Human Motion Synthesis from 3D Video

Peng Huang, Adrian Hilton, Jonathan Starck
Centre for Vision Speech and Signal Processsing
University of Surrey, Guildford, GU2 7XH, UK
{p.huang,a.hilton,j.starck}@surrey.ac.uk

Abstract

Multiple view 3D video reconstruction of actor performance captures a level-of-detail for body and clothing movement which is time-consuming to produce using existing animation tools. In this paper we present a framework for concatenative synthesis from multiple 3D video sequences according to user constraints on movement, position and timing. Multiple 3D video sequences of an actor performing different movements are automatically constructed into a surface motion graph which represents the possible transitions with similar shape and motion between sequences without unnatural movement artefacts. Shape similarity over an adaptive temporal window is used to identify transitions between 3D video sequences. Novel 3D video sequences are synthesized by finding the optimal path in the surface motion graph between user specified key-frames for control of movement, location and timing. The optimal path which satisfies the user constraints whilst minimizing the total transition cost between 3D video sequences is found using integer linear programming. Results demonstrate that this framework allows flexible production of novel 3D video sequences which preserve the detailed dynamics of the captured movement for actress with loose clothing and long hair without visible artefacts.

1. Introduction

Acquisition and reconstruction of human motion from temporal sequences of people has been a central issue in computer vision over the past decade with advances in the video-based recovery of both skeletal motion and temporal surface sequences which capture both the body, loose clothing and hair movement. The reuse of captured temporal sequences of people (2D video, 3D marker positions, skeletal motion, 3D video surfaces) for animation production is an important problem. Both 2D and 3D video sequences contain detailed information on changes in shape and appearance which is not represented in skeletal motion. There is considerable interest in the use of the surface detail information in animation production as it is prohibitively expensive to reproduce from the underlying skeletal motion. In this paper we introduce a framework for user controlled animation from captured 3D video sequences of people with loose clothing that preserves the non-rigid surface deformation from multiple view video reconstruction whilst allowing constraints on motion, timing and position.

Multiple view reconstruction of human performance as a 3D video has advanced to the stage of capturing detailed non-rigid dynamic surface shape of the body, clothing and hair during motion[1, 24, 19, 22]. Full 3D video scene capture holds the potential to create truly realistic synthetic animated content by reproducing the dynamics of shape and appearance currently missing from marker-based motion capture. However, in 3D video capture the acquisition results in an unstructured volumetric or mesh approximation of the surface shape at each frame without temporal correspondence, estimating correspondence has been the subject of much recent work [2, 1, 24, 23, 18, 17]. Although these techniques could be combined with the surface motion graph to achieve smooth transitions, accurate dense correspondence of dynamic surfaces remains an open problem. It makes the reuse of this kind of data more challenging than conventional motion capture data. In this work performing concatenative synthesis based on 3D shape and motion similarity does not require explicit surface correspondence.

For conventional motion capture, reuse is performed either by learning motion characteristics [13] or example-based methods [21, 12, 11, 4]. Since learning methods risk losing important detail by abstraction they cannot guarantee that synthesized motion is physically realistic and existing systems do not focus on the satisfaction of high-level constraints. Example-based methods which resample the captured data retain the realism and allow a higher-level control [5, 6, 16].

Previous research in example-based concatenative synthesis from motion capture [21, 12, 11, 4] employed a directed graph or Markov chain to represent temporal connections between frames and search for a path satisfying user constraints. These approaches only deal with low degreeof-freedom (DOF) skeletal motion capture data and cannot be directly extended to high DOF 3D video surface motion capture data.

Concatenative animation was introduced in video-based rendering as a means to record and replay the detailed dynamics of a scene from 2D video [16]. Previous work constructed animation transition graphs from 3D video either manually [19] or interactively [26]. Video segments are re-ordered and concatenated at transition points to generate new animated contents. The smoothness of transitions between video segments seriously affects the quality of the final synthesis results. Automatic identification of transitions based on similarity metrics becomes an important problem for high-quality synthesis.

In this paper, we extend example-based methods for motion synthesis from conventional motion capture to 3D video. Temporal 3D video sequence matching [7, 8] is used to automatically identify transitions. A framework is introduced to allow synthesis according to user defined constraints on location, timing and motion key-frames. The system is able to automatically detect transitions, construct a motion graphs and search for the optimal path to satisfy user-defined constraints. The realism and flexibility of the motion synthesis is demonstrated on real data from a public database of 3D video which contains sequences of an actress performing multiple motions with complex non-rigid movement of clothing and hair.

