COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2012; **23**:101–111
Published online 20 March 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1430

SPECIAL ISSUE PAPER

Parametric animation of performance-captured mesh sequences

Dan Casas*, Margara Tejera, Jean-Yves Guillemaut and Adrian Hilton

Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, UK

ABSTRACT

In this paper, we introduce an approach to high-level parameterisation of captured mesh sequences of actor performance for real-time interactive animation control. High-level parametric control is achieved by non-linear blending between multiple mesh sequences exhibiting variation in a particular movement. For example, walking speed is parameterised by blending fast and slow walk sequences. A hybrid non-linear mesh sequence blending approach is introduced to approximate the natural deformation of non-linear interpolation techniques whilst maintaining the real-time performance of linear mesh blending. Quantitative results show that the hybrid approach gives an accurate real-time approximation of offline non-linear deformation. An evaluation of the approach shows good performance not only for entire meshes but also with specific mesh areas. Results are presented for single and multi-dimensional parametric control of walking (speed/direction), jumping (height/distance) and reaching (height) from captured mesh sequences. This approach allows continuous real-time control of high-level parameters such as speed and direction whilst maintaining the natural surface dynamics of captured movement.

KEYWORDS

computer animation; 3D video; performance-based animation; surface motion capture

***Correspondence**

Dan Casas, Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, UK.
E-mail: d.casasguix@surrey.ac.uk

1. INTRODUCTION

Advances in three-dimensional (3D) actor performance capture from multiple-view video [1–3] have achieved detailed reconstruction and rendering of natural surface dynamics as mesh sequences. These approaches allow replay of the captured performance with free-viewpoint rendering for compositing of performance in post-production whilst maintaining photorealism. Captured sequences have subsequently been exploited for retargeting surface motion to other characters [4] and analysis of cloth motion to simulate novel animations through manipulation of skeletal motion and simulation of secondary cloth movement [5].

Animation from 3D performance capture has been achieved by concatenation of segments of multiple captured mesh sequences based on a manually defined transitions [6]. Automatic transition graph construction and path optimisation have been introduced [7], allowing offline key-frame animation. The level of movement control in these approaches is limited to transition between the capture movement sequences. Recent work [8] has exploited skeletal tracking of mesh sequences to allow increased manipulation of the captured movement with a skeletal rig, together with photorealistic rendering by indexing a corresponding video database. In these approaches, reuse of captured mesh sequences is limited to offline animation control.

This paper presents a framework for real-time interactive animation with continuous movement control using mesh sequences of captured actor performance. Two contributions are presented to achieve interactive animation: alignment of multiple mesh sequences of an actor performing different motions into a temporally consistent mesh structure and high-level parameterisation of captured mesh sequences. A shape similarity tree representing the shortest non-rigid mesh deformation path is introduced for alignment of frames from multiple captured sequences. This representation enables robust non-rigid alignment of multiple sequences allowing re-sampling with a consistent mesh structure and vertex correspondence. Techniques are introduced for parameterisation of aligned mesh sequences with intuitive high-level parameters such as speed and direction. Parameterisation is achieved by non-linear interpolation between multiple mesh sequences of related motions. This approach is analogous to previous techniques for parameterisation of skeletal motion capture [9]. A hybrid non-linear mesh sequence interpolation approach is proposed

to achieve real-time performance with accurate approximation of non-linear mesh deformation. Previous research in character animation [10] demonstrated the potential of this approach for interactive real-time animation from actor performance-captured sequences. The proposed approach is evaluated to characterise memory and time requirements for different parameterisations. Results on captured mesh sequences of actor performance demonstrate the potential of this approach for interactive character animation with continuous high-level parametric movement control and natural surface deformation.

2. RELATED WORK

2.1. 3D Video Capture and Reconstruction

Kanade *et al.* [11] pioneered the reconstruction of 3D mesh sequences of human performance for free-viewpoint replay with the Virtualized Reality^{TM} system by using a 5 m dome with 51 cameras. Multiple-view video reconstruction results in an unstructured mesh sequence with an independent mesh at each frame. Advances in performance capture from video have enabled reconstruction of mesh sequences for human performance capturing the detailed deformation of clothing and hair [1–3,12]. These approaches achieve a free-viewpoint rendering quality comparable with the captured video but are limited to performance replay. A critical step for editing and reuse of 3D video data is the temporal alignment of captured mesh sequences to obtain a consistent mesh structure with surface correspondence over time. A number of approaches have been proposed for temporal alignment of mesh sequences based on sequential frame-to-frame surface tracking. These can be categorised into two methodologies: model-based approaches which align a prior model of the surface with successive frames [1,2,13] and surface-tracking or scene flow approaches which do not assume prior knowledge of the surface structure [14–16]. Sequential alignment approaches have three inherent limitations: accumulation of errors in frame-to-frame alignment resulting in drift in correspondence over time; gross-errors for large non-rigid deformations that occur with rapid movements requiring manual correction; and sequential approaches are limited to alignment across single sequences. Recently, non-sequential alignment approaches [17,18] have been introduced to overcome these limitations, allowing the construction of temporally coherent 3D video sequences from multiple-view performance capture database, as used in this work.

