Free-viewpoint video synthesis for soccer games

Davide Lusuardi
Department of information engineering and computer science
University of Trento
Trento, Italy
davide.lusuardi@studenti.unitn.it

***Abstract*—In this document, we discuss some methods to accomplish free-viewpoint video visualization for soccer scenes. These methods generate novel views of actions from any angle and allow viewers to virtually fly through real soccer scenes.**

I. Introduction

Nowadays, sport broadcasting plays a large role in current society. Therefore, it is important to provide a high quality and visually pleasing reporting of sports events. As we know, incidents in sport tend to be over very quickly. Slow-motion replays can be used to illustrate these incidents as clearly as possible for the viewer.

Although time is stretched in these replays, there is no exploration of the spatial scene information, which is usually important for understanding the event. A system that allows a replay from any angle adds a lot of value to the viewer experience.

Free-viewpoint video (FVV) is one of the new trends in the development of advanced visual media type that provides an immersive user experience and interactivity when viewing visual media. Compared with traditional fixed-viewpoint video, it allows users to select a viewpoint interactively and is capable of rendering a new view from a novel viewpoint [1]. FVV has been a research topic in the field of computer vision since the virtualized reality system [2] was developed, ranging from static models for studio applications with a fixed capture volume, controlled illumination and backgrounds [3] to dynamic object models for sports scenes [4]–[6]. Live outdoor sports such as soccer involve a number of additional challenges for both acquisition and processing phases. The action take place over an entire pitch and video acquisition should be done at sufficient resolution in order to do analysis and production of desired virtual camera views.

In this paper we briefly present and compare some methods to accomplish free-viewpoint video visualization for soccer scenes...

II. Overview

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 [7]–[9]. Free-viewpoint video in sports TV broadcast production is a challenging problem involving the conflicting requirements of broadcast picture quality with video-rate generation. FVV techniques for generating novel viewpoints using a multiview camera setup can be catogorized into two classes: 3D reconstruction and image-based rendering. Using 3D reconstruction, it is possible to construct 3D models of objects to generate the desired view from an arbitrary viewpoint. The quality of the virtual view image generated by these methods depends on the accuracy of the 3D model. In order to produce an accurate model, a large number of video cameras surrounding the object are used. Moreover, ... TODO:01

It is not practical to apply these methods to a scene that contains multiple objects with complicated movements such as a sporting match. TODO

III. The iview system

In this section we present the DTI-funded collaborative project *iview* [10], a free-viewpoint video system which enables the production of novel desirable camera views such as the goal keeper view, the referee view or even the ball camera. This system exploits the already placed live TV broadcast cameras as the primary source of multiple view video. Usually football matches are covered by 12-20 high-definition cameras placed all over in the stadium providing wide-baseline views. Match cameras are manually controlled to follow the game play zooming in on events when occurs. However, only a fraction of these are focused on specific events of interest and can be used for production of free-viewpoint renders, the remaining cameras cover the pitch, crowd and coaches.

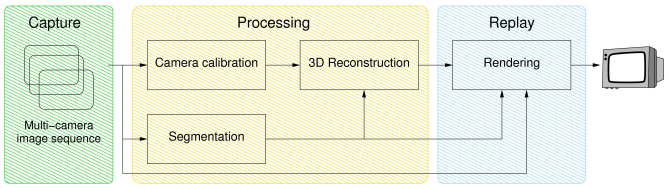

Figure 1. Overview of the free-viewpoint system. [TODO: cite 02.1]

The *iview* system is composed of three main modules as shown in Figure 1: capture, processing, and replay module.

Capture is performed using time synchronised acquisition from both auxiliary and match cameras. The minimal number of cameras is about four, but for good quality results a higher number is required. Camera synchronisation is achieved using standard genlock process.

The processing module computes a 3D model of the scene. This is done using segmentation of objects from the background and 3D reconstruction [11].

In order to allow the use of footage from match cameras and to avoid the need for prior calibration, automatic calibration of all cameras is performed using a line-based approach againts the pitch lines of the captured footage, achieving a root-mean-square error of 1-2 pixels for moving cameras. The calibration is very fast and robust, capable of real-time operation for use during live match footage. Calibration estimates the extrinsic and intrinsic parameters of each camera including lens distortion.

