BBC

Research & Development

White Paper

WHP 213

November 2011

Multi-view 4D reconstruction of human action for entertainment applications

Oliver Grau

BRITISH BROADCASTING CORPORATION

White Paper WHP 213

Multi-view 4D reconstruction of human action for entertainment applications

Oliver Grau

Abstract

Multi-view 4D reconstruction of human action has a number of applications in entertainment. This chapter describes a selection of application areas that are of interest to the broadcast, movie and gaming industries. In particular free-viewpoint video techniques for special effects and sport post-match analysis are discussed. The appearance of human action is captured as 4D data represented by 3D volumetric or surface data over time. A review of recent approaches identifies two major classes: 4D reconstruction and model-based tracking.

The second part of the chapter describes aspects of a practical implementation of a 4D reconstruction pipeline. Implementations of the popular visual hull are discussed, as a building block in many free-viewpoint video systems.

This document was originally published in 'Visual Analysis of Humans: Looking at People', Moeslund, Th.B.; Hilton, A.; Krüger, V.; Sigal, L. (Eds.), Springer 2011.

The original publication is available at www.springerlink.com.

Additional key words: Computer Vision, Computer Graphics, Broadcasting.

White Papers are distributed freely on request.

Authorisation of the Chief Scientist or General Manager is required for publication.

Multi-view 4D reconstruction of human action for entertainment applications

Oliver Grau

Abstract Multi-view 4D reconstruction of human action has a number of applications in entertainment. This chapter describes a selection of application areas that are of interest to the broadcast, movie and gaming industries. In particular free-viewpoint video techniques for special effects and sport post-match analysis are discussed. The appearance of human action is captured as 4D data represented by 3D volumetric or surface data over time. A review of recent approaches identifies two major classes: 4D reconstruction and model-based tracking. The second part of the chapter describes aspects of a practical implementation of a 4D reconstruction pipeline. Implementations of the popular visual hull are discussed, as a building block in many free-viewpoint video systems.

Oliver Grau
BBC Research & Development, 56 Wood Lane, London, UK, e-mail: Oliver.Grau@bbc.co.uk

1 Introduction

This chapter describes applications and approaches for capture and reconstruction of 4D appearance models of human action from multiple cameras. Then the use of these models in entertainment applications is outlined. Applications like free-viewpoint video aim to capture and reproduce the appearance of the human action as well as possible using a discrete number of multiple video streams.

A realistic synthesis of views from a continuum of new positions requires an underlining transformation model in order to generate these views from the captured video. This transformation can be formulated explicitly using a 3D geometrical description of the shape or by image-to-image correspondences between views. We refer to a representation that captures such human action as *4D model* and use this term equivalent with *3D model of human action*. Another term often used in related literature is *3D video*.

In the simplest case a 4D model is a sequence of 3D models, but can also include temporally aligned 3D data. By analogy to the way that 3D reconstruction aims to generate a 3D model from multiple views we use the term 4D reconstruction to describe the generation of a 4D model of action from multiple videos. A complementary approach is by using a model-based representation that usually fits a 3D shape model to captured data and then tracks motion over time.

There are a number of applications for 3D models of human action in entertainment. In particular the film industry increasingly makes use of 3D data to generate special effects using computer graphics methods [30]. The generated models employ a mix of different approaches, from completely manual to the use of laser scanning of static poses and motion capture techniques for the performance. In any case the effort in particular in the post-production phase is very high. The productions require the highest visual quality and high-end cinema productions dedicate a good proportion of the overall budget to the implementation of spectacular visual effects.

In broadcast production, budgets are on a much smaller scale. Furthermore, many productions are produced in real-time and broadcast immediately. Both requirements prohibit an extensive and therefore expensive post-production phase. Applications of 3D models of human action in broadcast require therefore methods that produce results without too much human intervention and a quick turnaround time or even real-time for live broadcast.

The production of video games is another domain that makes use of captured 3D human action. However, the data is used differently compared to applications in films and broadcast. The main difference lies in the fact that games always generate ‘new’ action on the fly, instead of just replaying captured action. This requires a different data representation. In practice the production process requires a lot of manipulation to achieve good visual results while keeping the amount of data and processing manageable on the end device.