2.2. Reuse and Editing of 3D Video Data

The lack of temporal coherence in the mesh sequence has prohibited the development of simple methods for manipulation. Animation from databases of mesh sequences of actor performance has been demonstrated by concatenating segments of captured sequences [6,7], which is analogous to previous example-based approaches to concatenative synthesis used for two-dimensional video [19–21]. Recently, example-based approaches through re-sampling video sequences have been extended to body motion [8,21], allowing offline animation via key-frame or skeletal motion. In [8], model-based skeletal tracking was used to re-sample segments from a database of video sequences based on pose-allowing photorealistic rendering with skeletal control. These approaches preserve the realism of the captured sequences in rendering but are limited to replay segments of the captured motion examples and do not allow the flexibility of conventional animation.

2.3. Reuse and Editing of Skeletal Data

Since the introduction of marker-based technologies for skeletal performance capture to the entertainment industry in the early 1990s, a range of techniques to support editing and reuse have been developed. Space-time editing techniques [22–24] provide powerful tools for interactive motion editing via key-frame manipulation. Brundelin and Williams [25] introduced parametric motion control by interpolating pairs of skeletal motions. Parametric motion synthesis was extended to blending multiple examples to create a parameterised skeletal motion space [9,26–28]. This allows continuous interactive motion control through high-level parameters such as velocity for walking or hand position for reaching. This paper introduces analogous parameterisation for 3D video mesh sequences to allow interactive animation control.

3. PERFORMANCE CAPTURE

Actor performance is captured in a controlled studio environment by using a multiple camera system for synchronised video acquisition [1–3]. Shape reconstruction is performed on a frame-by-frame basis using a multiple-view silhouette and stereo approach building on state-of-the-art graph-cut optimisation techniques [29]. This results in an unstructured mesh sequences with both the vertex connectivity and geometry changing from frame-to-frame.

To construct a character model for animation control, we require a set of temporally aligned mesh sequences for multiple motions with the same mesh structure at each frame across all sequences. Extending previous research in non-sequential alignment [17,18], we introduce a mesh sequence alignment approach to recover the non-rigid surface motion and represent all frames with a consistent structure, as illustrated in Figure 1(b–e). Alignment across multiple unstructured mesh sequences is performed by constructing an intermediate ‘shape similarity tree’. This represents the shortest non-rigid surface motion path required to align each frame. The shape similarity tree allows frames from different mesh sequences to be aligned on the basis of a measure of surface shape and motion similarity. The representation also ensures robust alignment of mesh sequences in the presence of large non-rigid deformations due to fast motion where sequential frame-to-frame

DOI: 10.1002/cav

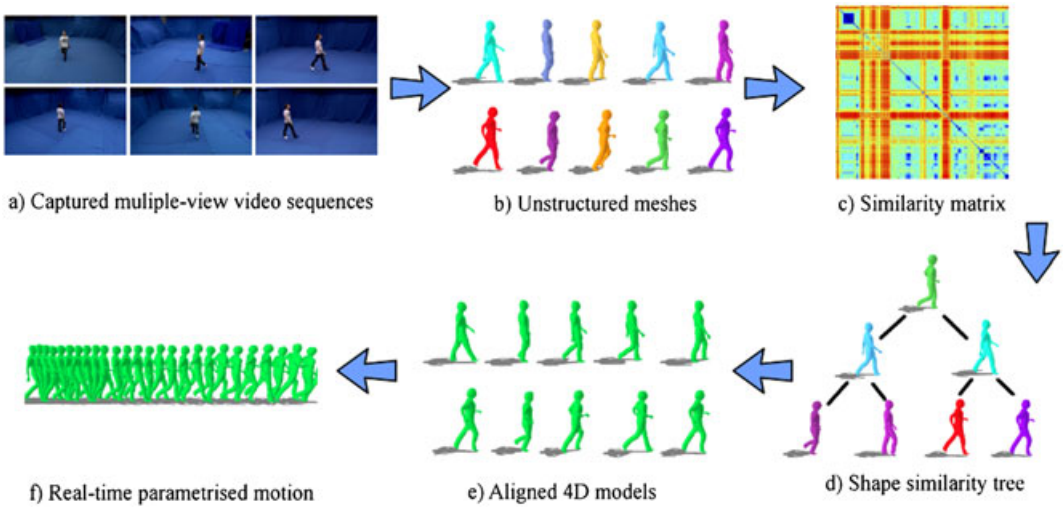


Figure 1. Overview. (a) Captured multiple-view video sequences, (b) unstructured meshes, (c) similarity matrix, (d) shape similarity tree, (e) aligned four-dimensional models, and (f) real-time parameterised motion.

surface tracking approaches may fail. The shape similarity tree is used to recover the non-rigid surface motion and obtain a consistent mesh structure for all sequences.

3.1. Shape Similarity Tree

To construct the shape similarity tree, we require a measure of similarity $s(M_i(t_u), M_j(t_v))$ between pairs of meshes which can be evaluated without prior knowledge of the mesh correspondence. A number of similarity measures for mesh sequences taking into account both shape and motion have been investigated [30,31]. In this work, we utilise the temporally filtered volumetric shape histogram [31] as a measure of shape and non-rigid motion similarity that has been shown to give good performance on reconstructed mesh sequences of people. Evaluation of shape similarity between mesh reconstructions for all frames across all sequences results in a similarity matrix as illustrated in Figure 1(c), where blue indicates high similarity and red indicates low similarity.