The segmentation is needed to separate the foreground, i.e. the players from the background. Matting of players from the green pitch is performed using chroma-keying matting. The authors developed and tested a k-nearest neighbour approach for chroma-keying and evaluated two other known techniques, *Fast green subtraction* in RGB colour space and keying in HSV colour space. The k-nearest neighbour classifier is controlled by a GUI where the user has to click on position in the image that correspond to background. The process is repeated until the resulting segmentation is satisfying. A deeper explanation is present in the paper [11] where the authors present and evaluate also *Fast green subtraction* and keying in HSV.

The accumulation of errors from calibration and matting can cause large errors in the reconstruction of the scene such as loss of limbs. Therefore, robust algorithms have been developed for scene reconstruction. One possible technique is called visual-hull (VH) and represents the maximum volume occupied by an object given a set of silhouettes from multiple views [2.2:8]. The visual-hull is a single global representation integrating silhouette information from all views. A polygonal mesh surface is typically extracted and texture mapped by resampling the captured multiple view video for rendering [2.2]. Due to accumulating errors in camera calibration and segmentation, visual-hull accuracy is reduced. A refinement of the view-dependent visual hull (VDVH) [2.1:12] using stereo correspondence to interpolate between captured views can be used to overcome these issues achieving the best alignment between adjacent views and hence improve visual quality. More information about *iview* 3D reconstruction can be found in [2,2.1,2.2].

Finally, the replay module renders the novel view of the scene using the computed 3D model together with the original camera images. Cameras closer to the virtual viewpoint are chosen to generate the novel viewpoint.

IV. Image-based rendering - extended plane sweeping

In this section we present the work of Goorts et al. [1], i.e. an image-based approach to generate virtual camera view interpolating real camera images. Instead of performing 3D reconstruction, image-based methods generate directly the image of the novel viewpoint. When multiple cameras are present, plane sweeping can be used [05:Yang et al., 2004 TODO] for both small and wide baseline setups. Plane sweeping has already been used for novel view point in soccer scenes. Goorts et al. [05:Goorts et al., 2012a; Goorts et al., 2013a TODO] present a method with two plane sweeps and a depth filtering step suitable for smaller baseline setups of about 1 meter. This method present some problems like disappearing players when they overlap in the image.

The method presented here is fully automatic and employ GPU parallel processing to achieve fast processing speed. The system setup consists of 7 static cameras placed in a wide baseline setup, i.e. 10 meters between each camera. All cameras are synchronized on shutter level using a global clock. The generation of novel viewpoint consists of two steps as shown in Figure 2: a first off-line preprocessing phase and a real-time interpolation phase. As explained in [1], the preprocessing phase is responsible of camera calibration, acquiring position, orientation and intrinsic parameters for each camera, and background determination. The real-time phase generates images for a chosen virtual camera position and a chosen time in the video sequence. More in details, camera images are debayered and segmented using GPUs. Debayering consists of converting the raw images to its RGB representation and segmentation is based on backgrounds obtained during off-line preprocessing. This method [1] allows fast segmentation in high quality. These images are then used to process foreground and background independently. The foreground rendering uses a plane-sweep approach followed by depth filtering and a depth- selective plane sweep as explained in [1]. In this way, the authors obtained high quality results using wide baseline setup and typical artifacts, such as ghost players, are removed.

V. Billboard-based visualization

In this section we present the work of Ohta et al. [12]. The authors use billboard representation to make a 3D model of each player. This method is simpler than full 3D reconstruction and require less computation. A player billboard is a small rectangle standing perpendicular to the ground and a 2D texture is shown on it. The difference between 3D reconstruction and billboard representation is shown in Figure 3: the visual difference is clear at a close viewpoint but becomes very small at a distant viewpoint.

The system proceed as follow: first extracts texture segments from camera videos, then selects appropriate textures according to the virtual viewpoint and finally layouts the player billboards in virtual space. Texture extraction phase consists in obtaining location of each player and extracting texture segments from every image video by projecting player location onto the image plane. Background region is removed in the texture by video capturing PC. Texture selection phase selects a set of texture to be sent to each viewer based on his viewpoint. Given a viewpoint, the system finds the camera that minimizes the angle between the line from the viewpoint to the player location and the line from the camera to the player location.