2. Related work

2.1. Motion synthesis

Motion synthesis from conventional motion capture can be categorized into learning and example-based methods. Learning approaches model general motion characteristics and cannot guarantee that the synthesized motion is physically realistic or looks natural. Example-based methods provide an attractive alternative as there is no loss of detail from the original motion dynamics. Current example-based methods that allow high-level constraints on synthesized motion (timing, position etc.) on skeletal motion capture and 2D video are reviewed.

Tanco and Hilton [21] introduced a two-level statistical model for skeletal motion capture to synthesize novel motions: a Markov chain on states (clusters of frames) and a Hidden Markov Model on frames. According to user-defined key-frames, synthesis is performed by searching for an optimal state sequence which minimizes the transition cost between key-frames. This approach does not allow user-defined constraints on position or timing. Similarly, Lee *et al*. [12] provided a two-layer structure allowing efficient search and interactive control from skeletal data. The recursive search terminates when the depth of the spanning tree reaches a given maximum. In both approaches a similarity metric based on the skeletal pose is used to identify transitions over a fixed temporal window. Kovar *et al*. [11] construct a directed graph on skeletal motion capture sequences, referred to as a *motion graph*, where edges correspond to segments of motion and nodes identify connections between segments. Motion segments include original motions and generated transitions. Synthesis is performed by an optimal graph walk that satisfies user-defined constraints. Similarly, Arikan *et al*. [4] employ a direct graph to connect motion segments where each node corresponds to a motion and each edge a transition. A hierarchical randomized search is used to generate motions. Reitsma and Pollard [15] evaluate motion graphs for skeletal motion data and find the capability degrades rapidly with increases in the complexity of environment or tasks. Wang and Bodenheimer [25] evaluate the optimally weighted cost metric for finding transitions through a cross-validation and user study.

Previous approaches on databases of skeletal motion capture exploit the known temporal correspondence for similarity metrics to identify transitions. Skeletal motion capture does not retain the detail of captured surface dynamics. Multiple view reconstruction of 3D video captures the detailed non-rigid surface dynamics as a surface mesh sequence without temporal correspondence. However, current methods cannot be directly extended to 3D video since similarity metrics have considered only skeletal pose. This does not account for surface shape deformation in clothing and hair. The challenge addressed in this work is to identify transitions for 3D video sequences without temporal correspondence and allow user-controlled synthesis to produce novel animations. In previous research on 3D video surface similarity has been defined either manually or through a shape descriptor. Starck *et al*. [20] manually identify transitions to construct a motion graph for interactive control using 3D video sequences to preserve dynamic surface shape and appearance. Xu [26] *et al*. re-use 3D video in a framework of motion editing they compute shape histograms in spherical coordinate system to measure frame-to-frame dissimilarity. In this paper we identify transitions between 3D video sequences using shape similarity over an adaptive window. A framework is introduced to allow concatenative synthesis from 3D video according to high-level user-specified constraints on motion, position and timing. This overcomes limitations of previous example-based approaches to animation from either skeletal motion capture or 3D video data.

2.2. 3D shape matching

In the 3D object retrieval literature, shape descriptors have been widely used to measure similarity. However, these descriptors aim to discriminate between rigid shapes for different object classes (book, mug, chair) and interclass variations (cars, chairs) instead of instances from sequences of the same moving non-rigid object, a person, which differ in both shape and motion. Although a number of researchers have addressed the problem of temporal similarity for skeletal motion in the concatenative motion synthesis literature [21, 12, 11, 4], temporal shape matching for 3D video sequences of people with unknown temporal correspondence has received limited investigation.

Osada *et al.* [14] introduced Shape Distribution that computes the probability distribution of geometric properties of an object as a signature to discriminate similar and dissimilar models. Johnson and Hebert [9] presented a 3D shape-based object recognition system using Spin Images that encode the density of mesh vertices projected onto an object-centred space into a 2D histogram. Ankerst *et al.* [3] provided a 3D Shape Histogram descriptor based on a partitioning of the space where an object resides to classify a molecular database. Kazhdan *et al.* [10] introduced a Spherical Harmonic Representation as a 3D shape descriptor for 3D object retrieval that is constructed by measuring the energy contained in different frequency bands. A comparison of these shape descriptors and their natural extension to temporal matching (via temporal filtering of static similarity) is provided in [8].

3. Surface Motion Graph

The framework comprises two stages: pre-processing the database of the 3D video sequences to construct a *surface motion graph*; and motion synthesis by optimizing the graph-path to satisfy user-defined constraints and minimize the transition cost between 3D video segments. The surface motion graph represents the possible transitions between 3D video sequences which is analogous to motion graphs [11] for skeletal motion capture sequences. Transition points between 3D video sequences are identified without temporal correspondence using a volumetric temporal shape similarity metric.

