Exploring the Use of Skeletal Tracking for Cheaper Motion Graphs and On-Set Decision Making in Free-Viewpoint Video Production

Andrew MacQuarrie
andrew.macquarrie.13@ucl.ac.uk
University College London
United Kingdom

Anthony Steed
a.steed@ucl.ac.uk
University College London
United Kingdom

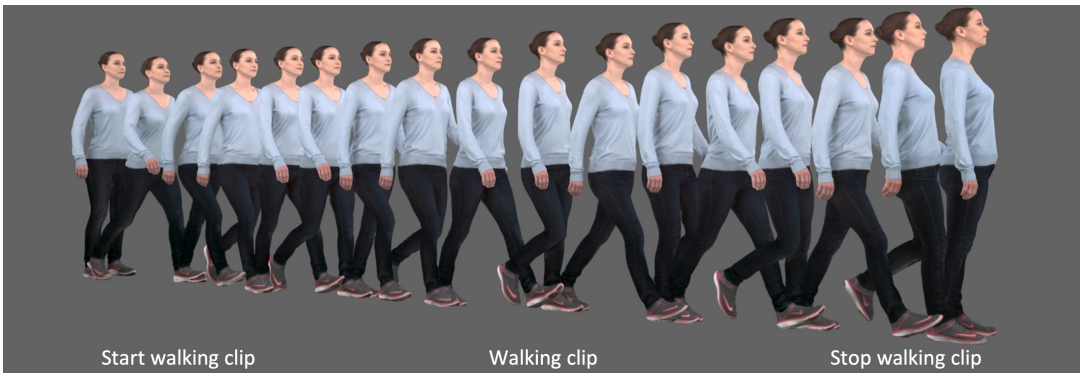

Figure 1: 3D meshes from "start walking 1", "walking on treadmill 2", and "stop walking 1" clips combined into a motion sequence. The frames at which these clips are cut together were identified using our skeleton-based match point detector system.

ABSTRACT

In free-viewpoint video (FVV), the motion and surface appearance of a real-world performance is captured as an animated mesh. While this technology can produce high-fidelity recreations of actors, the required 3D reconstruction step has substantial processing demands. This means FVV experiences are currently expensive to produce, and the processing delay means on-set decisions are hampered by a lack of feedback. This work explores the possibility of using RGB-camera-based skeletal tracking to reduce the amount of content that must be 3D reconstructed, as well as aiding on-set decision making. One particularly relevant application is in the construction of Motion Graphs, where state-of-the-art techniques require large amounts of content to be 3D reconstructed before a graph can be built, resulting in large amounts of wasted processing effort. Here, we propose the use of skeletons to assess which clips of FVV content to process, resulting in substantial cost savings with a limited impact on performance accuracy. Additionally, we explore how this technique could be utilised on set to reduce the possibility of requiring expensive reshoots.

CCS CONCEPTS

• **Computing methodologies** → **Shape analysis**; *Virtual reality*; *Motion capture*.

CVMP 2020, Dec. 17–18, London, UK
2020. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

shape analysis, free-viewpoint video production

ACM Reference Format:
Andrew MacQuarrie and Anthony Steed. 2020. Exploring the Use of Skeletal Tracking for Cheaper Motion Graphs and On-Set Decision Making in Free-Viewpoint Video Production. In *Proceedings of the 16th ACM SIGGRAPH European Conference Visual Media Production (CVMP 2020).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Free-viewpoint video (FVV) is a content creation technique in which the motion and surface appearance of a performance are captured by a number of cameras. RGB and sometimes infrared cameras are arranged around a performer, facing inwards, to capture that person from multiple perspectives simultaneously. Reconstruction techniques are then used to process these 2D views into an animated mesh with a video texture. This 3D asset can then be used (e.g. in a game engine such as Unity) for creating experiences viewable in virtual reality (VR) or augmented reality (AR) displays. As FVV characters are high fidelity, but support viewing with six degrees-of-freedom, FVV is becoming a popular content creation tool for AR/VR experiences. As this content is captured from the real world, it requires no pre- or post-hoc 3D modelling. It can result in media that is in some ways more realistic than motion-captured performances applied to rigged avatars. It also automatically captures secondary motions (e.g. clothes) and can capture appearance as well as movement.

One barrier that FVV faces in becoming a major content production tool is the cost of turning the 2D views from the camera rig into a 3D mesh, a process that we refer to as *3D reconstruction*. 3D reconstruction is an enormously data intensive process, requiring the compute power of render farms for long periods of time. This translates directly into cost, with this cost being linearly tied to the amount of 3D reconstructed content. So while a day of filming in an FVV studio can result in a substantial amount of *captured* content, production budgets require the careful selection of which clips to 3D reconstruct into mesh assets.

As production costs increase with the amount of content that is 3D reconstructed, one possible technique to reduce costs is through content reuse. For FVV this could mean looping repetitive movements such as in walk cycles. The cyclic motion of walking means a short clip can be looped to give the impression of a much longer sequence. Hereafter we refer to this process as *video loops*. Another technique to allow content reuse is through Motion Graphs. In Motion Graphs, short clips of content (e.g. walking, turning, etc.) are sequenced together interactively. Hereafter, we refer to these as *video sequences*. Similarly to video, FVV is often considered to be fixed at the point of filming, meaning FVV content can lack the interactivity that rigged avatars possess. As a result, video sequences also have the potential added benefit of allowing FVV content to respond to real-time events by sequencing together appropriate clips.

To create both video loops and sequences, we must find points at which the cut can be disguised. This requires that the shape and dynamics of the character performance be similar at these moments. We call these moments *match points*. As we will discuss in Section 2, a large amount of research has been done to explore how good match points can be identified from the 3D mesh. We argue that these state-of-the-art methods are cost prohibitive in most production contexts, as the requirement to perform 3D reconstruction on all captured content before searching for match points is extremely wasteful.

Through the identification of match points before 3D reconstruction, our method can avoid processing large numbers of frames which are later deemed unsuitable. We propose identifying match points by comparing 3D skeletons derived from multi-view RGB camera data. To assess the performance of our method against the state-of-the-art, we use receiver operating characteristic (ROC) curves to identify match points in a synthetic dataset that we constructed. To show that our method also works on real-world data, we visualise results using heatmaps and visual examples. A small drop in performance against the state-of-the-art, coupled with substantial cost savings, indicates that our skeleton-based technique may be a viable mechanism for reducing costs and processing time when producing types of FVV content for which these techniques are suitable (e.g. when building Motion Graphs, content with loopable motion, etc.). Our method can also help with creating controllable characters in the context of FVV, opening up this production technique to a new group of creators looking to make interactive content. We also show that these techniques may be suitable for use on set. On-set usage would allow for takes to be assessed for the quality of match points available, i.e. takes with poor or ambiguous quality match points could be re-filmed immediately, with feedback provided to the actor to ensure similar body pose and dynamics. This would further reduce costs by minimizing the need to re-shoot at a later date, and improve the quality of the end product through better match points. This work builds on a previous poster on our technique, presented at IEEEVR, by locating it within the research landscape through background research and providing a full description of the process for reproducibility [MacQuarrie and Steed 2020]. This work additionally explores further examples and use cases. These include using our technique to generate scores to assess match point quality, as well as the feasibility and potential uses of deploying it on set.