2 Applications

Typical applications for 3D models of human action in broadcast and film are seeking to overcome the limitations of physical cameras. An important application is free-viewpoint video (FVV), i.e. the freedom to chose a camera viewpoint independent from real camera positions. This enables a number of special effects.

The movie 'The Matrix' became famous for the *bullet time*™effect. In this effect the action is stopped or slowed down and the camera viewpoint is then moved around the actors to explore the spatial domain. This effect was created with an array of cameras that synchronously capture a single image. The technique is also known as 'time slicing' and has been pioneered by a number of people including 19-century photographer Eadweard J. Muybridge. Macmillan continued this work in the 1980s [1].

Kanade applied a similar technique to the US Superbowl, with 30 actively panning and tilting cameras known as the Eye Vision system [18]. An operator interactively defines the point of interest on the pitch and the cameras centre synchronously to this point. This gives the impression during replay that the virtual camera is rotating around this pivot point.

Time slicing tries to explore the spatial domain by playing images captured at the same time, but from different cameras combined into a video sequence. Therefore the camera path is determined by the position of the real cameras and cannot be changed during replay. In contrast to that, FVV allows full control over the position of the virtual camera position at replay. In particular, camera positions or movements can be rendered that would be otherwise impossible to achieve with a real camera.

Another application of FVV is pre-visualisation for planning of shots. This is an issues for example when real and virtual scene content is mixed in a scene or if complicated camera movements are intended. With the help of FVV the interaction of the scene components and the camera path can be planned on the computer [10] and can reduce rehearsal and expensive on-set production time.

FVV can also be applied to visualise events or incidents. An important application is to generate new views from sport events, like football or rugby that enhance the coverage of such events. Since many players are involved and occlude each other in these sports it is usually hard to explain tactical aspects of key incidents from just fixed camera positions. FVV enables sport presenters to explore interesting incidents by moving to new viewpoints, like a virtual flight down to pitch level or overhead. This is a powerful tool to visualise spatial relationships between players and their tactics. An example [1] is depicted in figure 1.

3D data of action also has other uses, without directly generating new visual data. One example is the generation of additional data, like statistical information about speed and biometrical data of athletes. Another application area is the implementation of interaction between real actors and virtual scene components. In [13] an actor feedback system for the production of special effects is described. The feedback is provided with a view-dependent projection system. The head position is sensed in a

[1] result from the TSB *iview* project [12]

Fig. 1 Use of free-viewpoint video to visualise incidents in sport

non-intrusive way from a multi-camera system. The system computes a visual hull and finds the head position in the volumetric reconstruction. Moreover, the 3D data of the actor can be used to compute collisions with virtual components or 'virtual sensors'. This can then be used to trigger actions in a virtual world.

The remainder of this chapter focuses on free-viewpoint video applications, but the approaches discussed in the following sections can be applied to other applications such as those discussed here. Where appropriate, requirements for specific applications will be mentioned.

3 Approaches

This section gives a brief overview of techniques to acquire a 4D model of human action. There are two general classes of approaches: *4D reconstruction* on one hand computes the geometry of the action without specific assumptions about the actors. On the other hand *model-based tracking* usually starts with a 3D shape model and an articulated motion model and then determines motion parameters over time.

A further class of approaches are *image-based interpolation techniques*. These techniques do not compute an explicit 3D volumetric or surface representation of the scene, but synthesise new views directly from images. The techniques can be used to implement some spatio-temporal effects, including slow-motion [41] and interpolation between different camera views [34]. In the following we concentrate on the explicit approaches. Generally, there is no clear distinction between explicit and image-based methods, since sophisticated image-based methods use correspondences, like dense disparity maps, which can be regarded as a kind of 3D representation. On the other hand the explicit model-based approaches make also use of image-based methods to generate high-quality synthesised views. For an overview on image-based approaches see, e.g. [35].