Surface motion transitions from a 3D video sequence $X = \{x_i\}$ to $Y = \{y_j\}$ are defined as an overlapped subsequence of n frames. Smooth transitions could be generated by linear blending overlapped frames,

$$z_k = (1 - \alpha(t))x_{i+k} + \alpha(t)y_{j+k} \qquad (1)$$

where z_k denote the blended frame of x_{i+k} and y_{j+k}, $k = -\frac{n}{2}...\frac{n}{2}$. Since the process of blending requires surface correspondence which is unknown in this work, Equation 1 is only used as a guide to identify transitions frames $x_i \in X$ and $y_j \in Y$ and adaptive window length n which maximize the similarity between the motion sequences X and Y over the transition. The concatenation then performs as a switch from X to Y at the centre of the window. Although previous work on estimating dense surface correspondence [2, 1, 24, 23, 18, 17] could be combined with surface motion graph to achieve smooth transitions, accurately estimating dense correspondences of dynamic surfaces remains an open problem. Linear blending of overlapped frames to generate a smooth transition will be solved in future work.

Subsequent sections present a metric for temporal shape similarity without surface correspondence and introduce the estimation of transitions according to Equation 1 for constructing the surface motion graph.

3.1. Temporal Shape Similarity

To identify possible transitions between 3D video sequences without temporal correspondence we use a time-filtered volumetric shape histogram to define a similarity metric over a temporal window. The time-filtered volumetric shape histogram has previously been shown to give good performance for human motion recognition on 3D video sequences of people [8]. A shape histogram partitions the space containing an object into disjoint cells corresponding to the bins of a histogram. Given a 3D surface mesh, a volume-sampling spherical shape histogram is constructed as follows:

1. A volumetric representation is constructed by rasterizing the surface into a set of voxels that lie inside the model.
2. Space is transformed to a spherical coordinates system (r, ϕ, θ) around the centre of mass of the model.
3. A 3D histogram is constructed in spherical coordinates, accumulating the voxels in the volume representation with bins size $(\Delta r, \Delta\phi, \Delta\theta)$.
4. The final histogram is normalized by the total number of occupied voxels.

To estimate the orientation about the vertical axis which maximizes the shape similarity, or equivalently minimizes the difference in shape distribution, the spherical histogram similarity is evaluated for all feasible rotations in θ. Instead of rotating the 3D mesh, we generate a high resolution histogram first and shift it with 1° resolution in θ, and re-bin to a coarse histogram. For frame i with no rotation, the high resolution histogram $H^*_{i,0} = h^*_i(r, \phi, \theta)$ and its coarse histogram is $H_{i,0}$; for frame j with α° rotation (α is integer), the high resolution histogram $H^*_{j,\alpha} = h^*_j(r, \phi, f(\theta, \alpha))$, $f(\theta, \alpha) = (\theta + \alpha)\%360^\circ$, and its coarse histogram is $H_{j,\alpha}$. The similarity between frame i and j is computed as the minium Euclidean Distance between $H_{i,0}$ and $H_{j,\alpha}$,

$$s(i, j) = \min_{\alpha \in \{0,...,359\}} \{\parallel H_{i,0} - H_{j,\alpha} \parallel\} \qquad (2)$$

In this paper, we set $(\Delta r, \Delta\phi, \Delta\theta) = (10, 20, 40)$ for the coarse histogram, the optimal bin size reported for human shape similarity [8].

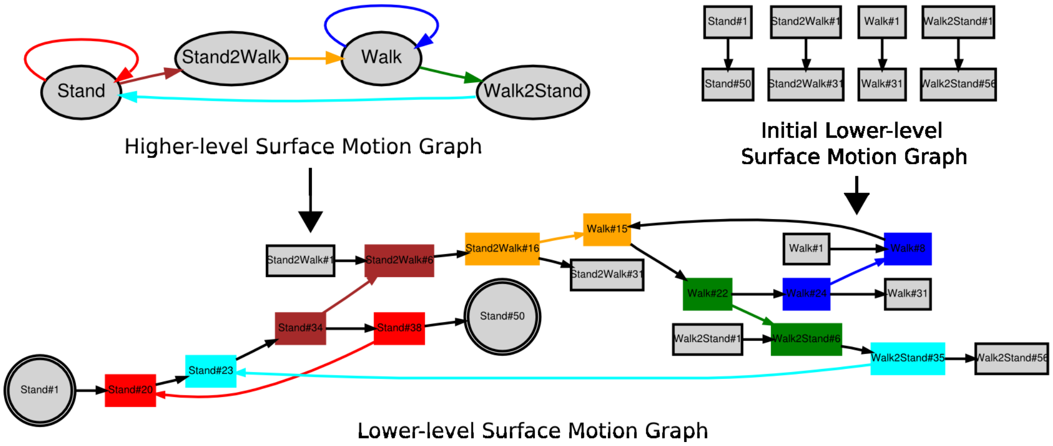


Figure 1. An example of Surface Motion Graph. Double circles denote start and end key-frames. The higher-level graph represents the 3D video sequences and transitions, the lower-level graph represents all possible motion graphs between a particular start and end keyframe.