Shape similarity is used to construct a tree representing the shortest non-rigid surface motion path required to align all meshes $\{M_i(t_u)\}_{u=1}^{N_i}$ from multiple captured mesh sequences. Initially, a complete graph Ω is constructed with nodes for all meshes $M_i(t_u)$ in all sequences $i = [1, N]$ and edges $e_{iujv} = e(M_i(t_u), M_j(t_v))$ connecting all nodes. Edges e_{iujv} are weighted according to the similarity measure $s(M_i(t_u), M_j(t_v))$. The shape similarity tree T_{sst} minimising the total non-rigid surface motion required for alignment can then be evaluated as the minimum spanning tree of the complete graph Ω.

$$T_{\text{sst}} = \arg\min_{T \in \Omega} \left(\sum_{(i,j,u,v) \in T} s(M_i(t_u), M_j(t_v)) \right) \quad (1)$$

Parallel implementation of Prim's minimum spanning tree algorithm requires $O(n \log n)$ time where n is the number of graph nodes [32]. This is prohibitively expensive for the graph Ω that typically has $10^3 - 10^5$ nodes. In practice and as can be observed from the similarity matrix, Figure 1(c), many mesh pairs have a low similarity and are therefore not suitable for pairwise alignment. To reduce the computational cost in constructing the shape similarity tree, we prune edges in the graph Ω according to a minimum similarity threshold s_{min}. A suitable similarity threshold can be calculated automatically from the similarity matrix as the minimum of the maximum similarity for each row in the matrix. Setting the threshold in this way ensures that all frames have at least one tree connection within the threshold. Computation time for a 1500 node graph takes < 2 seconds on a single CPU.[†]

3.2. Mesh Sequence Alignment

The shape similarity tree T_{sst} defines the shortest path of non-rigid surface motion required to align the mesh for every frame across all sequence. Starting from the root node M_{root}, we align meshes along the branches of the tree by using a pairwise non-rigid alignment.

Non-rigid pairwise mesh alignment uses a coarse-to-fine approach combining geometric and photometric matching in a Laplacian mesh deformation framework [33]. This builds on recent work using Laplacian mesh deformation for sequential frame-to-frame alignment over mesh

[†]All paper timings are single threaded on an Intel Q6600 2.4GHz CPU.

sequences [1,14]. Here, we use both photometric scale-invariant feature transform features [34] and geometric rigid patch matching [14] to establish correspondence between pairs of meshes. The combination of geometric and photometric features increases reliability of matching by ensuring that there is a distribution of correspondences across the surface. Alignment is performed starting from a coarse sampling (30 patches) which allows large deformations and recursively doubling the number of patches in successive iterations to obtain an accurate match to the surface. Because estimated feature correspondences are likely to be subject to matching errors, we use an energy-based formulation to introduce feature matches as soft constraints on the Laplacian deformation framework as proposed in [33]:

$$\bar{x} = \arg\min_{x} \|Lx - \delta(x_0)\|^2 + \|W_c(x - x_c)\|^2 \qquad (2)$$

L is the mesh Laplacian, $\delta(x_0)$ are the mesh differential coordinates for the source mesh with vertex positions x_0. x is a vector of mesh vertex positions used to solve for $Lx = \delta$. x_c are soft constraints on vertex locations given by the feature correspondence with a diagonal weight matrix W_c. A tetrahedral Laplacian system [1] is used on the basis of the discrete tetrahedron gradient operator G [35] with $L = G^T DG$, where D is a diagonal matrix of tetrahedral volumes. Equation (2) solves for the non-rigid deformation which minimises the change in shape whilst approximating the feature correspondence constraints.

Pairwise non-rigid alignment across the branches of the shape similarity tree T_{sst} results in known correspondence between the root mesh M_{root} and all other meshes $M_i(t_u)$. This correspondence allows every mesh to be re-sampled with the structure of the root mesh, giving a consistent connectivity for all frames over all captured mesh sequences.

Results of global alignment are presented in Figure 2. A street dancer sequence of 1800 frames is aligned using the non-sequential presented approach. Mesh alignment is shown using a flower patter texture applied over the mesh.

4. MESH SEQUENCE PARAMETERISATION

Interactive animation from temporally aligned mesh sequences requires the combination of multiple captured sequences to allow continuous real-time control of movement with intuitive high-level parameters such as speed and direction for walking or height and distance for jumping. Methods for parameterisation of skeletal motion capture have previously been introduced [9,27,28] which allow continuous high-level movement control by linear interpolation of joint angles. Blending of meshes based on linear interpolation of vertex position is computationally efficient but may result in unrealistic deformation or mesh collapse if there are significant differences in shape. Non-linear blending of meshes produces superior deformation [33,36–38] but commonly requires least-squares solution of a system of equations which is prohibitive for real-time interaction. In this paper, we introduce a hybrid solution which approximates the non-linear deformation whilst maintaining real-time performance. This allows high-level parameterisation for interactive movement control by blending of multiple mesh sequences.

Three steps are required to achieve high-level parametric control from mesh sequences: time-warping to align the mesh sequences; non-linear mesh blending of the time-warped sequences; and mapping from low level blending weights to high-level parameters (speed, direction, etc.). In this section, we focus on real-time non-linear mesh blending which is the novel contribution of this work. As in previous work on skeletal motion parameterisation, we assume that individual mesh sequences $M_i(t)$ are temporally aligned by a continuous time-warp function $t = f(t_u)$ [25,39] which aligns corresponding poses prior to blending such that $t \in [0, 1]$ for all sequences.