2 RELATED WORK

2.1 Free-Viewpoint Video

Free-Viewpoint Video (FVV) was pioneered by the work of Kande et al. in [Kanade et al. 1997]. A seminal series of works brought high-quality FVV to fruition [De Aguiar et al. 2008; Narayanan et al. 1998; Starck and Hilton 2003, 2007; Starck et al. 2005; Vlasic et al. 2008]. In our work, we rely on a reconstruction technique similar to that described in [Collet et al. 2015]. Previous works were built on in [Collet et al. 2015] – through multimodal reconstruction and saliency-based adaptive meshing – to produce improvements in reconstruction robustness and visual quality. In this method, a character performance is captured by an array of inward-facing RGB and infrared cameras, arranged in a cylinder. For each camera view, green screen and depth-from-stereo techniques are used to segment the character from the background. A 3D mesh of the character is then built for each frame using 3D reconstruction algorithms, which is textured using data from the RGB cameras. This results in a video-textured 3D mesh that captures appearance and surface dynamics, but in which the meshes are temporally unstructured.

There has been work exploring the creation of temporally consistent meshes. In one approach, a parameterized template mesh is fitted [Carranza et al. 2003; Loper et al. 2015]. Another technique is to deform the mesh from a single frame over time to represent the changing shape [Ahmed et al. 2008; Budd et al. 2013; Cagniart et al. 2010; Huang et al. 2011; Mustafa et al. 2016; Tung and Matsuyama 2010]. Both of these techniques can produce excellent results, and a temporally consistent mesh brings some advantages over unstructured meshes. Temporal consistency allows the use of mesh editing techniques, reduces data bandwidth requirements, and means shape comparisons between frames can be trivially calculated using the Euclidean distance between corresponding vertices [Casas et al. 2012a,b]. However, model-based approaches may produce artifacts when deformations cannot be well represented by the model's parameter space [Collet et al. 2015]. Likewise, mesh-deformation techniques may produce artifacts in the case of large or rapid shape changes, or if the mesh drifts across the object's surface during deformation [Bojsen-Hansen et al. 2012; Casas et al. 2012b; Mustafa et al. 2016]. In the FVV system in use in this work, the mesh topology of the resulting 3D avatars are temporally unstructured [Collet et al. 2015].

2.2 Video Loops and Sequences

The seamless looping of 2D videos to create the illusion of a single, endless video was proposed in [Schödl et al. 2000], who called this

technique Video Textures. Video Textures leverage similarities in the appearance and dynamics between frames in a video sequence, cutting backwards and forwards in the video in a way that is essentially imperceptible. This drew from work on concatenating video clips to generate longer sequences in [Bregler et al. 1997], and was extended to portray human motion in [Flagg et al. 2009].

An analogous technique has been applied in the realm of 3D motion capture data [Arikan and Forsyth 2002; Kovar et al. 2008; Lee et al. 2002; Tanco and Hilton 2000]. In these techniques, a corpus of motion capture data is examined to identify similar appearance and dynamics between frames. These match points between frames can then act as edges in a "Motion Graph" of motion elements, allowing motions to be sequenced together in a plausible way at run-time. Further work on Parameterized Motion Graphs allowed even greater control of character movements [Heck and Gleicher 2007; Rose et al. 1998].

There has been a large amount of work looking at how similar techniques can be applied in the field of FVV [Casas et al. 2012b; Prada et al. 2016]. In order for motion graph techniques to be successful in the context of FVV, there are two main components required. First, good match points must be established in which the transition between frames will be least noticeable. Secondly, the frames must be blended together. In our work, we focus on how to identify good match points between frames in FVV sequences. As such, we do not blend our meshes together. However, making effective blends between FVV clips when the topology of the mesh is unstructured has been considered in [Prada et al. 2016].

2.3 Match Point Identification

Our work is concerned with identifying points at which cuts between shots – or, in the case of motion loops, cuts back into the same shot – will be least noticeable. To allow a convincing blend to be achieved, frames must be found in which the character is in near-identical poses. For temporally consistent meshes, shape similarity can be calculated as the Euclidean distance between corresponding vertices [Casas et al. 2012a,b]. For unstructured meshes, shape similarity is harder to assess. There has been a great deal of work in the assessment of shape similarity, both to facilitate 3D object search and retrieval and in the context of 3D video creation [Huang et al. 2010a; Kazhdan et al. 2003; Shilane et al. 2004; Tangelder and Veltkamp 2004]. Here, we present an overview of the works relevant to our technique, and describe how our skeleton-based shape comparison system adds to previously utilized methods in the context of 3D video creation.

Starck and Hilton explored how FVV sequences could be cut together by manually identifying match points [Starck and Hilton 2007]. Xu et al. considered automatic detection through the use of 3D histograms [Xu et al. 2006]. In [Huang et al. 2010a], Huang et al. compared various shape similarity metrics in the context of identifying match points between unstructured mesh sequences of human motion. They compared a number of 3D shape descriptors, including Shape Histograms [Ankerst et al. 1999] and Shape Spherical Harmonics [Kazhdan et al. 2003]. Using ROC curves, they compared these techniques using both synthetic and real datasets. Huang et al. concluded that Shape Histograms performed best, and reported an optimal binning arrangement [Huang et al. 2010a, 2007].

Huang et al. continued this investigation in [Huang et al. 2010b] by comparing Shape Histograms against Reeb Graphs [Tung and Schmitt 2005], a skeleton-based shape similarity metric. Their findings concluded that both perform similarly on real and synthetic datasets. While the Reeb Graph used in [Huang et al. 2010b] is skeleton-based, the nodes of the graph encode information about the 3D mesh such as local surface area. As a result, these Reeb Graphs require the 3D meshes be reconstructed before they can be compared. Huang et al. also explored simplifying motion graph searches using skeletons that were manually annotated on the 3D mesh [Huang et al. 2015].