3.2. Transitions

In this work, we define transitions to maximize the temporal shape similarity according to the linear blend in Equation 1. Given two 3D video sequences $X = \{x_i\}$ and $Y = \{y_j\}$, the (i, j) element of the frame-to-frame shape similarity matrix S_{XY} is defined as

$$S_{XY}(i, j) = s(x_i, y_j) \tag{3}$$

where $s(x_i, y_j)$ is computed by Equation 2. Since the shape similarity is symmetric $s(i, j) = s(j, i)$, the temporal shape similarity is computed as a linearly weighted average of the shape similarity for individual frames about the central frame of the window,

$$S'_{XY}(i, j, n) = \sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}} S_{XY}(i + k, j + k) \cdot w(k) \tag{4}$$

where $w(k)$ is normalised version of $w'(k)$, *i.e.* $w(k) = w'(k)/\sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}} w'(k)$ and $w'(k) = \min\{k + \frac{n+1}{2}, -k + \frac{n+1}{2}\}$. Note that n is the window size and transition length, if n is an even number, the central frame (i, j) is at half way between frames. The transition between X and Y is the global minimum of $S'_{XY}(i, j)$ over all possible i, j, n,

$$(i_{opt}, j_{opt}, n_{opt}) = \arg\min_{i,j,n}\{S'_{XY}(i, j, n)\} \tag{5}$$

Without blending, the concatenation is performed as a switch at the central frame, *i.e.* $x_{\lfloor i_{opt} \rfloor} \rightarrow y_{\lfloor j_{opt} \rfloor + 1}$.

3.3. Automatic Graph Construction

A surface motion graph is a two-level directed graph. In the higher-level, each node represents a motion and each edge a transition with direction. In the lower-level, each node represents a frame and each edge a sequence of frames connecting them. Figure 1 illustrates an example of surface motion graph for four motions. Once the user have defined the higher-level graph and selected key-frames, the lower-level graph will be generated automatically as follows:

1. Initialization: Insert all 3D video sequences as edges and their terminal frames as nodes to create a disconnected graph (Figure 1 upper-right).

2. Identify transitions: If there is an edge between 3D video sequences X and Y in the high-level graph, the transition represented as $(i_{opt}, j_{opt}, n_{opt})$ is evaluated according to Equation 5.

3. Insert transitions and key-frames: The start and end of transitions, key-frames, if they are not in the graph, are inserted as nodes, breaking existing edges (3D video sub-sequences) into smaller ones. Transitions are inserted as edges to join either different 3D videos or different parts of the same sequence.

4. Motion Synthesis

Motion synthesis is performed by optimizing over all possible paths on the lower-level surface motion graph between user selected key-frames to minimize the total transition cost whilst satisfying constraints on position and timing. Note that the optimization considers loops to allow repetition of cyclic motion.

4.1. Cost Function

The cost $C(F)$ of a path F through the surface motion graph is defined as a combination of the total transition cost $C_s(F)$, location cost $C_d(F)$ and time cost $C_t(F)$,

$$C(F) = C_s(F) + w_d \cdot C_d(F) + w_t \cdot C_t(F) \tag{6}$$

where w_d and w_t are weights for distance and time constraints respectively. $w_d = 1/0.3$ and $w_t = 1/10$ are set equal to an error of 30 centimeters in distance with an error of 10 frames in time [4] with a relative weight of 1.0 to emphasise the smoothness cost. Individual cost terms for transitions, distance and time are defined as follows:

Total Transition Cost $C_s(F)$ for a path F is defined as the sum of dissimilarity for all transitions between the input 3D video sequences concatenated to,

$$C_s(F) = s(F) = \sum_{i=1}^{N_t} s_i \qquad (7)$$

where N_t denotes the total number of transitions and s_i the filtered dissimilarity for ith transition .

Distance Cost $C_d(F)$ for a path F with N_f frames is computed as the absolute difference between the user-specified target distance d_T and the total travelled distance $d(F)$,

$$C_d(F) = |d(F) - d_T| \qquad (8)$$

where $d(F) = \sum_{i=1}^{N_f-1} \parallel \mathbf{c}_{i+1} - \mathbf{c}_i \parallel$. $\mathbf{c}_{i+1}$ and $\mathbf{c}_i$ denotes the projection of the centroid of the mesh at frame $i+1$ and i onto the ground respectively along the vertical axis.

