Improving Free-Viewpoint Video Content Production Using RGB-Camera-Based Skeletal Tracking

Andrew MacQuarrie, Anthony Steed*
Department of Computer Science, University College London, United Kingdom

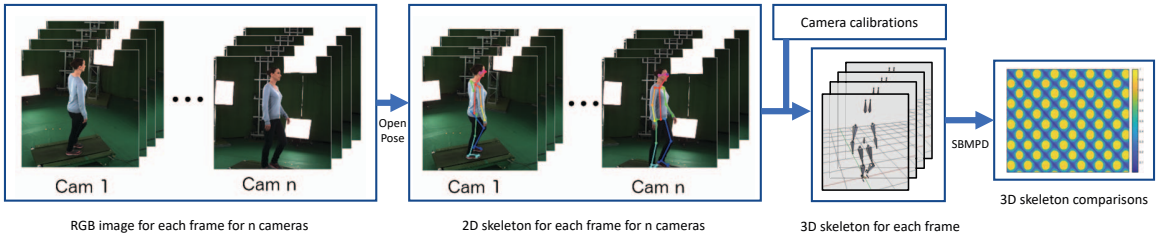

Figure 1: Flow of data from capture to comparison.

Abstract

Free-Viewpoint Video (FVV) is a type of volumetric content in which an animated, video-textured 3D mesh of a character performance is constructed using data from an array of cameras. Previous work has demonstrated excellent results when creating motion graphs from FVV content, but these techniques are often prohibitively expensive in practice. We propose the use of skeletons to identify cut points between FVV clips, allowing a minimal set of frames to be processed into a 3D mesh. While our method performed with 2.8% poorer accuracy than the state-of-the-art for our synthetic dataset, cost and processing time requirements are dramatically reduced.

Index Terms: [**Computing methodologies**]—**Shape analysis**; Virtual reality; Motion capture; Motion processing

1 Introduction and Related Work

In Free-Viewpoint Video (FVV), a number of inward-facing cameras are arranged around a central character performance [5]. Data from these camera views is processed to produce an animated, video-textured 3D mesh, which can then be integrated into a virtual 3D scene. In our work, we rely on a reconstruction technique similar to that described by Collet et al. in [3].

FVV is an increasingly popular method for producing immersive content. While FVV has many advantages as a content production tool (e.g. realistic secondary motions such as clothes) it also has issues. Two of the main issues are the cost and processing time. Aside from the standard costs of media production, FVV requires an expensive data-processing step to turn the high volume of 2D video data into an animated 3D mesh. This cost results from the time required by a server farm to process the 3D model.

One mechanism to reduce production costs is through content reuse. For example, processing a shorter duration of FVV content into a 3D mesh, and then looping it in such a way as to disguise the loop point. An example of this is a walk cycle, where the periodic movement can allow a shorter video to be looped. This technique can be considered analogous to Video Textures [8].

Another issue with FVV is that the content is fixed at the point of filming, and therefore lacks the interactivity that rigged avatar performances can provide. One mechanism to combat this issue is through the use of motion graphs [6]. In motion graphs, clips of captured motion (e.g. walking, jumping, etc) are cut together, allowing the sequence of motion elements to be varied at runtime.

Both motion graphs and loops have been explored in FVV content before with excellent results [2, 7]. To create loops and motion graphs, good points must be identified in which the cut can be disguised. This requires that the character performance exhibit the same shape and dynamics at these moments. A large amount of research has been done to explore how good match points can be identified from the 3D mesh (e.g. [4]).

We argue that comparisons performed on the 3D mesh happen too late in the pipeline, as processing these 3D meshes has a high cost. Performing comparisons on the 3D meshes requires that all frames be processed before comparison, with large numbers of meshes being discarded after comparisons indicate they are not suitable due to poor quality match points.

To counteract this wasteful process, match points must be identified before 3D reconstruction. Here, we propose identifying match points by comparing 3D skeletons derived from multi-view RGB camera data. Using receiver operating characteristic (ROC) curves, we evaluate a skeleton-based technique for identifying match points in a synthetic dataset. Additionally, we demonstrate our technique works on real-world data. Based on this analysis, we propose that skeletons represent a viable mechanism for reducing production costs and processing time.

2 Skeleton-Based Match Point Detector

The overall structure of our Skeleton-Based Match Point Detector (SBMPD) system is shown in Figure 1. We consider a volumetric capture studio with n calibrated RGB cameras (in our case, $n = 53$). For each of these camera views, we identify the 2D skeleton using OpenPose [1]. We discard any joint positions that have been identified by OpenPose with low confidence. Using the intrinsic and extrinsic calibration matrices for each camera, a 2D joint position becomes a ray from each camera. The 3D position of a joint is taken to be the point with the minimum sum of squared distances to all rays for that joint. The output of this process is the 25 joints of a 3D skeleton per frame.

