to calculate 2D clipping rectangles including them in the image captured from each viewpoint. The rectangle information shows where an object is located in a multi-viewpoint video. It is necessary to note that it is difficult to track the position of the ball and to calculate its rectangle because it situates at a significant distance from our cameras and the size is smaller than the resolution of laser sensor; moreover, the ball usually moves at a high speed. Currently, we obtain

ball position by manually annotating ball-touch and reducing the touching players position to key frame ball position. We process each rectangle for each of players and referees in a camera, as shown in Figure 2, then we compose a VTC.

Because we use multiple cameras, the scene is recorded simultaneously in multiple videos. The objects in the scene might be recorded from various angles and the rectangle of one objects might be from many cameras. If we apply texture mapping to the billboards, the image will be as shown as in Figure 3. We clip the moving players and referees from each video frame of multi-camera shooting using their rectangle information.

Using the method described in the previous paragraphs, we obtained an image sequence of the each object using individual frames from the multi-viewpoint videos. The sequence is used to present object animation. For one object, one video image is selected at each frame to present his movement. We encountered certain difficulties when selecting these player videos. We describe the problems and solutions in the following of this section. Consequently, we introduced an integration function over different cameras to fill the catalogue with proper images of the objects in every frame.

First, the required object texture might not always be available from one camera due to camera's field of view; therefore, we require a strategy to ensure that a player's image always appears in a VTC scene provided that the player is recorded by at least one of the multi-viewpoint cameras during any circumstances. We search the current processing player from all multi-viewpoint cameras and find an appropriate view for insertion to a VTC.

Second, because a given object is recorded from various angles, we want to observe it from a proper view in the 3D space. For example, when the current virtual user camera is located at a front corner of the soccer field as shown in Figure 4, we prefer to sample the images of players from camera No. 20, No. 1 and No. 2 whose positions are near the virtual camera.

At minimum, the player located far away from the camera appears to be small in the final video, which can cause a poor visual effect in the final synthesis. In order to solve this problem, we introduce the following two strategies. We consider such an effect when choosing a proper video scene and do not take it into account if the rectangle is too small. Furthermore, we prefer to sample the object images from cameras whose positions are close to the player. Then, we give some weight to both strategies. According to the weight adjustment, we achieve reasonable visual effect for image synthesis.

Using the three steps described in previous, we obtained a proper image of one object from a reasonable multi-viewpoint video for one frame. We put all these image of all movable objects to one image as a video frame of VTC. At the moment, the resolution of the player image is 72 × 128 and 360 × 640 for the entire VTC video frame.

Subsequently, we perform the background subtraction [6] for the video frame.

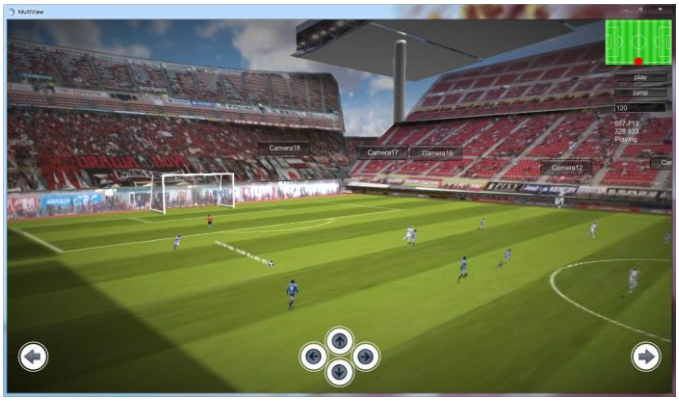

Figure 6. Results from final synthesis on the proposed interface.

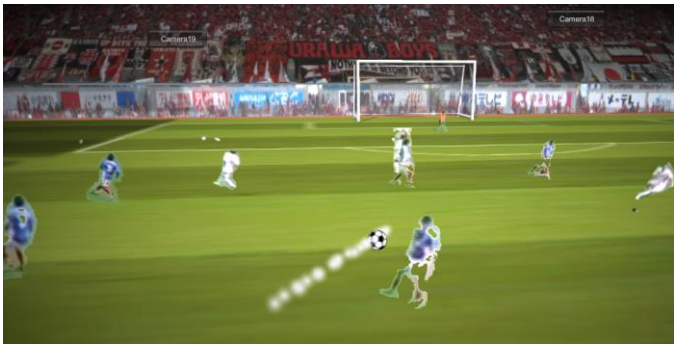

Figure 7. View from the referee, camera is now focusing on the ball.