Time Cost $C_t(F)$ for a path F with N_f frames is evaluated as the absolute difference between the user-specified target time t_T and the total travelled time $t(F)$,

$$C_t(F) = |t(F) - t_T| \qquad (9)$$

where $t(F) = N_f$. Here, the time is measured in "frames" and the rate of captured 3D video sequences is $25fps$.

4.2. Path Optimization

In this section, we present an efficient approach to search for the optimal path that best satisfies the user defined constraints. The optimal path F_{opt} is found to minimize the combined cost $C(F)$ defined by Equation 6,

$$F_{opt} = \arg\min_{F}\{C(F)\} \qquad (10)$$

Enumerating all possible paths from the start to end key-frame and evaluating the combined cost will give the global optima. But when cycles appear in the graph, the number of paths may become infinite. To avoid this, in stead of enumerating all paths, we enumerate all walks (paths without any loop) from the start to end key-frame and all loops attached to each walk. The global optimal path must be one of compositions of a walk and attached loops. Given $\mathbf{l} = \{l_i\}$ and $\mathbf{n} = \{n_i\}$, $i = 0, \dots, N_l$, let l_0 denote a walk, $n_0 = 1$, $\{l_1, \dots, l_{N_l}\}$ attached loops and $\{n_1, \dots, n_{N_l}\}$ corresponding repetitions, a path F can be represented as $\mathbf{n} \cdot \mathbf{l}$ and the optimization becomes,

$$F_{opt} = \mathbf{n_{opt}} \cdot \mathbf{l_{opt}} = \arg\min_{\mathbf{n},\mathbf{l}}\{C(\mathbf{n} \cdot \mathbf{l})\} \qquad (11)$$

The transition, distance and time costs become,

$$C_d(F) = |\mathbf{n} \cdot d(\mathbf{l}) - d_T| \qquad (12)$$

$$C_t(F) = |\mathbf{n} \cdot t(\mathbf{l}) - t_T| \qquad (13)$$

$$C_s(F) = \mathbf{n} \cdot s(\mathbf{l}) \qquad (14)$$

where $f(\mathbf{l}) = \{f(l_0), \dots, f(l_{N_l})\}$, $f = d$ or t or s. Once the surface motion graph is constructed, there may be more than one walks from the start to end key-frames, let be N_k walks, for each walk $l_{k,0}$, the corresponding set $\mathbf{l_k} = \{l_{k,0}, \dots, l_{k,N_{k,l}}\}$ is determined, the objective is to find an optimal set of repetitions $\mathbf{n_{k,opt}}$ according to Equation 11,

$$\mathbf{n_{k,opt}} = \arg\min_{\mathbf{n_k}}\{C(\mathbf{n_k} \cdot \mathbf{l_k})\} \qquad (15)$$

and the index of global optimal walk and repetitions k_{opt} will be,

$$k_{opt} = \arg\min_{k=1,\dots,N_k}\{C(\mathbf{n_{k,opt}} \cdot \mathbf{l_k})\} \qquad (16)$$

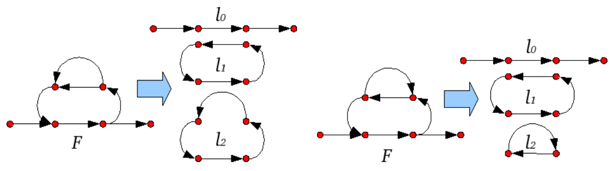
finally, the optimal path F_{opt} is found as a composition of optimal walk and loops $\mathbf{l_{opt}} = \mathbf{l_{k_{opt}}}$ together with optimal repetitions $\mathbf{n_{opt}} = \mathbf{n_{k_{opt},opt}}$,

$$F_{opt} = \mathbf{n_{k_{opt},opt}} \cdot \mathbf{l_{k_{opt}}} \qquad (17)$$

The decomposition of an arbitary path to a walk and attached loops is described in Section 4.2.1 and the composition of a walk and attachted loops back to a path in Section 4.2.3. The optimization of repetitions for a given walk and loops according to Equation 15 are solved as Integer Linear Programming (ILP) problems in Section 4.2.2.

4.2.1 Graph Path Decomposition

Depth-First Search (DFS) is performed to decompose all possible paths between user-selected start and end key-frames to walks and loops . Given an adjacency matrix adj for the graph, source and sink (start and end key-frames), the algorithm is implemented recursively and its pseudo code is presented in Algorithm 1. This gives a set of walks and each of them associated with a set of loops. Note that although loops may be nested such that one loop included another, in our optimization all possible nested sub-loops are included as illustrated in Figure 2.