3.1 4D reconstruction

Model-free 4D reconstruction does not make specific assumptions about the scene content. The most common approach is to reconstruct the shape of the scene independently for each time instant. Early approaches adopted well-known 3D reconstruction methods for that purpose, namely two-camera stereo matching that works best for cameras relatively close together, and silhouette-based computation for wide-baseline camera configurations. Both approaches are complementary:

Stereo reconstruction (or stereo vision) aims to find dense pixel correspondences between displaced camera views. The standard approach is window-based matching of (luminance) pixel values between views. Two-camera stereo matching (see for example [32] for an overview) has been used in early work. A number of methods have been developed to combine partial two-view stereo depth maps into a complete surface description, e.g. [27, 20]. Kanade built a half-dome studio system with 51 cameras, using stereo matching to provide a free-viewpoint experience [29]. Stereo vision usually produces poor results at object boundaries, since accurate matching is harder to achieve here.

Silhouette-based approaches rely on known camera parameters and segmentation of the foreground objects against background. In film and TV production, segmentation can sometimes be enforced by use of chroma-keying [36]. The object silhouettes and camera parameters are used to compute an approximation of the scene. Laurentini coined the term *visual hull* for this class of reconstruction, as it only represents a convex approximation of the visible object shape [22]. The visual hull (VH) can be computed very robustly and many 4D reconstruction approaches use it solely, or as an initial solution. A number of algorithms exist to compute the visual hull, section 4.3.1 gives an overview. The visual hull shows a number of typical artefacts. The most important of these are 'phantom volumes'. These areas stay falsely occupied, because they are not visible as background against the object silhouettes. The fewer cameras that are used, the more likely phantom volumes are to appear.

Early work on free-viewpoint video has been implemented predominately based on visual hulls and extensions have been the subject of recent research. Many approaches were developed for controlled studio environments, using approximately 6-16 cameras [26, 13, 37, 47].

Recent work on 3D reconstruction (of static objects) is looking into improved multi-camera stereo methods. Particular progress was made by combining silhouette-based approaches with stereo matching (e.g. [16]). Further advances were made by applying global optimisation techniques, like graph-cuts, e.g. [45, 8]. See [33] for an overview. Some of the advanced methods are applied to 4D reconstruction problems. Starck for example combined stereo-matching with a visual-hull computation [37].

The focus of recent work on 4D reconstruction goes beyond simple concatenation of independently-generated 3D models. Approaches are looking into estimating dense or implicit correspondences of surface points over time. This would open up an number of new applications, like slow-motion, better compression of 4D con-

tent and extended editing capabilities. Dense correspondence can be established by matching surface points [43, 38]. Cagniart, et al. presented a method to match 3D surfaces temporally based only on the geometry [3].

3.2 Model-based tracking

Model-based tracking is an approach to capture 3D human action by tracking motion. Early work targeting communication applications was using simple articulated 3D models, which were divided into partially-rigid components, for example for head-and-shoulder scenes. Kappei and others showed how 3D shape and motion for such models can be derived from monoscopic camera sequences [19, 21].

More recent work is making use of articulated models developed in computer graphics for animation and uses multi-camera set-ups to capture full body shape and motion [46, 4]. These approaches work in two stages:

1. An articulated shape and appearance model is initialised using a 3D modelling technique.
2. Tracking the action by estimating motion parameters from multiple images.

Both sub-tasks benefit from recent advances in 3D modelling and model-based markerless tracking. More recent work also demonstrates the ability to edit the character, i.e.. the shape and appearance of the actor and the animation [44].

An advantage of the model-based tracking approach is that it uses only one 3D shape and appearance model that can be updated during the tracking. Therefore it uses only one 3D geometrical topology and only needs to update form and motion parameters over time. This separation of form and motion results in a compact set of parameters and is compatible with animation models used in post-production and gaming applications. On the other hand model-based approaches are quite limited in capturing local appearance or flexible shape changes, for example caused by moving cloth. Model-free 4D reconstruction is seen to have clear advantages here to capture more realistic shape and appearance of the action.

Another advantage of model-based approaches is the implicit surface definition and its known temporal transformation. That means that the position of surface points over time is known. As mentioned this temporal correspondence or alignment has to be determined for 4D reconstruction by matching.

4 A 4D reconstruction pipeline

This section describes the components of a 4D reconstruction system from a practical point of view. The application of 3D reconstruction in the entertainment industry raises a number of constraints that algorithms have to comply with: Firstly a system used in production must be easy to set up and run. Any elaborate calibration

sessions or extensive parameter adjustment before or during operation must been kept to a minimum, otherwise there is the danger that this will cause disruption of the production process and keep expensive production crews idle. That means that suitable algorithms should have as few parameters as possible that need manual configuration, or ideally have a fully-automatic set-up process.