Because our proposed system works in an online environment, and because one of our objectives is to reduce the bandwidth of network streaming, these videos are not used immediately. Instead, we group the images of multiple objects with the same frame time into one video frame in order to best utilize modern data compression algorithms such as JPEG [10] for one single frame and H264/AVC for the video sequence to minimize the transition bandwidth. The series of image frames is encoded to a video file using the H264 protocol. We call the final video file VTC; Figure 5 shows the results of such a video. VTC is used to synthesize the moveable objects in a 3D environment.

VTC generation runs in the following environment: a standard PC with two Intel® Xeon quad-core 3.0GHz processors, 48 GB memory, and a NAS with about 4 MB/s of read/write speed. To process 20 video files with 25 moveable objects required approximately ten hours.

C. VTC streaming and billboard texture mapping

We make the VTC video offline and use it to synthesize objects with billboard. In this section, we discuss the texture mapping on billboard. We apply a traditional sprite to the billboard facing the current virtual user camera. Such billboards are used to present moveable objects. All our 3D scenes are made by Unity™ Pro and the billboard is a quad primitive set to face the current user camera during each frame. We stream the VTC in real time and process every video frame. The built-in video stream module in Unity™ is not as powerful as we believe. For example, it cannot seek the specific time of a vid-

eo, and it only supports limited video file formats such as the OGG video format [11][12]. To solve these problems, we developed a video plug-in for Unity™ using the FFmpeg library [13] to stream the VTC video file and convert the frame image to the Unity™ texture format. Then, we applied texture mapping to the billboard at each frame to obtain smooth animation. Finally, we shade the billboard frame by frame using alpha blending to compose the scene.

D. 3D Interface

Human interaction and 3D environment presentation are a vital part of our system. The realization of our main 3D interface is described in this section.

The VTC video and the billboard are used to synthetize moving objects to obtain a reasonable performance and to save bandwidth, whereas static background objects such as the soccer stadium and the field are rendered using 3D models. The general real-time 3D game engine called Unity™ is used to build the basic 3D environment. We built the 3D models, including the building and soccer ground of the Toyota Stadium in Toyota, Aichi Prefecture, Japan, where we recorded an entire soccer match using multi-viewpoint cameras described as Section III-A. In addition, we also captured photos and used them as the texture of the 3D models. Because we used Unity™, some graphic rendering techniques such as HDR, volume light and soft shadow are available to improve the realism of the 3D environment.

Because an entire virtual soccer match is to be simulated, the movement of objects is also presented by the system. Given that we recorded the position of players and referees using the two laser sensors on the soccer field as described in Section III-A. Then, the real coordinate is converted to the 3D virtual soccer field. All the position data are synchronized to the time in the video; We place the billboards at these player positions. The ball is so small that is difficult to obtain from the multi-viewpoint videos; consequently, we simply use a fixed ball texture in the billboard. We add particle effects such as the tracks of the ball shown in Figure 7 to enhance the visual effects.

The system provides users with certain control methods for exploring the 3D space in order to help them find their favorite point of view and to provide a unique exploring experience. Figure 6 shows the main screen of our 3D interface.

Buttons are placed at the original location of the multi-viewpoint cameras. The proposed system can navigate rapidly to any real camera position. Next and previous camera selection buttons are placed at the right bottom and left bottom corners of the interface.

Pitch and yaw control of the cameras are provided at the center bottom portion of the interface. A traditional first-person view camera control that uses the mouse drag function to control the camera pitch and yaw is provided; such control type is also compatible with touch-screen monitors and mobile devices such as tablet-PCs. However such first-person camera control is tedious especially in ball tracking.

One of our exploring methods allows users a unique experience. Virtual camera can be fixed to a specific object, such as a keeper or a shooter on the field and always look at the ball, as is shown in Figure 7. This way, the users can feel as though they have joined the match.

A minimum set of preset cut-scenes is provided to allow users to enjoy moments of good player shots, such as goals scored.

To evaluate our system, we measured the system's performance using a laptop PC with the following specifications: Core i7 2.6 GHz CPU, 8 GB Memory, Intel® HD Graphics 4000. The system can run at approximately 60 fps with a standard 720p HD resolution of 1280 × 720 with normal graphic quality settings for Unity™, and approximately 30fps with a 1080p HD resolution of 1920 × 1080.