(a) (b)

Figure 2. Examples of Graph Path Decomposition that consider nested sub-loops.

4.2.2 Integer Linear Programming

Optimization of Equation 15 is non-linear. It can be converted to constrained Integer Linear Programming (ILP) sub-problems. Given a walk and attached loops $\mathbf{l} = \{l_i\}$, l_0 denotes the walk, $n_0 = 1$, the corresponding repetitions $\mathbf{n} = \{n_i\}$ are optimized as four independent ILP sub-problems, $p = 1, 2, 3, 4$, as follows,

$$\begin{aligned} &minimize & \mathbf{C_p} \cdot \mathbf{n_p} \quad &(18a)\\ &subject\ to & n_{p,0} = 1 \quad &(18b)\\ && 0 \leq n_i \leq +\infty,\ integer \quad &(18c)\\ && \mathbf{C_{d,p}} \cdot \mathbf{n_p} \geq 0 \quad &(18d)\\ && \mathbf{C_{t,p}} \cdot \mathbf{n_p} \geq 0 \quad &(18e) \end{aligned}$$

$\mathbf{C_{d,p}}$, $\mathbf{C_{t,p}}$, $\mathbf{C_{s,p}}$ and $\mathbf{C_p}$ are 1D vectors, their elements are computed as follows,

$$\begin{aligned} C_{d,p,i} &= sign_d(p) \cdot w_d \cdot (d(l_i) - d_T) \quad &(19a)\\ C_{t,p,i} &= sign_t(p) \cdot w_t \cdot (t(l_i) - t_T) \quad &(19b)\\ C_{s,p,i} &= C_s(l_i) \quad &(19c)\\ C_{p,i} &= C_{d,p,i} + C_{t,p,i} + C_{s,p,i} \quad &(19d) \end{aligned}$$

where $sign_d = (1, 1, -1, -1)$ and $sign_t = (1, -1, 1, -1)$. For each sub-problem $\mathbf{n_{p,opt}}$ is solved efficiently using a standard ILP solver. The optimal repeat times of loops for a given $\mathbf{l}$ is then computed as the one that achieves the minimun combined cost,

$$\mathbf{n_{opt}} = \arg\min_{p=1,2,3,4} \{\mathbf{C_p} \cdot \mathbf{n_{p,opt}}\} \quad (20)$$

4.2.3 Graph Path Composition

Once the optimal walk and loops $\mathbf{l_{opt}}$ with repetitions $\mathbf{n_{opt}}$ have been evaluated, a complete path can be composed by $F = \mathbf{n_{opt}} \cdot \mathbf{l_{opt}}$. The final motion sub-sequences are concatenated head-to-tail by matching the centroid of mesh at transitions. However, some loops may indirectly connect to the walk via other loops, *e.g.* l_2 via l_1 connecting to l_0 in Figure 2(b), if the repetition of via-loops are zero, we cannot make the path. The feedback strategy is presented in Algorithm 2, when the isolation of loops happens, a constraint is added to the ILP solver.

5. Results and Evaluation

3D character animations are synthesized from a public available database of 3D video [19] which comprises an actress (Roxanne) wearing three different costumes: a game character with shorts and t-shirt (Character1); a long flowing dress (Fashion1); and a shorter tight fitting dress (Fashion2). For each costume periodic (walk,run,stagger), motion transitions (walk to run, run to walk, walk to stand, stand to walk) and other motions (hit, twirl) are included. The captured 3D video sequences are unstructured meshes with unknown temporal correspondence and different mesh connectivity at each time-frame. A mesh contains approximately 100k vertices and 200k triangles.

Surface motion graphs are automatically constructed from 3D video sequences for the performer within three different constumes. Optimization is done in seconds for a user-defined constraints on distance and time. Motions are concatenated without post-processing or blending to show the raw 3D video transitions. Synthesis results are presented in accompanying videos. An example of selected frames from a synthesized motion for Fashion1 captured in a virtual camera view is shown in Figure 3. These results demonstrate that the motion synthesis preserves the detailed clothing and hair dynamics in the captured 3D video sequences and does not produce unnatural movements at transitions.