Another aspect is the processing speed of the system that either allows use under real-time conditions or with very short turnaround times. Therefore methods and algorithms selected for application in media processing should be fast and robust. This section describes a processing pipeline that has been developed and tested with respect to these requirements.

Fig. 2 shows the modular groups and functional blocks of the system. The modular blocks are a logical grouping of functionalities. In practice functional blocks might be implemented across a number of IT components in a distributed system [6]. A challenge of such an implementation is the synchronisation of information and control of the data flow in a production environment.

The user interface controls the operational parameters of the system. This includes set-up of cameras and functional parameters. As many of these parameters have an impact on the visual quality of the synthesised images the operator needs visual feedback at all stages of the processing pipeline.

The capture component reads in the video stream from cameras[2]. Broadcast equipment traditionally uses a reference video signal to synchronise (or 'genlock') video sources such as cameras, and a time-code signal to allow recordings to be time-aligned. The use of time-code often requires extra connections or devices to embed the time signal in the video stream. An approach that avoids this by tight time synchronisation of the IT components is described in [6].

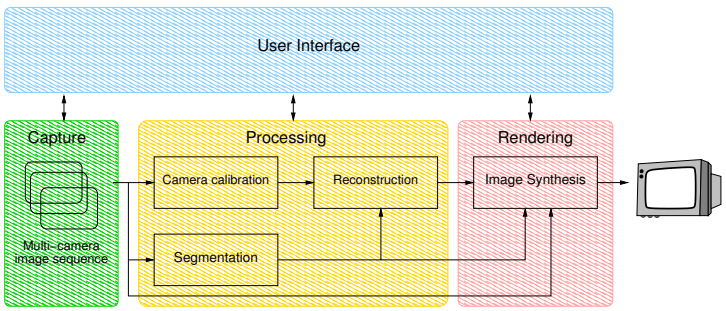

Fig. 2 Flow diagram of the processing pipeline

The processing module computes a 3D model of the scene. The functional blocks include camera calibration, segmentation of objects from the background and 3D reconstruction. These blocks will be described in more detail below.

The replay module renders the captured scene in real-time using the computed 3D model and the original camera images.

[2] This is called 'ingest' in broadcast.

The system as implemented as a distributed system can operate in real-time. The implementation cited in [6] uses one server for two HD broadcast camera streams. The camera calibration (if cameras are moving) and segmentation is computed on these servers. Segmentation and camera parameters are then passed on to another server, which is collecting data and computes the 3D reconstruction. Some applications do not require real-time rendering (such as sport post-match analysis) or do not require the highest quality reconstruction, which is not feasible with real-time capable algorithms. In this case the images are stored locally on the capture servers and the processing is run at a later stage.

4.1 Camera Calibration

Camera calibration is a well studied problem. Most studio-based capture systems assume that the cameras are mounted statically and a calibration can be carried out once before the system is used. For this purpose chart-based calibration is a well-suited approach, see for example [42]. In less controlled environments, i.e. when a chart-based calibration is not possible for whatever reasons, structure-from-motion methods can be applied [15].

During production of broadcast programmes, camera parameters are typically changing all the time. A typical example is in sport coverage, where camera operators are permanently following the action by panning and tilting the camera and changing zoom and focus to capture close-ups. For these kind of scenarios the camera calibration has to estimate the camera parameters frame by frame. Although there are mechanical sensors available for broadcast cameras, a more flexible approach is to estimate camera parameters directly from the image information only. This would also increase the scope of the 3D-multiview application, since it can be applied to image feeds without the need to get access to the cameras.

A suitable approach in sport applications is a line-based approach for the calibration of camera parameters against the pitch lines. This method is very fast, can be computed robustly in real-time on a PC and gives online updates of camera parameters for moving cameras [40]. An overview of this method was presented in Section 4 of the preceding chapter.

4.2 Segmentation

The process of classifying the pixels of a digital image into foreground and background is called segmentation or matting (in the film industry) and keying (in broadcast). The result of this process is stored as the alpha-channel of an image. The main application for keys or mattes is in compositing, for example to exchange the background of a scene with a different image.