3D skeletons are then compared to assess frame similarity. The 3D skeletons are first aligned. To ensure actions remain on the ground plane, rotations are only calculated around the "up" vector. Each entry in a 3D skeleton similarity matrix $S_s(i, j)$ is taken to be the summed Euclidean distance of the joint locations be-

*andrew.macquarrie.13@ucl.ac.uk, a.steed@ucl.ac.uk

978-1-7281-6532-5/20/$31.00 ©2020 IEEE
DOI 10.1109/VRW50115.2020.00237

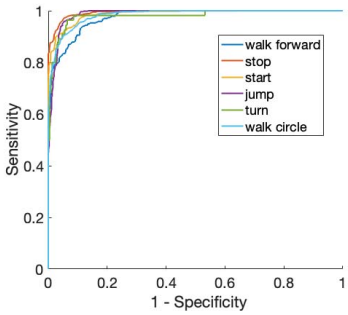

Figure 2: ROC curve showing SBMPD performance for self-similarities against TGT (temporal window = 2).

tween the 3D skeletons for frames i and j. The similarity matrix is then temporally filtered to include dynamics as in [8]. In this technique, static frame comparisons over a temporal window are incorporated into a frame's measure. In practice, this is achieved by applying a convolution to S_s [8].

3 Evaluation Methodology

We evaluated the accuracy of our SBMPD when identifying match points between frames using ROC curves on synthetic data, as in [4]. This synthetic data was constructed by applying six motion capture performances to a rigged avatar. To ensure the synthetic data approximates the inputs for our skeleton-based technique, we re-created the layout of the physical RGB cameras of the volumetric capture rig in Blender version 2.79b. From the synthetic data, we generated a temporal ground truth (TGT) as in [4]. These ground truth distances between frames were then normalized into the range [0,1]. As in [4], we then use a threshold value of 0.3 to create a ground-truth binary classification matrix.

We also perform an evaluation of our SBMPD on real data. As the topology of the 3D mesh in our FVV data is not fixed so a TGT cannot be established, ROC curve analysis is not appropriate. Instead, we analysed visually how well our skeleton-based method works through heatmaps and examples of identified match points.

4 Results

An ROC curve showing the SBMPD self-similarity performance against the TGT binary classification matrix is shown in Figure 2. The standard way to report the accuracy of a discriminator modelled using ROC curves is the area under the curve (AUC), where an AUC of 1 indicates ideal discrimination for a dataset. The ROC curve shown in Figure 2 shows an AUC of 0.988 for self-similarity comparisons. We achieved an AUC of 0.972 across all pairwise comparisons.

We also tested our SBMPD on real data. We take as an example the creation a motion sequence, in which we want the character to start walking from a standing position. To achieve this, we will need to identify a match point between a "start walking" clip and a "walking" clip. Figure 3 shows normalized SBMPD scores for transitions between frames in a "start walking" clip and a "walking" clip. As can be seen in Figure 3, no suitable transitions are identified for earlier frames in the "start walking" clip. This is correct – as the actor was standing still before they started walking, there would be no good cut into a walking clip from these frames. Later in the "start walking" clip, suitable match points are found in the "walking" clip at frames where the shape and dynamics of the motions would allow a reasonable cut. In our accompanying video, we show a cut identified by our SBMPD as being suitable. We also show an entire sequence composed of "start", "walk" and "stop" motion clips in our accompanying video.

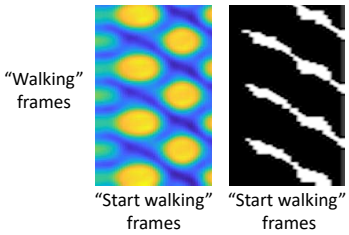

Figure 3: "Start walking" against "walking". Left: heatmap showing normalized SBMPD score between frames (a darker colour means frames are more similar). Right: The heatmap thresholded at 0.3.

5 Comparison Against the State-of-the-Art

The state-of-the-art in terms of match point identification could be considered to be techniques that employ temporally consistent meshes, as these allow the trivial comparison of shapes using the Euclidean distance between corresponding vertices (e.g. [2]). This is analogous to the way our ground truth was created for our synthetic dataset. In this way, temporally consistent meshes could be considered to produce ideal identification of match points, with an AUC of 1, verses our AUC of 0.972 across all shots. Therefore our SBMPD could be considered to perform with 2.8% poorer accuracy than the state-of-the-art for our synthetic dataset.

This decrease in accuracy comes with substantial improvements in cost and time requirements, however. The mesh processing cost for state-of-the-art mesh comparison techniques increase linearly with the amount of content being compared, while mesh processing costs for our SBMPD system increase linearly with the amount of content required for the final output. In a real-world example, this represented a 6.5-fold cost and processing time reduction, although this reduction can be substantially more depending on the amount of content captured and the final output duration.

Acknowledgments

This work was supported in part by grants EP/N509577/1 and EP/M029263/1 from the UK Engineering and Physical Sciences Research Council (EPSRC). The authors would like to thank Dimension Studio and Digital Catapult for their support.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[2] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton. Interactive animation of 4d performance capture. *IEEE transactions on visualization and computer graphics*, 19(5):762–773, 2012.

[3] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015.

[4] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *International Journal of Computer Vision*, 89(2-3):362–381, 2010.

[5] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1):34–47, 1997.

[6] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, p. 51. ACM, 2008.

[7] F. Prada, M. Kazhdan, M. Chuang, A. Collet, and H. Hoppe. Motion graphs for unstructured textured meshes. *ACM Transactions on Graphics (TOG)*, 35(4):108, 2016.

[8] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 489–498. ACM Press/Addison-Wesley Publishing Co., 2000.