Motion synthesis is evaluated for the three surface motion graphs which represents potential transitions with four pairs of key-frames for each costume as shown in Table 1. Evaluation is performed by synthesizing motions for target constraints on distance of $1-20m$ in $1m$ intervals and times of $1-40s$ in $1s$ intervals giving 800 sequences for each key-frame pair and 9600 synthesized sequences in total. The maximum, minimum and root mean square errors over all synthesized sequences for distance moved and timing are presented in Table 1 for the sequences generated from each key-frame pair. This analysis shows that the maximum distance and timing errors are less-than 1% of the target indicating that the path optimization generates sequences which accurately satisfy the user-defined constraints. Smoothness cost is evaluated as a weighted average of Hausdorff Distance for overlapped individual frames at transitions. The weights are set to decrease about the central frame the same way with $w(k)$ in Equation 4. Computation times are given for an ILP solver together with a Matlab implementation of the synthesis framework running on a single processor machine. The computation time is approximately constant with respect to the distance and timing constraints as indicated by the low standard deviation.

6. Conclusion

In this paper, we presented a framework for concatenative human motion synthesis from 3D video sequences according to user-defined constraints on movements, position and timing. Transistions between 3D video sequences are identified without the requirement for temporal correspondence using 3D shape similarity over an adaptive temporal window. A surface motion graph is automatically constructed to represent potential transitions for both cross-transitions between different motion sequences and self-transitions in the cyclic motion. Path optimization is performed between user-specified key-frames using standard

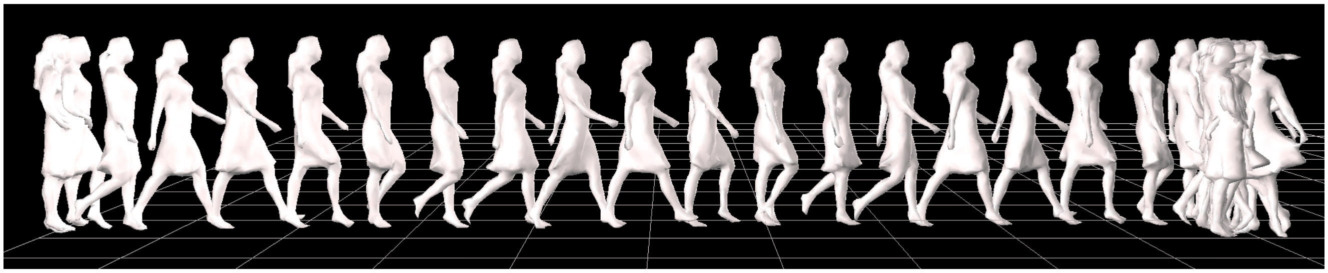


Figure 3. Example meshes for synthesized motion from 3D video database of Roxanne. Fashion1: Pose#1 → Twirl#85. Target traversing distance and time is 10 metres and 500 frames.

SMG: Key-frames	Smoothness (cm)		Distance error (m)			Time error (frame)			Cputime (sec.)
	min	*max*	*min*	*rms*	*max*	*min*	*rms*	*max*	$mean \pm dev.$
Character1: Stand#1 → Hit#45	4.24	19.79	0.0001	0.17	0.96	0	0.16	22	14.43 ± 7.05
Stand#1 → Walk#16	4.24	26.80	0.0002	0.21	0.92	0	4.28	23	12.21 ± 4.74
Walk#16 → Jog#13	4.24	26.80	0.0001	0.21	0.96	0	4.81	24	13.54 ± 4.98
Jog#13 → Hit#45	4.24	25.76	0.0002	0.20	0.98	0	3.88	24	12.20 ± 3.42
Fashion1: Pose#1 → Twirl#85	5.82	20.54	0.0001	0.23	0.95	0	3.76	24	12.79 ± 2.92
Pose#1 → Walk#15	5.82	20.54	0.0001	0.47	0.99	0	4.31	24	5.65 ± 1.46
Walk#15 → WalkPose#37	4.23	13.09	0.0000	0.33	1.00	0	5.85	25	12.01 ± 4.85
WalkPose#37 → Twirl#85	4.23	20.54	0.0001	0.29	1.00	0	4.77	25	10.10 ± 3.07
Fashion2: Pose#1 → Twirl#100	4.49	21.20	0.0000	0.23	0.95	0	7.74	25	12.87 ± 3.61
Pose#1 → Walk#15	4.87	18.12	0.0008	0.39	0.98	0	8.78	25	7.09 ± 1.91
Walk#15 → WalkPose#37	4.49	16.05	0.0002	0.36	1.00	0	8.24	25	14.95 ± 6.11
WalkPose#37 → Twirl#100	4.49	18.12	0.0002	0.28	1.00	0	8.16	24	10.51 ± 2.77

Table 1. Evaluation for Roxanne. A grid of target 20×40 (metres$\times$seconds) is tested for each pair of key-frames shown in the first column.