The performance of other shape descriptors for the purpose of match point detection was explored in [Veinidis et al. 2017]. Veinidis et al. explored several other shape descriptors, e.g. PANORAMA [Papadakis et al. 2010], in which a 3D shape is projected onto panoramic cylinders before comparison. Similarly to Huang et al., ROC curves were used to evaluate performance, and all investigated shape descriptors required the complete reconstruction of the 3D shape before comparisons could be made. In the work of Prada et al. on motion graphs for unstructured meshes [Prada et al. 2016], match points in FVV were identified through exhaustive search carried out on the meshes, using a technique for comparing 3D shapes originally proposed by Funkhouser et al. [Funkhouser et al. 2004]. Surface colour information was considered by Huang et al. in their work on shape-colour histograms [Huang et al. 2015].

Match points represent points in two FVV sequences where the 3D shape has similar appearance and dynamics. The above works generally handle appearance similarity, and then extend this to dynamics by ensuring a similar appearance between neighbouring frames over a temporal window (e.g. [Huang et al. 2009, 2010a]). We discuss this process in more details in Section 3.

As shown here, there is a large body of work that explores shape similarity metrics in the context of FVV production. These works have been shown to produce excellent results. Each of these works, however, requires the reconstructed 3D meshes to be available before the comparisons can be performed. In this work, we explore the use of 3D skeletons as a way to avoid performing the expensive and time-consuming processing step prior to comparisons. As less information is available at the comparison stage, it is anticipated that our method may yield less accurate results than the state-of-the-art mesh comparisons, but reduce the overall cost and time of production.

3 SKELETON-BASED MATCH POINT DETECTOR

The use of skeletons to describe the shape of a FVV sequence may initially seem problematic. As FVV techniques capture surface dynamics not represented in traditional motion capture, this nuance will be lost in a skeleton-based descriptor. However, the current generation of FVV struggles to process thin elements such as loose hair and fabric. In practice, actors generally tie their hair up, and wear relatively well fitting clothing. As a result of the topology, our method should not result in "false negatives", but only "false positives", with the errors likely contained in secondary motions.

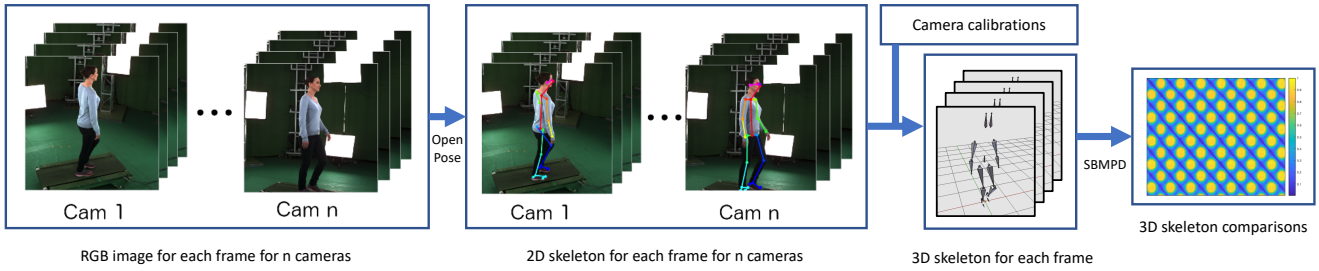

Figure 2: Data flow through our SBMPD system, from RGB image data input to 3D skeleton comparison output.

As a result, skeletons may often be a suitable descriptor to establish match points. In Section 8, we discuss possible ways in which our method could be extended to detect secondary motions that do not fit the skeleton.

As FVV captures the surface appearance of a performance, it would not be acceptable for the actors to wear motion capture markers. Additionally, the volumetric capture technology in use here uses semi-structured infrared light as part of its 3D reconstruction technique [Collet et al. 2015], so there would be interference from off-the-shelf RGB-D cameras such as Kinects as these also use infrared light. Instead, our skeleton-based match point detector (SBMPD) identifies the 3D location of a performer's joints using an array of calibrated RGB cameras. These 3D skeletons are then compared to identify good match points between frames. The overall structure of our method is shown in Figure 2, and we now describe each of its components in more detail.

2D Skeletons. Our system works on data captured in an FVV studio with n RGB cameras (in our case, $n \approx 50$). RGB cameras must be calibrated. Using OpenPose [Cao et al. 2017], 2D skeletons are detected from each RGB camera for each frame. Each RGB camera records images of 2048x2048 pixels. The output of this process is the 2D location of 25 joints identified for each frame for each camera.

3D Skeletons. The n 2D skeletons for each frame are then combined to create a single 3D skeleton per frame. As OpenPose provides a confidence value for each joint, we use a cutoff value of 0.5 to ignore joints that were identified with low confidence. Using the camera calibration matrices, the 2D location of each joint in each camera can be considered a ray in 3D space. Each joint's 3D coordinate is taken to be the point with the minimum sum of squared distances to all rays for that joint, which we find using an open source Matlab function[1].

Skeleton comparison. Similarity between frames is then assessed by comparing these 3D skeletons. First, we align the 3D skeletons using an open source implementation of the Kabsch algorithm[2]. As freely rotating the character around all axes may cause animations to leave the ground plane (e.g. walking into the sky or through the ground), we instead restrict rotation calculations to be purely around the "up" vector. A similarity matrix $S_s(i, j)$ is constructed, in which each entry is taken to be the summed Euclidean distance of the joint locations between the 3D skeletons for frames i and j.

Temporal filtering. The skeleton comparison above only provides the similarity of static skeletons, so no dynamic information is captured. A common method to capture dynamic frame information is through *temporal filtering*. This process has been used before in the context of Video Textures to maintain dynamics when finding match points [Efros et al. 2003; Schödl et al. 2000], and has also been used in the context of 3D shape matching [Huang et al. 2010a]. In this technique, comparisons from neighbouring frames in the static similarity matrix S_s are incorporated into a frame's measure. To achieve this, a convolution is applied to S_s. Entry $S_{w=t}(i, j)$ in a temporal similarity matrix is the mean of cells $S_s(i - t, j - t)$ to $S_s(i + t, j + t)$, where t is the temporal window size.

4 EVALUATION METHODOLOGY

4.1 Synthetic Data

We evaluated our skeleton-based match point detector using a method similar to that employed in the related works most similar to our own [Huang et al. 2010a, 2015; Veinidis et al. 2017]. Specifically, we use ROC curves, evaluating the detector against a ground truth. To allow a ground truth to be available, the technique was evaluated against synthetic data.

To construct the synthetic data set, motion capture data was applied to a rigged avatar. We generated synthetic data using an avatar animated using six motion capture performances freely available on Mixamo.com. As the 3D mesh of the avatar has a known and fixed topology, we establish ground truth similarity by calculating the average difference in position and velocity for each vertex. For this, 3D shapes must first be aligned, which we achieved as for 3D skeletons, using an open source implementation of the Kabsch algorithm. The alignment rotation is only calculated around the "up" vector to ensure actions do not leave the ground plane. In this way, ground truth was calculated in a similar way to [Huang et al. 2010a].