A number of approaches have been developed to create a key or matte automatically. Chroma-keying is a long-established technique for special effects in film- and TV-productions (see [36] for an overview) and relies on the subject being filmed in front of a screen with known colour (usually blue or green). This technique is very robust, but limited to controlled environments.

Difference keying is another popular technique that works in two steps: First an image of the background (the background plate) is acquired. The alpha value is then estimated from the difference of an image with subject to the background plate.

More recent work is addressing keying of natural images, i.e. images with varying foreground and background colour. In order to solve the segmentation problem, the user is required to interactively provide hints in form of a tri-map or scribbles to indicate regions of foreground and background (see, e.g. [5, 17, 31, 23]). These techniques are applied to still images with recent extensions to video [2]. However, because these methods are computationally very expensive and require a lot of manual input, their use is usually restricted to offline applications.

For carrying out 3D reconstruction with a basic visual hull algorithm only binary values $\{0,1\}$ are required. However, to achieve high quality results in rendering, a continuous value of alpha in the interval [0..1] is needed. These intermediate values for alpha occur for example in transparent objects or motion blur and in particular at the border of foreground objects when only a part of a pixel belongs to the object (mixed pixels). The continuous alpha values, together with pre-multiplied colour values are used by the image synthesis module that might need to blend textures from different cameras. The image+alpha format is accepted by most graphics systems, for example OpenGL. Pre-multiplied colour values represent the foreground colour of mixed pixels and can be obtained by solving the compositing equation:

$$C = \alpha F + (1 - \alpha)B \tag{1}$$

with the combined colour C, foreground colour F and the background colour B.

This requires solving for both, α and background colour B. For more details see for example [36, 17].

4.2.1 Sport Broadcast

Applying free-viewpoint video to outdoor broadcast, for example for post-match analysis, raises a number of problems not found in a controlled studio environment [12]. For the segmentation of players, colour-based methods such as chroma-keying against the green of football and rugby pitches have been considered. However, the colour of grass on pitches varies significantly. This is due to uneven illumination and anisotropic effects in the grass caused by the process of lawn-mowing in alternating directions. Under these conditions simple chroma-keying gives a segmentation that is too noisy to achieve a high-quality visual scene reconstruction. Difference keying can produce better results and is also able to segment pitch lines, logos and other

markings of the background. The background model or ‘background plate’ can be created by either taking a picture of the scene without any foreground objects or if this is not possible the background plate can be generated by applying a temporal median filter over a sequence to remove moving foreground objects.

Broadcast cameras have a control known as aperture correction or sometimes ‘detail’. The aperture correction is used to ‘sharpen’ an image and is one element that distinguishes the ‘TV-look’ from ‘film-look’. Effectively the correction emphasises high-frequency image components and is therefore a high-boost filter. Figure 3 shows an example of a broadcast image. The image was taken during a rugby match with a Sony HDC-1500 high definition camera.

Fig. 3 Image of a sport scene from a broadcast camera (left) and detail (right)

A high level of aperture correction causes a significant colour shift to pixels close to luminance edges. As can be seen in the close-up on the right of figure 3, this can affect an area about 2-3 pixels around contour edges and leads to incorrect segmentation results using colour-based segmentation methods. The segmentation can be improved by compensating for the effects of the aperture correction. Figure 4 shows a close-up of results from a colour-based segmentation using the original image on the left, and after compensation for aperture correction on the right[11].

Fig. 4 Detail of segmented broadcast picture before compensation for aperture correction (left) and after compensation (right)

4.3 Reconstruction

The reconstruction module computes a temporal geometric model of the action. For real-time applications and wide-baseline camera setups, methods based on visual hull computation are generally used to compute a 3D reconstruction for each time instance.

Applications that require best visual quality and can afford offline processing may use more sophisticated reconstruction algorithms to extract temporally-consistent 4D data from the input data, as outlined in section 3.

In this section we discuss aspects of visual hull (VH) computation. The computation of the visual hull from 2D silhouettes is very robust, fast and well-suited for modelling of human action. The computation of the visual hull, also known as shape-from-silhouette, is equivalent to an intersection of the back-projected 2D silhouettes (visual cones) in 3D. A number of approaches to compute this intersection have been suggested in the literature that differ in the underlying data structure and the processing.