Figure 2. Top: reconstructed frames of a street dancer sequence, each of them coloured differently to represent a non-aligned structure. Bottom: aligned frames, coloured using a flower patter to show the performance of the non-sequential alignment algorithm.

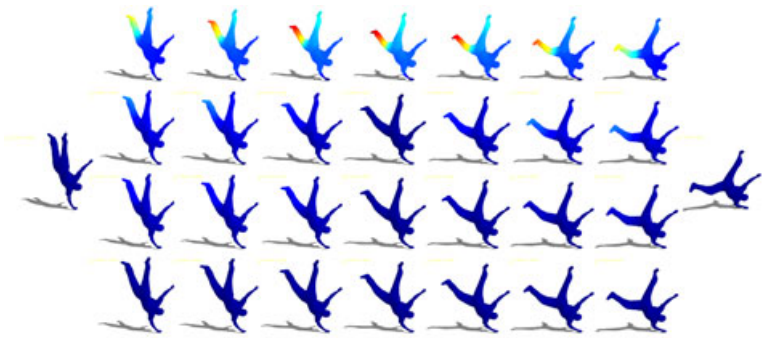
DOI: 10.1002/cav

4.1. Real-Time Non-Linear Mesh Sequence Blending

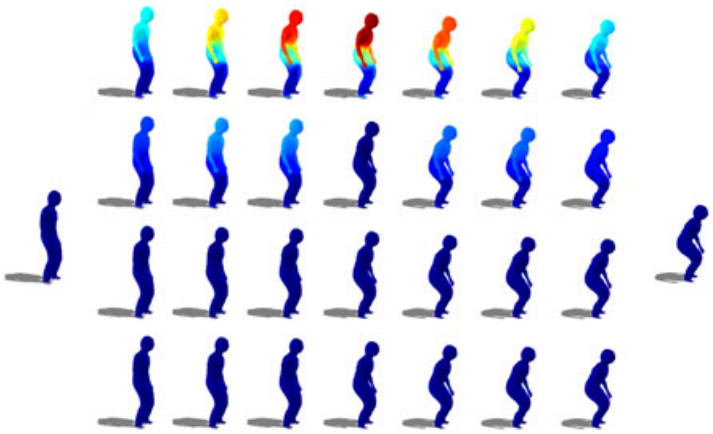
In this work, we introduce a real-time approach to mesh blending that exploits offline pre-computation of non-linear deformation for a small set of intermediate parameter values. Differences between the linear and non-linear mesh deformation are pre-computed and used to correct errors in linear deformation. This approach approximates the non-linear to within a user-specified tolerance whilst allowing real-time computation with a similar cost to linear blending. The price paid is a modest increase in memory required to store intermediate non-linear mesh displacements for blending.

Given a set of N temporally aligned mesh sequences $\mathbf{M} = \{M_i(t)\}_{i=1}^{N}$ of the same or similar motions (e.g. walk and run), we want to compute a blended mesh deformation according to a set of weights $\mathbf{w} = \{w_i\}_{i=1}^{N}$: $M_{NL}(t, \mathbf{w}) = b(\mathbf{M}, \mathbf{w})$ where $b()$ is a non-linear blend function which interpolates the rotation and changes in shape independently for each element on the mesh according to the weights $\mathbf{w}$ and performs a least-squares solution to obtain the resulting mesh $M_{NL}(t, \mathbf{w})$. This non-linear operation can be performed offline by using existing approaches [33,36–38]; throughout this work, we employ a volumetric Laplacian deformation framework based on Equation (2).

Linear vertex blending gives an approximate mesh $M_L(t, \mathbf{w})$ as a weighted sum of vertex positions: $M_L(t) = \frac{1}{\sum w_i} \sum w_i M_i(t)$, where $w_i M_i(t)$ denotes the product of the mesh vertex positions $X_i(t)$ by weight w_i. Given the non-linear mesh deformation $M_{NL}(t, \mathbf{w})$ and linear approximation $M_L(t, \mathbf{w})$, we can evaluate a displacement field: $D_{NL}(t, \mathbf{w}) = M_{NL}(t, \mathbf{w}) - M_L(t, \mathbf{w})$. The exact non-linear deformation for blend weights $\mathbf{w}$ can then be recovered by linear interpolation together with a non-linear



(a)



(b)

Figure 3. Comparison of linear, non-linear and hybrid mesh blending. Heat-maps show error versus non-linear blending from dark-blue (zero) to red (maximum). (a) Result of blending two poses of a street dancer using linear (top row), hybrid with one and three reference meshes (2nd/3rd row) and non-linear (bottom row). Top row shows that linear blending results in large errors (red) for the left leg which are corrected with the hybrid approach. (b) Result of blending two equivalent poses of the low jump (left) and high jump (right) sequences. Second and third rows show that our proposed hybrid approach, with one and three references meshes, respectively, gives an approximation to the non-linear blending, whereas the top row shows large errors with linear blending.

Table I. Maximum vertex displacement error with respect to non-linear blending as a percentage of model size for in meshes in Figure 3.