Distances between frames in the ground truth were then normalized into the range [0,1]. This provides a ground truth matrix in which an entry $GT(i, j)$ contains the normalized difference between two frames i and j. This normalized ground truth matrix is then turned into a binary classification matrix by applying a threshold. We use a value of 0.3 as this was the threshold used in [Huang et al. 2010a]. The binary classification matrix BC is defined such that

[1] https://uk.mathworks.com/matlabcentral/fileexchange/37192-intersection-point-of-lines-in-3d-space
[2] https://github.com/charnley/rmsd

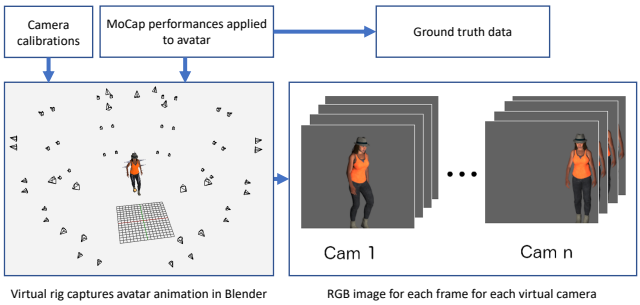

Figure 3: Flowchart showing synthetic data generation from an avatar animation.

$BC(i, j) = 1$ if $GT(i, j) < 0.3$ and 0 otherwise (i.e. $BC(i, j)$ encodes a binary value indicating if frames i and j are considered similar).

To ensure the synthetic data set accurately mirrored the inputs our skeleton-based technique would receive in the real world, the RGB camera layout of the physical volumetric studio cameras was re-created in Blender. In our case, this is ~50 inward-facing RGB cameras in a cylinder-shaped rig. The extrinsic matrix of each physical camera – calculated during the calibration step of a real production shoot – was applied to a virtual camera, exactly recreating the physical dimensions of the rig. The FOV of the virtual cameras were set to be equivalent to the physical cameras, and the avatar was scaled such that it represented an actor who was 172cm tall. Two-dimensional RGB image sequences of the avatar animations were generated for each of these camera views. This provides a dataset comparable to that produced by the volumetric capture rig in a real-world context, as well as ground-truth data. An overview of the process used to create a datatset for a single avatar animation is shown in Figure 3.

4.2 Real data

We also perform an evaluation of our skeleton-based match point detector on real-world data. Due to the fact that the 3D meshes in our FVV data are temporally unstructured, it is not possible to establish a ground truth for this real-world data. As a result, ROC curve analysis is not appropriate. Instead, we present examples and heatmaps to visually show how well our SBMPD method works. Additionally, through real-world data we analyze to what extent our method could be used on-set to derive near-real-time information on the quality of match points in filmed scenes, allowing the re-shooting of takes in which no suitable match points were detected.

5 RESULTS

5.1 Synthetic data

Here, we present the performance of our SBMPD on synthetic data using ROC curves. Six motion capture performances were used to animate an avatar. These animations were "walking forward" (45 frames), "start walking" (71 frames), "stop walking" (72 frames), "jump" (57 frames), "turn 180 degrees" (24 frames) and "walking in a circle" (67 frames).

A temporal ground truth (TGT) was used that captured the shape and dynamics of the mesh at each frame. The TGT was calculated as in [Huang et al. 2010a]. For each pair of frames across all animations, the 3D meshes for those frames were aligned, and the average difference in Euclidean distance and velocity for all vertices was calculated. The velocity of a vertex was calculated based on its location in the following frame. The average vertex Euclidean distance and velocity were each weighted by 0.5 and summed to provide a final TGT value. For further details, see [Huang et al. 2010a]. The self-similarity (i.e. an animation compared against itself) TGT and resulting binary classification matrix for each clip are shown in Figure 4.

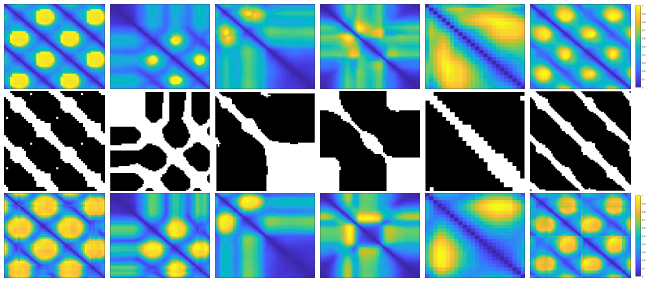

Figure 4: Self-similarity matrices for the six animations (left to right: walk forward, start, stop, jump, turn, walk circle). Top row: temporal ground truth normalized into $[0, 1]$ range. Center row: binary classification matrix i.e. top row thresholded at 0.3. Bottom row: our method i.e. SBMPD value for static frame similarity (temporal window size = 0). In the heatmaps (top and bottom rows) a darker color indicates frames are more similar.

The SBMPD as described in Section 3 was then used to calculate the difference between all frames in self-similarity sequences. The values generated by this method for a static frame (temporal window size = 0) are shown on the bottom row of Figure 4.

To incorporate temporal information, the similarity matrices are convolved as described in the "temporal filtering" paragraph in Section 3. This requires choosing an appropriate temporal window size, with the optimal window size being dependent on the rate of motion and frame rate of the capture [Huang et al. 2010a]. In all of our examples, the frame rate was 30fps. We show a range of temporal window sizes applied to the "walking forward" animation in Figure 5. As can be seen in Figure 5, the heatmaps show diagonal lines in both directions. Diagonal lines from top right to bottom left indicate incorrectly identified match points – these are points in which the static skeletons appear similar, but where the dynamics of the joints are different. An example of this in the walk cycle is when the arms in two frames are swinging in opposite directions, but the static frames appear similar. As shown in Figure 5, in this context a temporal window size of two is best to remove these incorrect matches, without blurring the matrix to the point where correct similarities cannot be identified (i.e. diagonal lines from top left to bottom right are preserved). As a result, we conclude that a window size of two is appropriate for the rate of movement in our synthetic data, and use this window size in the following analysis.

In Figure 6 an ROC curve shows the performance of our SBMPD, incorporating temporal information using a temporal window size of two and comparing against the TGT.