ILP solver to satisfy constraints on distance and timing with repeated motions for loops in the graph. Results demonstrate that concatenative synthesis of novel sequences accurately satisfy the user constraints and produce motions which preserve the detailed non-rigid dynamics of clothing and loose hair. Linear blending of meshes to produce smooth transitions which requires accurate dense correspondences of dynamic surfaces will be solved in future work. This approach greatly increases the flexibility in the reuse of 3D video sequences allowing specification of high-level user constraints to produce novel complex 3D video sequences of motion.

Acknowledgements

This work was supported by EPSRC grant EP/E001351.

References

[1] E. Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[2] N. Ahmed, C. Theobalt, C. Rossl, S. Thrun, and H.-P. Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. pages 1–8, June 2008.

[3] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, London, UK, 1999. Springer-Verlag.

[4] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. *ACM Trans. Graph.*, 21(3):483–490, 2002.

[5] C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[6] E. Cosatto and H. Graf. Sample-based synthesis of photo-realistic talking heads. In *CA '98: Proceedings of the Computer Animation*, page 103, Washington, DC, USA, 1998. IEEE Computer Society.

[7] P. Huang, J. Starck, and A. Hilton. A study of shape similarity for temporal surface sequences of people. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 408–418, Washington, DC, USA, 2007. IEEE Computer Society.

[8] P. Huang, J. Starck, and A. Hilton. Temporal 3d shape matching. *The Fourth European Conference on Visual Media Production (CVMP'07)*, pages 1–10, Nov. 2007.

[9] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans Pattern Anal Mach Intell*, 21(5):433–449, 1999.

[10] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, 2003.

[11] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, volume 21, pages 473–482, New York, NY, USA, July 2002. ACM Press.

[12] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, 2002.

[13] Y. Li, T. Wang, and H. Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, volume 21, pages 465–472, New York, NY, USA, July 2002. ACM Press.

[14] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, October 2002.

[15] P. S. A. Reitsma and N. S. Pollard. Evaluating motion graphs for character animation. *ACM Trans. Graph.*, 26(4):18, 2007.

[16] A. Schödl and I. A. Essa. Controlled animation of video sprites. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 121–127, New York, NY, USA, 2002. ACM.

[17] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. *IEEE International Conference on Computer Vision (ICCV)*, pages 1387–1394, 2005.

[18] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[19] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[20] J. Starck, G. Miller, and A. Hilton. Video-based character animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–58, New York, NY, USA, 2005. ACM Press.

[21] L. M. Tanco and A. Hilton. Realistic synthesis of novel human movements from a database of motion capture examples. In *HUMO '00: Proceedings of the Workshop on Human Motion (HUMO'00)*, page 137, Washington, DC, USA, 2000. IEEE Computer Society.

[22] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H. P. Seidel. Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):663–674, 2007.

[23] K. Varanasi, A. Zaharescu, E. Boyer, and R. P. Horaud. Temporal surface tracking using mesh evolution. In *Proceedings of the Tenth European Conference on Computer Vision*, volume Part II of *LNCS*, pages 30–43, Marseille, France, October 2008. Springer-Verlag.

[24] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):1–9, 2008.

[25] J. Wang and B. Bodenheimer. Synthesis and evaluation of linear motion transitions. *ACM Trans. Graph.*, 27(1):1–15, 2008.

[26] J. Xu, T. Yamasaki, and K. Aizawa. Motion editing in 3d video database. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 472–479, Washington, DC, USA, 2006. IEEE Computer Society.

Appendix

```
Find_all_walks(adj,source,sink,walk)
walk ← source;
if source=sink then
    walks = walk;
    return;
else
    walks=[];
    foreach node in source's neighbours do
        if node not in walk then
            new_walks =
            Find_all_walks(adj,node,sink,walk);
            walks ← new_walks;
        else
            loop=[];
            i = index of node in walk;
            loop ← walk(i:end);
            if loop not in loops then
                loops ← loop;
            end
        end
    end
end
```

Algorithm 1: Find all walks and loops by DFS

```
$\mathbf{I} := 1..N_l$;
$\mathbf{n}$ ← Solve ILP;
$k = 1$;
while exist $i \in \mathbf{I}, n_i \geq 1$ and $l_i$ not connected to $l_0$ do
    set a 2D matrix $\mathbf{A}$ according to $\mathbf{n}$:
    foreach $i \in \mathbf{I}$ do
        if $n_i == 1$ then
            $a_{k,i} = 1$;
        else
            $a_{k,i} = 0$;
        end
    end
    add constraints $\mathbf{A}\mathbf{n} \geq 1$ to ILP solver;
    $\mathbf{n}$ ← Solve ILP;
    $k = k + 1$;
end
```

Algorithm 2: Feedback strategy