4.3.1 Algorithms for visual hull computation

A basic implementation to compute the visual hull based on a volumetric data representation is described in algorithm 1:

Algorithm 1 Basic volumetric visual hull computation

```
for all voxel in V_def do
    voxel := true
end for
for all c in listofcameras do
    for all voxel = true do
        fp = FootPrintTest(c, voxel)
        if fp = AllZero then
            voxel := false
        end if
    end for
end for
```

This algorithm runs over all elements (voxels[3]) of a 3D array and projects each voxel into the 2D silhouette images and tests whether its footprint is on foreground or background. If the voxel footprint is on background it will be set to 'false'. From this volumetric data a surface description can be generated with an iso-surface extraction, for example using the marching cubes algorithm. Using a 3D array in this simple algorithm requires a high number of projections and footprint tests. To reduce this effort the use of hierarchical processing using octrees as a representation has been proposed [28, 39].

[3] Volumetric elements

Matusik et al. describe an algorithm that synthesises new views of an object directly without generating a surface model. Their algorithm samples the object based on lines along the viewing frustum of the new view and computes a VH for this parametrisation [26]. Grau describes an algorithm to compute a surface model using sets of line segments [13].

Matsuyama et al. describe an algorithm that computes a VH by projecting the silhouette images into the volumetric space[24]. This is achieved with a 2D transformation into planes of a volumetric 3D array and intersection.

Another approach computes a surface description directly without going into an intermediate volumetric representation[25, 7]. These algorithms reproject the boundaries or edges of the 2D silhouettes using the inverse camera projection matrix and find the intersections in 3D to build up a polyhedral surface description.

Although the algorithms discussed above aim to compute the visual hull of an object, they all have different characteristics. The (classical) volumetric algorithms with iso-surface generation are very popular, because of their low complexity. On modern machines they are also fast to compute. Furthermore, they are very robust against segmentation errors. A typical problem is that the surface models look crude or rough. This is an aliasing artefact introduced by the iso-surface extraction on binary voxel values. Grau [9] analyses this problem and suggests super-sampling to enhance the accuracy of the reconstruction without increasing the complexity of the surface model.

Other algorithms, like the direct polyhedral surface computation are usually more complex, but run very fast. However, they are generally more sensitive to errors in the 2D silhouettes than the volumetric methods.

4.3.2 Alternative computation strategies

The visual hull only provides an approximation to the 3D shape of foreground objects, but systems with a high number of cameras under controlled conditions can produce visual results that meet the needs of some applications, like simple special effects in TV shows. However some typical artefacts, like ‘phantom volumes’, become more pronounced when only a low number of cameras is used to capture the action. Another source of reconstruction errors comes from errors in the camera calibration. Several strategies have been developed to cope with the imperfections of VHs:

Erroneous camera calibration parameters are manifested by an erosion of the computed surface models. This can lead to loss of limbs of actors. More robustness against camera calibration errors can be achieved by over-estimating the visual hull. This results in a conservative visual hull (CVH) that is bigger than the object. Figure 5 shows an example of VH and CVH computed for a rugby game. The conservative effect can be achieved by dilating the 2D silhouettes. The amount of dilation should correspond to the maximal error (in pixels) of the camera calibration. During rendering, the camera images and the alpha channel from the segmentation are used to improve the shape of objects.

The VH or CVH can be used to initialise refinement methods. Guillemaut describes a method to compute a depth map per camera [14]. The process starts with an inital depth map derived from a VH. An algorithm formulated as a graph-cut problem then computes a refined depth map. Implicitly this methods also computes a refined segmentation. Fig. 5 compares results of this methods with VH, CHV and stereo matching.