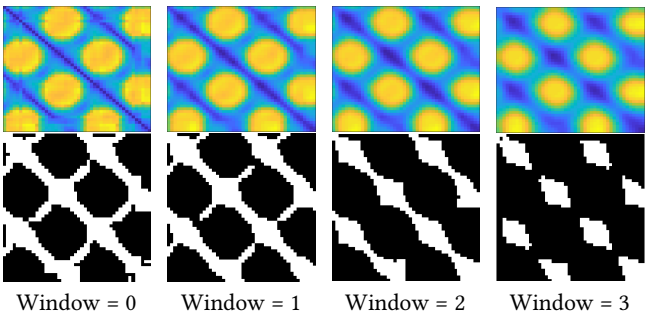

Figure 5: Above: heatmaps for normalized SBMPD values for self-similarity of "walking forward", convolved over different temporal window sizes. Darker colors indicate frames are more similar. The differences between the heatmaps are subtle, so below we show these heatmaps thresholded with a value of 0.4. This value of 0.4 was chosen by trial and error to graphically show the effect of the window size.

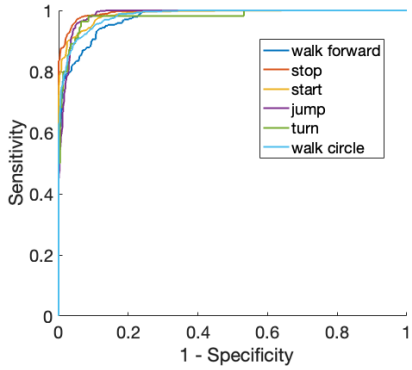

Figure 6: Self-similarity SBMPD performance as ROC curve against temporal ground truth (temporal window size = 2).

Discriminator accuracy can be reported statistically using the area under the curve (AUC), where the area under an ROC curve is given as a number. An AUC of one indicates ideal discrimination for a given dataset. In Figures 4 and 6 we have presented only self-similarity. It is also possible to calculate all pairwise comparisons between animations. Our SBMPD achieved an average AUC of 0.988 for self-similarity comparisons and 0.972 across all pairwise comparisons. A complete breakdown of each comparison result is shown in Table 1.

5.2 Real data

We tested our SBMPD on real data. Eleven shots were filmed. These were two "walking on treadmill" clips (250 and 362 frames), three "start walking" clips (47, 54 and 52 frames), two "stop walking" clips (87 and 63 frames), two "walk arc right" clips (30 and 36 frames), and two "walk arc left" clips (40 and 37 frames). Although we have a 3D processed mesh for all frames, unlike with the synthetic data above, the topology of the mesh is not fixed. As a result, it is not possible to create a ground truth. Instead of performing a statistical analysis,

Table 1: Area under ROC curves for all pairwise animation comparisons (temporal window size = 2). As the table would be symmetric, repeated values are not shown. Blank entries indicate an ROC curve could not be constructed due to a lack of acceptable transition frames in the temporal ground truth.

Animation	Walk forward	Start	Stop	Jump	Turn	Walk circle
Walk forward	.980					
Start	.970	.985				
Stop	.975	.978	.993			
Jump	–	.996	.996	.988		
Turn	.963	.915	.938	.982	.993	
Walk circle	.966	.952	.943	–	.961	.988

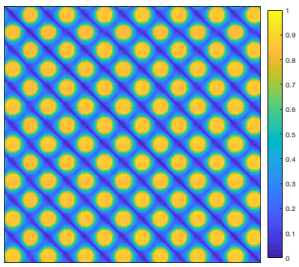

Figure 7: Normalized SBMPD results for static frame self-similarity of the real-world walk cycle "walking on treadmill 1" (250 frames). Darker colors indicate frames are more similar.

in this section we present heatmaps and examples of match points identified by our SBMPD.

As indicated by the names of our clips, we filmed multiple takes for each action. In practice, it would be reasonable to film this many takes in a volumetric capture studio. However, due to the cost of processing, turning each of these shots into a 3D mesh to identify good match points would likely be prohibitively expensive. Ideally, our SBMPD could be used to identify which of the similar shots should be processed to achieve the best end result. To this end, we also present visually the processed meshes for the best matches found by our SBMPD, and compare these against the meshes at match points our SBMPD indicated would be less good.

5.2.1 Using the SBMPD to Identify Match Points. First, we use a heatmap to show static frame self-similarity for a walk cycle in Figure 7. In this figure, the repeated motion caused by the nature of the walk cycle is clearly visible as diagonal lines. This indicates that OpenPose-based 3D skeletons are a viable way to identify shape similarity in data captured from the real world.

To incorporate the dynamics of the model, we convolve the similarity matrix as discussed in Section 3. Using the same technique outlined in Section 5.1, we identify that a temporal window size of two is best for the rate of motion in our real data. We use a temporal window size of two in the remainder of our analysis.

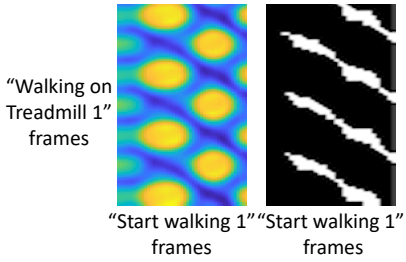

Figure 8: "Start walking 1" against 125 frames of "walking on treadmill 1" (temporal window = 2). Left: normalized heatmap of SBMPD score between frames. Right: heatmap thresholded at 0.3, with this value chosen by trial and error.

We start with an example of creating a motion sequence. In this example, we want the character to begin in a standing position before walking forwards. For this, we must find a match point to allow a cut into one of the walking clips from a "start walking" clip. Figure 8 shows normalized SBMPD scores for transitions between frames in the "start walking 1" clip and the "walking on treadmill 1" clip. We use only 125 frames from the latter clip to make the images easier to understand.

It is interesting to note that this combination of actions could not have been produced without cutting between two clips – as the physical size of the capture studio is limited, the actor could not start walking and then walk more than two steps at normal speed before leaving the capture volume. Although we film the actor walking on a treadmill, due to the jolt as the treadmill started and stopped, we were unable to film a natural looking "start walking" motion on the treadmill. As such, combining clips would have been necessary to achieve this sequence.

As can be seen in Figure 8, suitable transitions cannot be identified for earlier frames in the "start walking" clip. This is as expected; as the actor was in a neutral standing position before they started walking, there would not be a suitable cut into a walking clip from these frames. Later in the "start walking" clip, similarities in the pose and dynamics of the character between frames are found between the clips, allowing a reasonable cut to be identified. An entire sequence composed of "start", "walk" and "stop" clips is shown in Figure 1.

5.2.2 Using SBMPD Score to Assess Match Point Quality. As discussed in Section 3, the SBMPD "score" is the summed Euclidean distance of each of the 3D skeleton joints, convolved over a temporal window. As such, the score itself depends on whatever unit is used in the camera calibration matrices used to create the 3D skeletons. As a result, SBMPD scores are only comparable within the settings of a single calibrated rig. For reference, our SBMPD scores range from approximately 20 to 250, with a lower score indicating a better match.