Sequence	# of vertices	Method	Max. error (%)	Time (seconds/frame)
a) Street dancer	5580	Linear	14.38	0.008
		Hybrid 1 reference	3.67	0.015
		Hybrid 3 references	1.60	0.017
		Non-linear	0.00	0.749
b) Jumping pose	3000	Linear	9.14	0.004
		Hybrid 1 reference	1.34	0.014
		Hybrid 3 references	0.93	0.016
		Non-linear	0.00	0.789

Figure 4. Top row: left and right, two equivalent poses of walk and run sequences. In the middle, the resulting linear blended mesh, coloured using a heat-map (red largest error) to display the errors with respect to the non-linear result. Bottom row: results of our hybrid approach using different threshold ϵ (3, 6, 8 and 14 mm.). Grey represent areas with errors below the threshold (non corrected), and blue represents areas above the threshold that have been corrected.

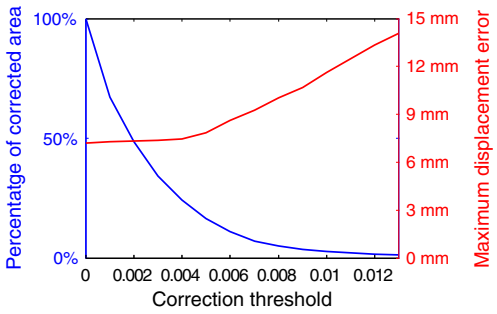
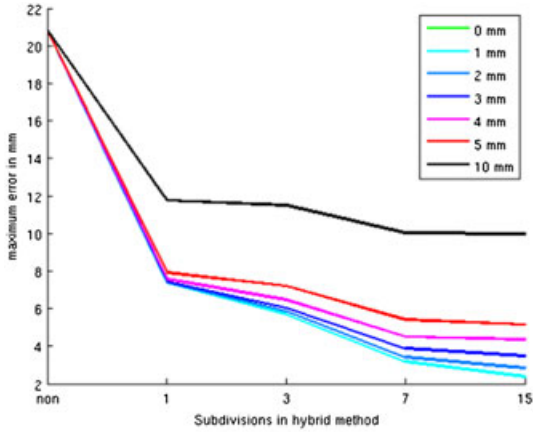
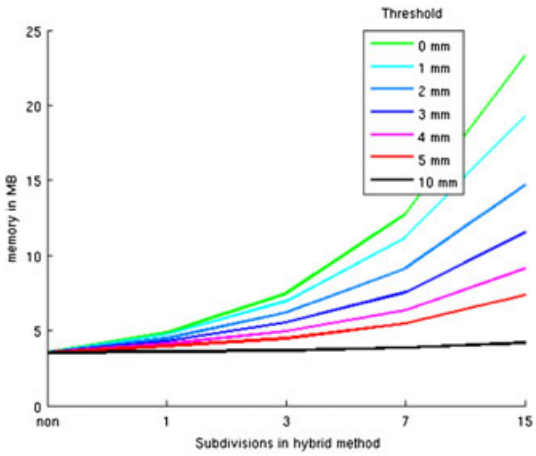


Figure 5. Maximum displacement error (red) and percentage of mesh corrected area (blue) for the pose interpolation shown in Figure 4, against the value of the threshold ϵ.

correction: $M_{NL}(t, \mathbf{w}) = M_L(t, \mathbf{w}) + D_{NL}(t, \mathbf{w})$. An advantage of storing the displacement field D_{NL} is that for blending between mesh sequences of similar motions, linear blending gives a reasonable approximation for large



(a)



(b)

Figure 6. Evaluation of the maximum error and memory usage of the hybrid blending method applied in the walk/run motion space shown in Figure 7(a). (a) Maximum displacement error of the hybrid method, depending on the number of subdivisions and threshold ϵ. (b) (b) Memory in MB of the precomputed matrix DNL, depending on number of subdivisions and threshold ϵ.

Comp. Anim. Virtual Worlds 2012; **23**:101–111 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

parts of the surface $D_{NL} \approx 0$ allowing efficient compression whilst accurately reproducing regions of significant non-linear deformation.

To accurately approximate the non-linear deformation for blending a set of N source meshes $\mathbf{M}$ with arbitrary weights $\mathbf{w}$, we pre-compute the non-linear displacement field $D_{NL}(t, \mathbf{w}_j)$ at a discrete set of intermediate weight values $\mathbf{w}_j$ to give an additional set of N_{NL} reference meshes for interpolation: $M_j(t, \mathbf{w}_j) = (M_L(t, \mathbf{w}_j) + D_{NL}(t, \mathbf{w}_j))$. Real-time online interpolation is then performed using a linear vertex blending with the non-linear correction:

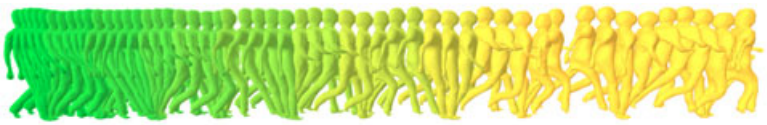
$$M(t, \mathbf{w}) = \sum_{j=1}^{N+N_{NL}} g(\mathbf{w}, \mathbf{w}_j)(M_L(t, \mathbf{w}_j) + D_{NL}(t, \mathbf{w}_j)) \tag{3}$$

where $g(\mathbf{w}, \mathbf{w}_j)$ is a weight function giving a linear blend of the nearest reference meshes and zero for all other meshes. Equation (3) gives an exact solution at the original and non-linear interpolated reference meshes and an approximate interpolation of the nearest reference meshes elsewhere. A recursive bisection of the weight space $\mathbf{w}$ is performed to evaluate a set of non-linearly interpolated source meshes such that for all $\mathbf{w}$, the approximation error $(M_{NL}(t, \mathbf{w}) - M(t, \mathbf{w})) < \epsilon$. Typically, for interpolation of mesh sequences representing related motions, only a single subdivision is required.