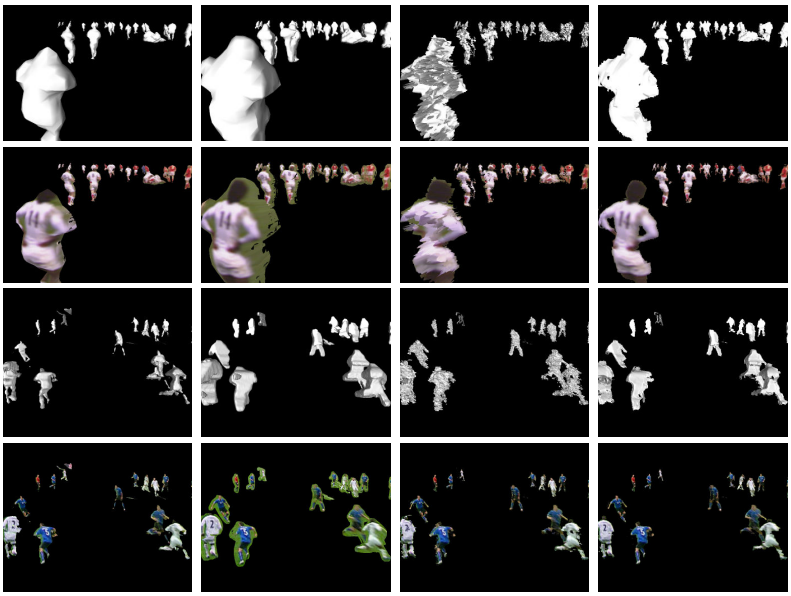

Fig. 5 Example of reconstruction results on rugby (top) and soccer (bottom) data from [14] (©IEEE 2010).

4.4 Rendering

The replay module uses the 3D models of the scene together with the original camera images to produce a novel view of the scene. Action captured and processed with model-based tracking, as described in section 3.2 can often be rendered with standard animation software.

Action captured with 4D reconstruction approaches often uses specific rendering methods. Matusik et al. describe a direct image synthesis approach [26] that does not produce intermediate volumetric or surface geometry. Wuermlin et al. use point-based rendering as an alternative to a surface representation [47].

A method based on the use of surface geometry and view-dependent texture mapping can give good results: Three or more camera images are used and blended together. Cameras closer to the synthetic viewpoint get a higher weight. One option to achieve this is to use a simple formula based on the angle between the virtual camera, the real camera and the scene interest point.

The main advantage of view-dependent texture mapping is that it can mask many imperfections or errors in the reconstructed shape. Furthermore, appearance (including specularities of surfaces) can be reproduced to a certain extent. The results in figures 1 and 5 are rendered using view-dependent texture mapping.

5 Conclusions

With recent advances in 3D and 4D reconstruction techniques, the use of these techniques in the entertainment industry is starting to emerge. The degree of operator input for these methods is dictated by production budgets and whether the programme is produced live or offline. The production pipeline described in the previous section was developed for use in broadcast, with a minimum of required user input and (potentially) allows real-time operation.

The method described here was based on the computation of visual hulls, mainly because of the requirement for real-time operation. The usefulness of this approach has been demonstrated for the visualisation of sport incidents. However, in this case the synthesised views are quite distant from the reconstructed objects.

For applications that require closer views of the action, for example in close-range studio setups, more sophisticated methods are required, as reconstruction errors become more disturbing. Advances in adaptation of global minimisation methods for 3D reconstruction as discussed earlier are also the most promising candidates to improve 4D reconstruction. Finally a combination with temporal alignment methods will open new applications as the resulting 4D data sets will better fit the graphics production pipelines used in the entertainment industry.