Here, we consider the issue of deciding which clips to process into 3D meshes. By finding the match points with the lowest SBMPD scores, we hope that we can find a good join without needing to process all of the clips. To demonstrate this, we again consider the example of cutting from a "start walking" clip into one of any of the walking clips.

Table 2: Lowest SBMPD values for all "start walking" clips to all "walking" clips. The lowest and highest of these scores are highlighted in bold. Note that the below scores represent the best possible match point between each pair of clips, as identified by our SBMPD.

Clip name	Start 1	Start 2	Start 3
Walking on treadmill 1	45	39	36
Walking on treadmill 2	41	36	**34**
Walk arc right 1	61	58	55
Walk arc right 2	62	57	58
Walk arc left 1	53	54	54
Walk arc left 2	65	**67**	66

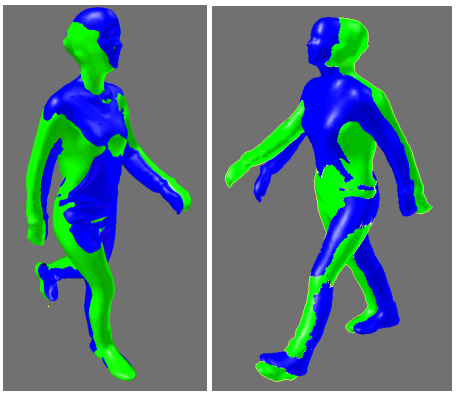

Figure 9: Left: best match point with a SBMPD score of 34. Right: "worst of the best" match point with a SBMPD score of 67. It is clear visually that the meshes on the left are more similar.

We calculate the SBMPD score for all of these possible pairs. Each of these pairwise results are shown in Table 2. Based on these SBMPD values, we identified that the best cut (i.e. match point with the lowest SBMPD score) is between "Start walking 3" and "Walking on treadmill 2". We also identified that the "worst of the best" match points is between "Start walking 2" and "Walk arc left 2". By this, we mean that the best match point between these two clips is worse than the best match point between all other pairs of clips.

To tell if our SBMPD had correctly identified how "good" these match points are, we examined the processed meshes. Note that in a real-world context, these meshes would not be available at the comparison stage, as they would not have been processed yet. By looking at the processed meshes as shown in Figure 9, it was clear visually that the meshes for the match point with the lower SBMPD score were more similar. As these two match points represent the best identified frames to cut between their respective clips, the SBMPD had correctly identified the better match. In practice, this would allow a decision to be made about which clips to process based on the 3D skeletons derived from the RGB camera feeds.

This indicates that decisions can be made about which clips to process into 3D meshes using the SBMPD score. It is interesting to note that in the "start, walk, stop" motion sequence shown in Figure 1, there was no great transition into a "stop walking" clip available. The best identified transition had an SBMPD score of 45. In all of the "stop walking" clips, our actor was leaning very slightly backwards, while in the "walking on treadmill" clips they were leaning slightly forwards. This is difficult to correct for, as we only rotate around the "up" axis to ensure animations remain on the ground plane. If we had had access to the 3D skeletons on set, we could have provided this note to our actor and re-filmed the motion, allowing a better transition to be available.

6 USING THE SBMPD SYSTEM ON SET

Ideally our SBMPD would work in near real-time. If match point quality could be assessed on set, clips with poor quality match points could be re-filmed immediately, potentially saving time and money if more studio time was required to re-film content at a later date. Near real-time 3D skeleton creation also has other useful on-set applications, such as driving rigged avatars to allow the immersive playback of a clip with 6DoF.

Comparisons between 3D skeletons, and generating the 3D skeletons from the 2D skeleton data, are trivial due to the reduced pose feature vector representation of 25 joint locations per frame. The longest running task is the processing of the RGB camera feeds into 2D skeletons by OpenPose. OpenPose is accelerated using the GPU. The input image from each RGB camera is 2048x2048 pixels. On a PC with an Intel i7-4790@3.6GHz CPU and an NVIDIA GeForce GTX 980 Ti graphics card, these images are processed by OpenPose into 2D skeletons at around 8fps. There are around 50 RGB cameras ($n \approx 50$) in the volumetric capture rig. Using this single PC, the RGB feeds could be processed at $8/n \approx 0.16$fps.

This is likely to be too slow for on-set usage. The OpenPose processing can be parallelized, however. As volumetric capture studios are likely to have large numbers of powerful GPUs available in the processing farm, it may be practical to have one GPU dedicated to each RGB camera feed. It may also be possible to reduce the processing complexity by using a subset of the RGB cameras. In future work, we intend to explore the impact that the number and spread of RGB cameras employed has on 3D skeleton and match point identification accuracy. Additionally, we intend to further explore the opportunities of using 3D skeletons on set.

7 COMPARISON AGAINST THE STATE-OF-THE-ART

The state-of-the-art in mesh comparisons produces excellent results in the context of identifying match points between FVV clips. However, the cost of processing all of the meshes before comparison is likely prohibitively expensive in all but the highest-budget productions. Our SBMPD is novel in that it attempts to find match points before mesh processing, allowing good match points to be identified at a fraction of the cost of state-of-the-art systems.

7.1 Accuracy

As our skeleton comparisons happen before 3D mesh processing, they do not have access to the 3D mesh during match point identification. As a result, it was anticipated that our SBMPD system would not reach the same levels of accuracy as mesh-based approaches, as these techniques have more information available to make comparisons. Despite this, our method performs well, most likely because 3D skeletons are an excellent descriptor of human pose. Techniques that result in temporally consistent meshes could be considered to be the state-of-the-art in regards to identifying match points between clips [Ahmed et al. 2008; Budd et al. 2013; Cagniart et al. 2010; Carranza et al. 2003; Huang et al. 2011; Loper et al. 2015; Mustafa et al. 2016; Tung and Matsuyama 2010]. While these techniques suffer from other limitations, as discussed in Section 2.1, comparisons between frames can be performed easily using the distance between corresponding vertices [Casas et al. 2012a,b]. As this is similar to how the ground truth was calculated for our synthetic data in Section 5.2, temporally consistent meshes may be thought of as producing ground-truth-level identification of match points. This means methods employing temporally consistent meshes could be considered to produce discrimination with an AUC of 1, while our AUC across all shots was 0.972. In this case, our SBMPD performed with 2.8% poorer accuracy than the state-of-the-art for the synthetic dataset. This small performance drop, however, comes with large improvements in processing time and cost requirements.