4.2. Evaluation of Hybrid Non-Linear Blending

Hybrid non-linear mesh blending allows accurate approximation of non-linear mesh deformation whilst maintaining the computational performance of linear blending to allow real-time interactive animation. Figure 3 (a and b) presents a comparison of errors for linear blending with the proposed hybrid non-linear approach with one and three interpolated reference meshes. Table I presents quantitative results for error and CPU-time. This shows that the proposed real-time hybrid non-linear mesh blending approach achieves accurate approximation even with a single intermediate non-linear displacement map (2nd row), whereas linear blending results in large errors (top).

The influence of the threshold ϵ in the final result is shown in Figure 4. For threshold values larger than 6 mm, only the areas with large errors (i.e. the limbs) are corrected by our hybrid approach, whereas the remaining of the mesh remains identical to the results of the linear approach. If the threshold ϵ is set to lower values, more regions of the



(a) Walk-to-run speed parametrisation, from walk (green) to run (yellow)

(b) Length of jump parametrised from short (red) to long (orange)

(c) Height of jump parametrised from low (grey) to high (purple)

(d) walk (green) to jog (yellow) motion parameterisation of a female character

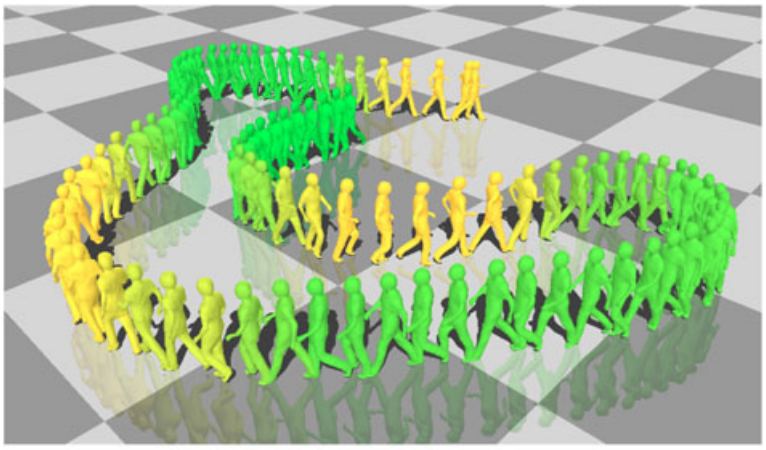
Figure 7. Examples of parameterised motions between two motion sequences with continuous parameter variation (every 5^{th} frame). (a) Walk-to-run speed parameterisation, from walk (green) to run (yellow), (b) length of jump parameterised from short (red) to long (orange), (c) height of jump parameterised from low (grey) to high (purple), and (d) walk (green) to jog (yellow) motion parameterisation of a female character.

linearly blended mesh will be corrected, although this might be unnecessary because they do not produce visual artefacts. This in an important observation that allows us to decrease the size of D_{NL} without compromising on perceived visual quality, as shown in z 6. Figure 5 presents a quantitative evaluation for the meshes blended in Figure 4, showing how the percentage of the area of the mesh, which is corrected, depends on ϵ and its influence on the final displacement error.

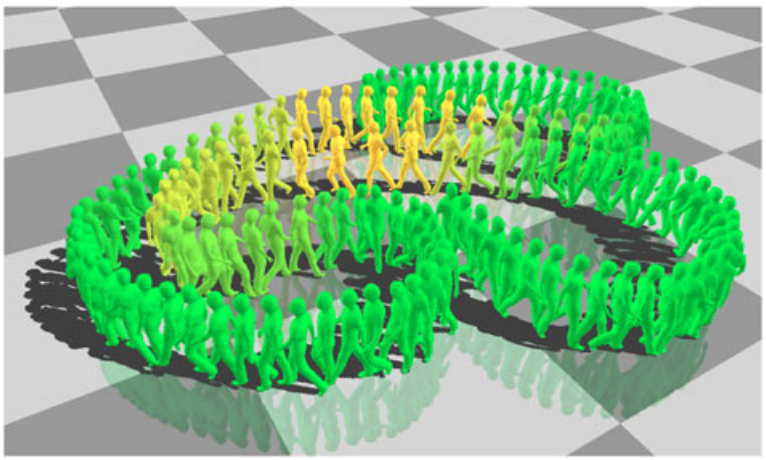
Figure 6 characterises the representation error and storage cost against the number of subdivisions for different error thresholds ϵ. Figure 6(b) shows that there is a relatively small error reduction for thresholds below 5 mm, whereas the memory usage increases significantly. This is caused by the 5 mm resolution of the original database, because details below this level are not reconstructed.

4.3. High-Level Parametric Control

High-level parametric control is achieved by learning a mapping function $h(\mathbf{w})$ between the blend weights and user-specified motion parameters $\mathbf{p}$. As in skeletal motion blending, the blend weights do not provide an intuitive parameterisation of the motion. We therefore learn a mapping $\mathbf{w} = h^{-1}(\mathbf{p})$ from the user-specified parameter to the corresponding blend weights required to generate the desired motion. Motion parameters $\mathbf{p}$ are high-level user-specified controls for a particular class of motions such as speed and direction for walk or run and height and distance for a jump. The inverse mapping function $h^{-1}()$ from parameters to weights can be constructed by a discrete sampling of the weight space $\mathbf{w}$ and evaluation of the corresponding motion parameters $\mathbf{p}$ [27].