One possible problem happens when players are overlapped each other at a certain camera and billboard texture could include both players. To eliminate extra player region, authors used stereo based method [**?**] as explained in [12].

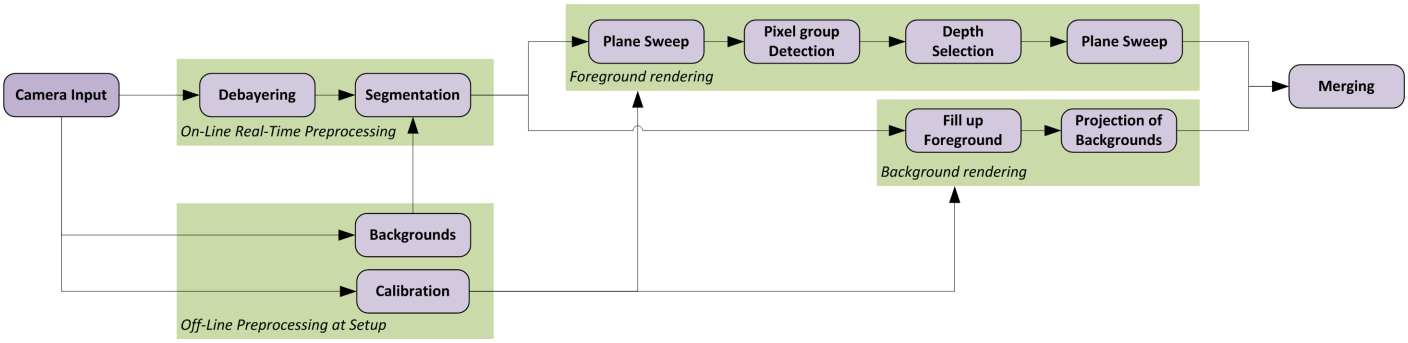

Figure 2. TODO

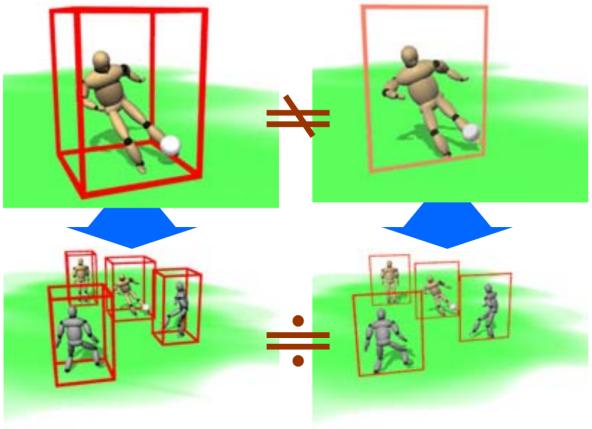

Figure 3. Appearance similarity between 3D reconstruction and billboard in close and distant view [12]

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [7]. The sentence punctuation follows the bracket [8]. Refer simply to the reference number, as in [9]—do not use "Ref. [9]" or "reference [9]" except at the beginning of a sentence: "Reference [9] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [2]. Papers that have been accepted for publication should be cited as "in press" [3]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [4].

REFERENCES

[1] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, and P. Bekaert, "Free viewpoint video for soccer using histogram-based validity maps in plane sweeping," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, 2014, pp. 378–386.

[2] N. Inamoto and H. Saito, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[3] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[4] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[5] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[6] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[7] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[8] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[9] ——, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1155–1166, 2007.

[10] "The iview project," http://www.bbc.co.uk/rd/projects/iview.

[11] O. Grau, G. Thomas, A. Hilton, J. Kilner, and J. Starck, "A robust free-viewpoint video system for sport scenes," in *2007 3DTV Conference*, 06 2007, pp. 1 – 4.

[12] Y. Kameda, T. Koyama, Y. Mukaigawa, F. Yoshikawa, and Y. Ohta, "Free viewpoint browsing of live soccer games," in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 01 2004, pp. 747–750.