7.2 Cost and time

The largest cost associated with a FVV production is likely to be the mesh processing costs. The cost of mesh processing with current state-of-the-art comparison techniques increase proportionally with the amount of content being compared, while mesh processing costs for our system are instead tied only to the duration of content required for the final output. We take as an example the start-walk-stop sequence created in Section 5.2, which was created from 11 clips totalling around 45 seconds. The commercial rate to process one minute of high-quality FVV content at the time of writing is around £6000[3]. To process all of these meshes for comparison would therefore cost £4500. Using our SBMPD to identify match points before mesh processing would have resulted in only processing exactly the frames required for the final seven second output, at a cost of around £700. We must also account for the processing incurred by our SBMPD. As outlined in Section 6, running OpenPose on the camera feeds is the processing dominant task. Given 50 render nodes in the farm, each processing a camera stream, we would expect OpenPose working at 8fps to process the 45s of footage (1350 frames, given 30fps) in approximately 2.8 minutes. Given the cost of processing on the render farm is £10/min[4], this equates to a cost of around £28. This means our SBMPD would cost £728, compared to £4500 for state-of-the-art methods. This represents a 6.2-fold reduction in cost for this example. It is important to note, however, that the cost saving can be much larger, depending on the duration of content captured and required for the final output.

While the time taken to process meshes varies depending on the speed and number of nodes in the render farm, a recent estimate

[3]Personal communication with Dimension Studio, 2020
[4]Personal communication with Dimension Studio, 2020

from a commercial FVV studio required one hour of processing per six seconds of content[5]. Using the same example from Section 5.2, this represents a wait of 7.5 hours using state-of-the-art mesh comparisons, against our SBMPD that would require just over an hour to processes the meshes for the final output. Additionally, it may be possible for our SBMPD to provide near-real-time feedback on if the correct content had been captured, as discussed in Section 6, which is not possible with mesh-based comparison techniques.

8 LIMITATIONS AND FUTURE WORK

By applying our SBMPD system to data captured in a volumetric studio, we have shown that our method works in a real-world context. It will require more data from a range of motions, actors, clothing and scenarios, however, before it can be claimed that our approach works well in practice. As our shape descriptor does not capture secondary motions, shoots involving loose hair and clothing may prove problematic for our method. We leave these investigations as future work.

As discussed in Section 2.2, creating loops and sequences of FVV clips has two components: identifying good match points, and blending the clips together at these points. This work focuses on the former, and we have not attempted to perform mesh blending on our results. This can make it hard to assess visually how our method compares to others. In future work, we intend to perform mesh blending to allow comparisons to be made more easily. We also do not perform time warping [Kovar and Gleicher 2003], which has been applied in the area of FVV to improve blends [Boukhayma and Boyer 2018]. Additionally, we have not created Motion Graphs automatically using our identified transitions. While such an investigation would be interesting, here we instead explored how our SBMPD could be used manually in FVV production workflows. As indicated in Section 2.2, however, Motion Graph construction requires identifying match points between clips. As such, state-of-the-art techniques entail wasted production effort when creating Motion Graphs that we have shown our system would significantly reduce, making such techniques usable in practice.

In our synthetic data analysis presented in Section 5.1, our ground truth data is created from a rigged articulated avatar. Therefore our synthetic animations do not exhibit secondary motions, such as cloth dynamics, which would be present in real data. Using a different dataset that simulated hair and cloth dynamics might improve ecological validity.

We have compared our results against the state-of-the-art in match-point identification, currently considered to be temporally consistent meshes [Casas et al. 2012a,b]. These techniques, however, suffer from other issues such as artifacts during large or rapid shape changes, as discussed in Section 2.1. A comparison against other techniques, such as Shape Histograms [Huang et al. 2010a], would be useful to further illuminate the trade-offs between cost, accuracy and final output quality. We intend to do this in future, either with help from the authors or by re-implementing their work.

Our SBMPD system achieves high ROC accuracy against the ground truth, and performs well on real-world data based on visual inspection. Due to the highly reduced feature representation used – only 25 joint positions of the 3D skeleton are stored per frame – it is unlikely that our SBMPD system will ever be as accurate as comparisons performed on the processed mesh. One possible approach to improve the results would be to take a hybrid approach, where the SBMPD is used to identify a small number of candidate match points, which are then processed into 3D meshes for more accurate comparison. OpenPose can also be used to identify finger positions and facial expressions, which could be incorporated into SBMPD comparisons. Additionally, it may be possible to use information from the 3D skeletons to allow comparisons using 2D image metrics, such as comparing between interpolated RGB views. We leave these investigations as future work.

Our mechanism to turn 2D skeletons into 3D skeletons is very simple. More robust 3D skeleton tracking could be employed. This could involve making more use of the 2D joint identification confidence values to weight their contribution to the 3D joint positions, or by performing temporal filtering. Both of these have previously been investigated in the context of 3D skeleton tracking from multi-view RGB cameras [Ohashi et al. 2018; Schwarcz and Pollard 2018].

A major advantage of our system is the time saved against state-of-the-art techniques and the potential of deploying a near-real-time version on set. A qualitative evaluation of how these improvements impact the content production pipeline would improve our understanding of how these techniques can be deployed in practice.

9 CONCLUSION

In this work, we considered the use of 3D skeletons to improve the FVV production pipeline. FVV is an increasingly common way to produce immersive content. Despite this, it has a number of limitations, including cost, processing time, and a lack of interactivity caused by the content being fixed at the point of filming. To reduce the amount of content that needs to be processed, it may be possible to reuse content through motion loops. To improve interactivity, previous works have explored how to create motion sequences from clips of FVV content. These works, however, have identified points to cut between clips by comparing the 3D meshes. As the main cost associated with FVV production is in creating these 3D meshes, we have argued that performing these comparisons earlier in the production pipeline could reduce cost and processing time. To this end, we evaluated the use of 3D skeletons derived from the 2D camera views using OpenPose to identify cut points between FVV clips.

We analyzed the performance of our method on synthetic data using ROC curves. Our method produced good results, achieving an average AUC of 0.988 for self-similarity comparisons and 0.972 across all pairwise comparisons. We also demonstrated through examples that our technique works well on real-world data. Although our SBMPD performed with 2.8% poorer accuracy than the state-of-the-art for our synthetic dataset, cost and processing time requirements increase with the duration of the final output rather than the compared content. In a real-world example, this translated to a 6.2-fold improvement. Despite the fact that 3D skeletons do not capture secondary motions such as clothing and loose hair, we believe our results indicate that this method has great potential to improve the FVV production pipeline in practice.

[5]Personal communication with Dimension Studio, 2020

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, and Sarah-Louise Young whose likeness appears in the figures of this work.

REFERENCES

Naveed Ahmed, Christian Theobalt, Christian Rossl, Sebastian Thrun, and Hans-Peter Seidel. 2008. Dense correspondence finding for parametrization-free animation reconstruction from video. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 1999. 3D shape histograms for similarity search and classification in spatial databases. In *International Symposium on Spatial Databases*. Springer, 207–226.