References

1. Time slice films. http://www.timeslicefilms.com/.
2. X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. In *ACM SIGGRAPH 2009 papers*, pages 1–11. ACM, 2009.
3. Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Probabilistic deformable surface tracking from multiple videos. In *ECCV 2010*, pages 326–339, 2010.
4. J. Carranza, C. Theobalt, M Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Trans. on Computer Graphics*, 22(3), July 2003.
5. Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271. IEEE Computer Society, December 2001.
6. J. Easterbrook, O. Grau, and P. Schübel. A system for distributed multi-camera capture and processing. In *Proc. of CVMP*, 2010.
7. Jean-Sebastien Franco and Edmond Boyer. Exact polyhedral visual hulls. In *In British Machine Vision Conference*, pages 329–338, 2003.
8. Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *International journal of computer vision*, 81(1):53–67, 2009.
9. Oliver Grau. 3D sequence generation from multiple cameras. In *Proc. of IEEE, International workshop on multimedia signal processing 2004*, Siena, Italy, September 2004.
10. Oliver Grau. A 3D production pipeline for special effects in tv and film. In *Mirage 2005, Computer Vision/Computer Graphics Collaboration Techniques and Applications*, Rocquencourt, France, March, 2005. INRIA.
11. Oliver Grau and Jim Easterbrook. Effects of camera aperture correction on keying of broadcast video. In *Proc. of the 5rd European Conference on Visual Media Production (CVMP)*, 2008.
12. Oliver Grau and et al. A robust free-viewpoint video system for sport scenes. In *Proc. of 3DTV-Conference*, 2007.
13. Oliver Grau, Tim Pullen, and Graham A. Thomas. A combined studio production system for 3-d capturing of live action and immersive actor feedback. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):370–380, March 2004.
14. J.Y. Guillemaut, J. Kilner, and A. Hilton. Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 809–816. IEEE, 2010.
15. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
16. C. Hernández Esteban and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.
17. Peter Hillman, John Hannah, and David Renshaw. Foreground/background segmentation of motion picture images and image sequences. *IEE Transactions on Vision, Image and Signal Processing*, 152(4):387–397, August 2005.
18. T Kanade et al. Eyevision at super bowl xxxv. web, 2001.
19. Frank Kappei and C.-E. Liedtke. Ein verfahren zur modellierung von 3d-objekten aus fernsehbildfolgen. In *Mustererkennung 1987, 9. DAGM-Symposium*, pages 277–281, 1987.
20. R. Koch. Model-based 3-d scene analysis from stereoscopic image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 49(5):23–30, 1994.
21. Reinhard Koch. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):556–568, 1993.
22. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 150–162, 1994.
23. A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 228–242, 2007.
24. Takashi Matsuyama, Xiaojun Wu, Takeshi Takai, and Toshikazu Wada. Real-time dynamic 3-d object shape reconstruction and high-fidelity texture mapping for 3-d video. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):357–369, March 2004.

25. W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Proc. of 12th Eurographics Workshop on Rendering*, pages pages 116–126, 2001.
26. Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374. ACM Press, 2000.
27. M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.
28. M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40:1–29, 1987.
29. Peter Rander, P. J. Narayanan, and Takeo Kanade. Virtualized reality: constructing time-varying virtual worlds from real world events. In *IEEE Visualization*, pages 277–284, 1997.
30. Doug Roble and Nafees Bin Zafar. Don't trust your eyes: Cutting-edge visual effects. volume 42, pages 35–41, 2009.
31. Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
32. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.
33. S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. 1:519–528, 2006.
34. S.M. Seitz and C.R. Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30. ACM, 1996.
35. Heung-Yeung Shum, Sing Bing Kang, and Shing-Chow Chan. Survey of image-based representations and compression techniques. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(11):1020–1037, November 2003.
36. Alvy Ray Smith and James F. Blinn. Blue screen matting. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 259–268, New York, NY, USA, 1996. ACM.
37. J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *Proc. of ICCV*, pages 915–922, 2003.
38. J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
39. Richard Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.
40. Graham A. Thomas. Real-time camera pose estimation for augmenting sports scenes. In *Proc. of 3rd European Conf. on Visual Media Production (CVMP2006)*, pages 10–19, London, UK, November 2006.
41. Graham A. Thomas and H. Y. K. Lau. Generation of high quality slow-motion replay using motion compensation. In *Proc. of International Broadcasting Convention*, 1990.
42. R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics and Automation*, 3(4):323–344, 1987.
43. S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 475–480, 2005.
44. D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*, pages 1–9. ACM, 2008.
45. G. Vogiatzis, C.H. Esteban, P.H.S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE transactions on pattern analysis and machine intelligence*, pages 2241–2246, 2007.
46. S. Weik, J. Wingbermühle, and W. Niem. Automatic creation of flexible antropomorphic models for 3D videoconferencing. In *Computer Graphics International, 1998. Proceedings*, pages 520–527. IEEE, 1998.
47. S. Würmlin, E. Lamboray, O.G. Staadt, and M.H. Gross. 3D Video Recorder: a System for Recording and Playing Free-Viewpoint Video. In *Computer Graphics Forum*, volume 22, pages 181–193. Wiley Online Library, 2003.