(a) Example 1 of a path interactively travelled.



(b) Example 2 of a path interactively travelled.

Figure 8. Meshes showing the path interactively travelled by the user. Four original motions (walk, jog, 90° left/right turn) allow full control of both speed and direction.

DOI: 10.1002/cav

5. RESULTS

Datasets used in this work are reconstructed from multiple-view video capture of actor performance with 8HD cameras equally spaced in a circle and capture volume 5 m^2 × 2 m. Reconstruction is performed using multi-view stereo followed by temporal alignment of all frames to have a consistent mesh structure. Throughout this work, we use a single intermediate mesh for hybrid non-linear interpolation which gives a very close approximation, as shown in Figure 3.

Figure 7 shows parameterised motion spaces for walking and jumping constructed from pairs of mesh sequences. Figure 8 presents a multi-parameter character animation constructed from four mesh sequences with walking speed and direction control, in which our hybrid blending method with 1 reference runs at 0.020 seconds/frame using four input sequences and three blending weights, with a maximum displacement error of 0.73% with respect to the non-linear approach. Rendered meshes are coloured to show the parameter change. The supplementary video[‡] shows the real-time interactive animation control. These results show that the proposed mesh sequence blending approach using the hybrid non-linear deformation achieves a natural transition between the captured motions.

6. CONCLUSIONS

A system for real-time interactive character animation from multiple camera capture has been presented. The approach is based on a database of temporally aligned mesh sequence reconstructions of an actor performing multiple motions. Extending previous research, a shape similarity tree has been used to allow non-sequential alignment of a database of captured sequences into a consistent mesh structure with temporal correspondence. Real-time continuous high-level parametric motion control is achieved by blending multiple mesh sequences of related motions. This allows interactive control of a 3D video character, analogous to previous research with skeletal motion capture [9,26]. A hybrid mesh blending approach has been introduced, which combines the realistic deformations achieved with non-linear blending [33,36–38] with the fast performance of linear blending. An evaluation of the proposed approach has been presented, showing how the final performance varies depending on the threshold ϵ and how memory and time requirements scale. Results for a single CPU-based implementation of the parametric control demonstrate a frame-rate $>$ 50 Hz for a mesh resolution of 3000 vertices. Further research will investigate transitions between parametric motions, which would increase the range of parameterised motions in our final scenario. The incorporation of photorealism in the final render will also be investigated.

[‡]http://vimeo.com/28291052

REFERENCES

1. de Aguiar E. d, Stoll C, Theobalt C, Ahmed N, Seidel H-P, Thrun S. Performance capture from sparse multi-view video. *ACM SIGGRAPH* 2008; **27**(3): 1–10.
2. Vlasic D, Baran I, Matusik W, Popović J. Articulated mesh animation from multi-view silhouettes, In *Proceedings of ACM SIGGRAPH*, Los Angeles, California, 2008; 1–10.
3. Starck J, Hilton A. Surface capture for performance based animation. *IEEE Computer Graphics and Applications* 2007; **27**(3): 21–31.
4. Baran I, Vlasic D, Grinspun E, Popovic J. Semantic deformation transfer, In *Proceedings of ACM SIGGRAPH*, New Orleans, Louisiana, 2009; 1–6.
5. Stoll C, Gall J, de Aguiar E, Thrun S, Theobalt C. Video-based reconstruction of animatable human characters, In *ACM SIGGRAPH ASIA*, Seoul, South Korea, 2010; 1–10.
6. Starck J, Miller G, Hilton A. Video-based character animation, In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005; 49–58.
7. Huang P, Hilton A, Starck J. Human motion synthesis from 3D video, In *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR*, Miami, Florida, 2009; 1478–1485.
8. Xu F, Liu Y, Stoll C, Tompkin J, Bharaj G, Dai Q, Seidel H-P, Kautz J, Theobalt C. Video-based characters — creating new human performances from a multi-view video database, In *Proceedings of ACM SIGGRAPH*, Vancouver, Canada, 2011; 1–10.
9. Rose C, Cohen M, Bodenheimer B. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 1998; **18**(5): 32–40.
10. Casas D, Tejera M, Guillemaut J-Y, Hilton A. Parametric control of captured mesh sequences for real-time animation. In *Proceedings of the 4th International Conference on Motion in Games (MIG)*, Vol. 7060, Lecture Notes in Computer Science. Springer, Edinburgh, UK, 2011; 242–253.
11. Kanade T, Rander P. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia* 1997; **4**(2): 34–47.
12. Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R. High-quality video view interpolation using a layered representation. Los Angeles, California, 2004; 600–608.
13. Carranza J, Theobalt C, Magnor M, Seidel H-P. Free-viewpoint video of human actors, In *Proceedings of ACM SIGGRAPH*, San Diego, California, 2003; 565–577.