Okan Arikan and David A Forsyth. 2002. Interactive motion generation from examples. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 483–490.

Morten Bojsen-Hansen, Hao Li, and Chris Wojtan. 2012. Tracking surfaces with evolving topology. *ACM Trans. Graph.* 31, 4 (2012), 53–1.

Adnane Boukhayma and Edmond Boyer. 2018. Surface Motion Capture Animation Synthesis. *IEEE transactions on visualization and computer graphics* 25, 6 (2018), 2270–2283.

Christoph Bregler, Michele Covell, and Malcolm Slaney. 1997. Video Rewrite: driving visual speech with audio.. In *Siggraph*, Vol. 97. 353–360.

Chris Budd, Peng Huang, Martin Klaudiny, and Adrian Hilton. 2013. Global non-rigid alignment of surface sequences. *International Journal of Computer Vision* 102, 1-3 (2013), 256–270.

Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. 2010. Free-form mesh tracking: a patch-based approach. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1339–1346.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.

Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. 2003. *Free-viewpoint video of human actors*. Vol. 22. ACM.

Dan Casas, Margara Tejera, Jean-Yves Guillemaut, and Adrian Hilton. 2012a. 4D parametric motion graphs for interactive animation. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM, 103–110.

Dan Casas, Margara Tejera, Jean-Yves Guillemaut, and Adrian Hilton. 2012b. Interactive animation of 4D performance capture. *IEEE transactions on visualization and computer graphics* 19, 5 (2012), 762–773.

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 69.

Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. *Performance capture from sparse multi-view video*. Vol. 27. ACM.

Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. 2003. Recognizing action at a distance. In *null*. IEEE, 726.

Matthew Flagg, Atsushi Nakazawa, Qiushuang Zhang, Sing Bing Kang, Young Kee Ryu, Irfan Essa, and James M Rehg. 2009. Human video textures. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*. ACM, 199–206.

Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. 2004. Modeling by example. In *ACM transactions on graphics (TOG)*, Vol. 23. ACM, 652–663.

Rachel Heck and Michael Gleicher. 2007. Parametric motion graphs. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*. ACM, 129–136.

Peng Huang, Chris Budd, and Adrian Hilton. 2011. Global temporal registration of multiple non-rigid surface sequences. In *CVPR 2011*. IEEE, 3473–3480.

Peng Huang, Adrian Hilton, and Jonathan Starck. 2009. Human motion synthesis from 3d video. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1478–1485.

Peng Huang, Adrian Hilton, and Jonathan Starck. 2010a. Shape similarity for 3D video sequences of people. *International Journal of Computer Vision* 89, 2-3 (2010), 362–381.

Peng Huang, Jonathan Starck, and Adrian Hilton. 2007. A study of shape similarity for temporal surface sequences of people. In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*. IEEE, 408–418.

Peng Huang, Margara Tejera, John Collomosse, and Adrian Hilton. 2015. Hybrid skeletal-surface motion graphs for character animation from 4d performance capture. *ACM Transactions on Graphics (ToG)* 34, 2 (2015), 17.

Peng Huang, Tony Tung, Shohei Nobuhara, Adrian Hilton, and Takashi Matsuyama. 2010b. Comparison of skeleton and non-skeleton shape descriptors for 3d video. In *Proceedings of the 3DPVT International Symposium*.

Takeo Kanade, Peter Rander, and PJ Narayanan. 1997. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia* 4, 1 (1997), 34–47.

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. 2003. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, Vol. 6. 156–164.

Lucas Kovar and Michael Gleicher. 2003. Flexible automatic motion blending with registration curves. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 214–224.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2008. Motion graphs. In *ACM SIGGRAPH 2008 classes*. ACM, 51.

Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. 2002. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (ToG)*, Vol. 21. ACM, 491–500.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 248.

Andrew MacQuarrie and Anthony Steed. 2020. Improving Free-Viewpoint Video Content Production Using RGB-Camera-Based Skeletal Tracking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 775–776.

Armin Mustafa, Hansung Kim, and Adrian Hilton. 2016. 4D match trees for non-rigid surface alignment. In *European Conference on Computer Vision*. Springer, 213–229.

PJ Narayanan, Peter W Rander, and Takeo Kanade. 1998. Constructing virtual worlds using dense stereo. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 3–10.

Takuya Ohashi, Yosuke Ikegami, Kazuki Yamamoto, Wataru Takano, and Yoshihiko Nakamura. 2018. Video motion capture from the part confidence maps of multi-camera images by spatiotemporal filtering using the human skeletal model. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4226–4231.

Panagiotis Papadakis, Ioannis Pratikakis, Theoharis Theoharis, and Stavros Perantonis. 2010. PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. *International Journal of Computer Vision* 89, 2-3 (2010), 177–192.

Fabián Prada, Misha Kazhdan, Ming Chuang, Alvaro Collet, and Hugues Hoppe. 2016. Motion graphs for unstructured textured meshes. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 108.

Charles Rose, Michael F Cohen, and Bobby Bodenheimer. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–40.

Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa. 2000. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 489–498.

Steven Schwarcz and Thomas Pollard. 2018. 3D Human Pose Estimation from Deep Multi-View 2D Pose. In *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2326–2331.

Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. 2004. The princeton shape benchmark. In *Proceedings Shape Modeling Applications, 2004*. IEEE, 167–178.

Jonathan Starck and Adrian Hilton. 2003. Model-based multiple view reconstruction of people. In *IEEE international conference on computer vision*. 915–922.

Jonathan Starck and Adrian Hilton. 2007. Surface capture for performance-based animation. *IEEE computer graphics and applications* 27, 3 (2007), 21–31.

Jonathan Starck, Gregor Miller, and Adrian Hilton. 2005. Video-based character animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 49–58.

Luis Molina Tanco and Adrian Hilton. 2000. Realistic synthesis of novel human movements from a database of motion capture examples. In *Proceedings Workshop on Human Motion*. IEEE, 137–142.

Johan WH Tangelder and Remco C Veltkamp. 2004. A survey of content based 3D shape retrieval methods. In *Proceedings Shape Modeling Applications, 2004*. IEEE, 145–156.

Tony Tung and Takashi Matsuyama. 2010. Dynamic surface matching by geodesic mapping for 3d animation transfer. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1402–1409.

Tony Tung and Francis Schmitt. 2005. The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling* 11, 01 (2005), 91–120.

Christos Veinidis, Ioannis Pratikakis, and Theoharis Theoharis. 2017. On the retrieval of 3D mesh sequences of human actions. *Multimedia Tools and Applications* 76, 2 (2017), 2059–2085.

Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. 2008. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 97.

Jianfeng Xu, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2006. Motion editing in 3d video database. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. IEEE, 472–479.