E. Object synchronization

The movement of objects and the streamed video run in their own timeline; therefore, the timelines should be synchronized. Because we use FFmpeg to stream the VTC video, we obtain enough space in the video buffer to perform slight adjustments when the playback time is different to the movement simulation time.

We built an intermediate time server to ensure synchronization between movement and VTC. The time server sends a timestamp to the client (e.g., our system) continuously. Our system calculates the interpolated position of the objects and adjusts the VTC time according to the timestamp received.

IV. CONCLUSION

We have proposed a novel method for synthesizing moveable objects in a soccer match using multi-viewpoint videos as input, and a new viewer system that uses streaming video-textured billboards to perform the synthesis in real time. We believe that our work is a help in the research of online view synthesis and view of multi-viewpoint videos.

The resulting system runs on a standard PC with normal integrated graphic cards at a real-time interactive speed. The proposed system was designed with free-viewpoint exploration. The system can achieve a good experience when users wants to watch the event from a specific perspective that might not be presented by traditional videos. Because background objects in the 3D environment are presented through 3D models and only moveable objects are synthetized by VTC, we believe that our system can also be used for other sport events or for other types of scene synthesis using multi-viewpoint videos.

There is one major limitation to our system. The amount of time required for offline processing (mentioned in Section III Part B) is significant. In most circumstances, manual intervention such as ball position calculation is required when processing VTC offline. If the system is used for televised or online broadcasting, the processing of video contents will be an additional cost to contents providers.

The system requires several optimization. In certain circumstances, VTC is not as flexible as we believe. The player rectangle result might be with a good resolution but appear in the opposite of current user camera. The player rectangle selection result between good visual effects such as an appropri-

ate resolution or real synthesis results such as the correct orientation of rectangle continues to be a problem. A higher accuracy of rectangle selection strategy is required when generating VTC. For future work, we will consider dynamic VTC. We can generate VTC via a cloud server in real time because the strategy for generating VTC is based on real-time client calculation based on the position and direction relationship between cameras and moveable objects.

V. ACKNOWLEDGEMENT

This work has been supported in part by National Institute of Information and Communication Technology for the research of Ultra-Realistic Communication Forum in Japan.

REFERENCES

[1] Grau, O., Thomas, G.A., Hilton, A. Kilner, J., Starck, J., "A Robust Free-Viewpoint Video System for Sport Scenes" 3DTV Conference, Kos Island, pp. 1-4, 2007

[2] Avidan, S., Sashua, A., "Novel view synthesis in tensor space," In: Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, pp. 1034-1040, 1997

[3] Kanade,T. , Rander, P. W., Narayanan, P.J. "Virtualized reality: concepts and early results". In IEEE Workshop on Representation of Visual Scenes, pp. 69–76, 1995.

[4] J. Starck, A. Hilton, "Virtual view synthesis of people from multiple view video sequences", Graphical Models, vol. 67, issue. 6, pp. 600-620, 2006

[5] Martinian, E. and Behrens, A. and Xin, J. and Vetro, A., "View Synthesis for Multiview Video Compression" Picture Coding Symposium, Beijing, 035, 2006

[6] H.264/AVC Reference Software Encoder Documentation JM 18.6.
(http://iphome.hhi.de/suehring/tml/doc/lenc/html/)

[7] Songkran Jarusirisawad, Kunihiko Hayashi, Hideo Saito, Naho Inamoto, Tetsuya Kawamoto, Naoki Kubokawa, Toru Fujiwar, "The intermediate view synthesis system for soccer broadcasts". In: Proceedings of ASIAGRAPH 2008, Shanghai, 2008

[8] Oliver Grau and Adrian Hilton and Joe Kilner and Gregor Miller and Tim Sargeant and Jonathan Starck, "A Free-Viewpoint Video System for Visualization of Sport Scenes" Motion Imaging, New York, pp. 213-219, 2007

[9] Horiuchi, T., Sankoh, H., Kato, T., and Naito, S. Interactive music video application for smartphones based on free-viewpoint video and audio rendering. In Proceedings of the 20th ACM international conference on Multimedia, pp. 1293–1294, 2012.

[10] JPEG Homepage
http://www.jpeg.org/jpeg/index.html

[11] Vorbis.com, the xiph open source community
(http://www.vorbis.com/faq/#what)

[12] Unity Script Reference
(https://docs.unity3d.com/Documentation/ScriptReference/WWW.html)

[13] FFmpeg, About FFmpeg
(http://www.ffmpeg.org/about.html)