14. Cagniart C, Boyer E, Ilic S. Free-form mesh tracking: a patch-based approach, In *Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010; 1339–1346.
15. Vedula S, Baker S, Rander P, Collins R, Kanade T. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2005; **27**(3): 475–480.
16. Wand M, Adams B, Ovsianikov M, Berner A, Bokeloh M, Jenke P, Guibas L, Seidel H-P, Schilling A. Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data. *ACM Transactions on Graphics* 2009; **28**(2).
17. Huang P, Budd C, Hilton A. Global temporal registration of multiple non-rigid surface sequences, In *CVPR*, Colorado Springs, Colorado, 2011; 3472–3480.
18. Cand Budd, Huang P, Hilton A. Hierarchical shape matching for temporally consistent 3D video, In *Proceedings International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT 2011)*, Hangzhou, China, May 2011.
19. Bregler C, Covell M, Slaney M. Video rewrite: driving visual speech with audio, In *Proceedings of ACM SIGGRAPH*, Los Angeles, California, 1997; 1–8.
20. Schodl A, Szeliski DH, amd Salesin R, Essa IA. Video textures, In *Proceedings of ACM SIGGRAPH*, New Orleans, Florida, 2000; 489–498.
21. Flagg M, Nakazawa A, Zhang Q, Kang S-B, Ryu YK, Essa I, Rehg JM. Human video textures, In *ACM Symposium on Interactive 3D Graphics*, Boston, Massachusetts, 2009.
22. Gleicher M. Motion editing with spacetime constraints, In *ACM Symposium on Interactive 3D Graphics*, 1997.
23. Lee J, Shin SY. A hierarchical approach to interactive motion editing for human-like figures, In *Proceedings of ACM SIGGRAPH*, 1999; 39–48.
24. Popovic Z, Witkin A. Physically based motion transformation, In *Proceedings of ACM SIGGRAPH*, 1999; 11–20.
25. Brundelin A, Williams L. Motion signal processing, In *Proceedings of ACM SIGGRAPH*, 1995; 97–104.
26. Wiley DJ, Hahn JK. Interpolation synthesis for articulated figure motion, In *IEEE Virtual Reality International Symposium*, 1997; 157–160.
27. Kovar L, Gleicher M. Automated extraction and parameterization of motions in large date sets. In *Proceedings of ACM Siggraph* 2004; **23**(3): 559–568.
28. Mukai T, Kuriyama S. Geostatistical motion interpolation, In *Proceedings of ACM SIGGRAPH*, Los Angeles, California, 2005; 1062–1070.
29. Seitz SM, Curless B, Diebel J, Scharstein D, Szeliski R. A comparison and evaluation of multi-view stereo reconstruction algorithms, In *Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, New York, 2006; 519–528.
30. Tung T, Matsuyama T. Dynamic Surface matching by geodesic mapping for animation transfer, In *Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010; 1402–1409.
31. Peng Huang, Adrian Hilton, Jonathan Starck. Shape similarity for 3d video sequences of people. *International Journal of Computer Vision* September 2010; **89**: 362–381.
32. Chong KW, Han Y, Lam TW. Concurrent threads and optimal parallel minimum spanning trees algorithm. *Journal of the ACM* 2001; **48**(2): 297–323.
33. Sorkine O. Differential representations for mesh processing. *Computer Graphics Forum* 2006; **25**(4): 789–807.
34. Lowe D. Distinctive image features for scale invariant keypoints. *International Journal of Computer Vision* 2004; **60**(2): 91–110.
35. Botsch M, Sorkine O. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics* 2008; **14**(1): 213–230.
36. Kircher S, Garland M. Free-form motion processing. *ACM Transactions on Graphics* 2008; **27**(2): 1–13.
37. Sumner RW, Popovic J. Deformation Transfer for triangle meshes, In *Proceedings of ACM SIGGRAPH*, Los Angeles, California, 2004; 399–405.
38. Xu W, Zhou K, Yu Y, Peng Q, Guo B. Gradient domain editing of deforming mesh sequences. *Proceedings of ACM SIGGRAPH* 2007; **26**(3).
39. Witkin A, Popovic Z. Motion Warping, In *Proceedings of ACM SIGGRAPH*, Los Angeles, California, 1995; 105–108.

AUTHORS' BIOGRAPHIES

Dan Casas is a PhD candidate at the Centre for Vision, Speech and Signal Processing, University of Surrey, UK. He received an MEng. degree from Universitat Autònoma de Barcelona in 2009. His research interests include character animation, motion synthesis and free-viewpoint video.

Margara Tejera is a PhD candidate at the Centre for Vision, Speech and Signal Processing, University of Surrey, UK. She holds a Masters degree in Telecommunication Engineering from the University of Seville, Spain. Her current research work aims to generate algorithms and tools that allow the synthesis of novel 3D video sequences.

Jean-Yves Guillemaut received an MEng degree from the Ecole Centrale de Nantes, France, in 2001, and a PhD degree from the University of Surrey, UK, in 2005. He is currently a research fellow in the Centre for Vision, Speech and Signal Processing, University of Surrey, UK. His research interests include free-viewpoint video and 3D TV, image/video-based scene reconstruction and rendering, image/video segmentation and matting, camera calibration and active appearance models for face recognition.

Adrian Hilton (BSc(hons), DPhil, CEng) is a professor of Computer Vision and Director of the Centre for Vision, Speech and Signal Processing at the University of Surrey, UK. His research interest is robust computer vision to model and understand real-world scenes. Contributions include technologies for the first hand-held 3D scanner, modelling of people from images and 3D video for games, broadcast and film. He currently leads research investigating the use of computer vision for applications in entertainment content production, visual interaction and clinical analysis.