Editing Mesh-based Character Animations using Sample Sequences

Norimichi UKITA† and Edilson DE AGUIAR††

† Nara Institute of Science and Technology

†† Disney Research Pittsburgh

E-mail: †ukita@is.naist.jp, ††edilson@disneyresearch.com

Abstract We propose a new simple approach to represent and manipulate a mesh-based character animation preserving its time-varying details. Our method first decomposes the input mesh animation into coarse and fine deformation components. A model for the coarse deformations is constructed by an underlying kinematic skeleton structure and blending skinning weights. Thereafter, a non-linear probabilistic model is used to encode the fine time-varying details of the input animation. The user can manipulate the corresponding skeleton-based component of the input, which can be done by any standard animation package, and the final result is generated including its important time-varying details. By converting an input sample animation into our new hybrid representation, we are able to maintain the flexibility of mesh-based methods during animation creation while allowing for practical manipulations using the standard skeleton-based paradigm. We demonstrate the performance of our method by converting and editing several mesh animations generated by the state-of-the-art performance capture approaches.

Key words 3D Character Animation, Mesh Animation, Motion Editing, Gaussian Process Latent Variable Models

1. Introduction

Recently, a variety of mesh-based approaches have been developed that enable the generation of computer animations without relying on the classical skeleton-based paradigm [1]. The advantage of a deformable model representation is also demonstrated by the new performance capture approaches [2], [3], where both motion and surface deformations can be captured from input video-streams for arbitrary subjects. This shows the great flexibility of a mesh-based representation over the classical one during animation creation.

Although bypassing many drawbacks of the conventional animation pipeline, a mesh-based representation for character animation is still complex to be edited or manipulated. Few solutions are presented in the literature [4]〜[8], but in general it is still hard to integrate these methods into the conventional pipeline. Other approaches try to convert or represent mesh animations using a skeleton-based representation to simplify the rendering [9] or editing tasks [2], [10]. However, these editing methods are not able to preserve fine time-varying details during the manipulation process, as for instance the waving of clothes for a performing subject.

For editing mesh-based character animations, an underlying representation (i.e. skeleton) is desired since it simplifies the overall process. At the same time, the time-varying details should be preserved during manipulation. These two constraints guide the design of our new hybrid representation for mesh-based character animation. Our method decomposes the input mesh animation into coarse and fine deformation components. A model for the coarse deformation is constructed automatically using the conventional skeleton-based paradigm (i.e. kinematic skeleton, joint parameters and blending skinning weights). Thereafter, a model to encode the time-varying details is built by learning the fine deformations of the input over time using a pair of linked Gaussian process latent variable models (GPLVM [11]). Our probabilistic non-linear formulation allow us to represent the time-varying details as a function of the underlying skeletal motion as well as to generalize to different configurations such that we are able to reconstruct details for edited poses that were not used during training. By combining both models, we simplify the editing process: animators can work directly using the underlying skeleton and the corresponding time-varying details are reconstructed in the final edited animation.

We demonstrate the performance of our approach by performing a variety of edits to mesh animations generated from the state-of-the-art performance capture methods. As seen in Fig. 1 and in the results (Sect. 6.), our approach is able to convert a mesh-based character animation into a new hybrid representation that is more flexible for editing purposes and it can be easily

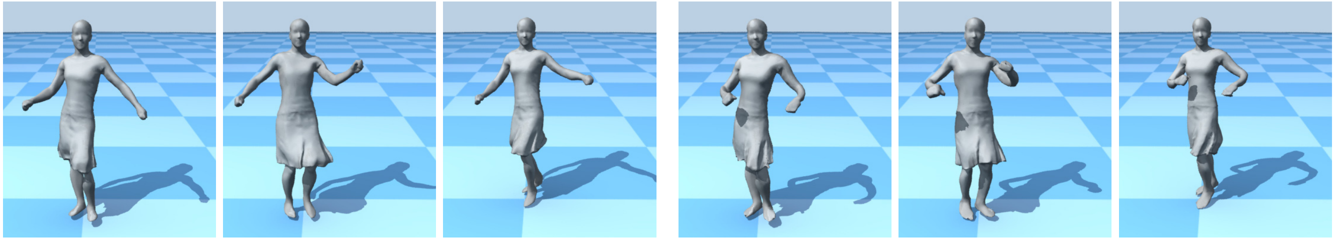


Fig. 1 Our approach represents an input mesh-based character animation (left - three particular frames) into a new hybrid representation that simplifies the editing process and preserves the important time-varying details (right - edited frames).

integrated in the conventional animation pipeline.

The main contributions of our paper are:

- an approach to efficiently convert a mesh-based character animation into a skeleton-based representation;
- a robust method to learn time-varying details using a non-linear probabilistic technique;
- a simple approach to represent and edit a mesh-based character animation preserving its time-varying details.

The paper is structured as follows: Sect. 2. reviews the most relevant related work and Sect. 3. briefly describes our overall approach. Thereafter, Sect. 4. details our automatic method to convert a mesh-based character animation into the skeleton-based format and Sect. 5. describes how the time-varying details are learned using a non-linear probabilistic technique. Experiments and results are shown in Sect. 6. and the paper concludes in Sect. 7..

2. Related Work

Creating animations for human subjects is still a time-consuming and expensive task. In the traditional framework, the character animation is represented by a surface mesh and an underlying skeleton. The surface geometry can be hand-crafted or scanned from a real subject and the underlying skeleton is manually created, inferred from marker trajectories [12] or inferred from the input geometry [13]. The skeleton model is animated by assigning motion parameters to the joints and the geometry and skeleton are connected via skinning (see [14] for a review).

Given the complexity of this process, many related methods have been developed to simplify this pipeline bypassing many drawbacks of the conventional framework [1]. In particular, the recent progress of deformation transfer [15], [16], surface capture [17], [18] and mesh-based performance capture methods [2], [3] is enabling the creation of an increasing number of mesh-based animations for human subjects. As a result, editing and reusing these animations is becoming an important issue.

A number of approaches have been developed to process and edit general mesh animations [4]∼[8], but unfortunately these methods can not be easily used by animators or integrated into the conventional animation pipeline. For animations that can be represented by an underlying kinematic skeleton, e.g. human subjects, an underlying representation is more flexible for editing operations, it enables its integration into a conventional animation package and it simplifies the overall process. Recent techniques to simplify the rendering task for such mesh animations [9] and new methods to convert a sequence of mesh poses [10] or mesh animations [2] to a skeleton-based format have been investigated. Our technique extends these latter editing approaches by preserving the fine time-varying details during the manipulation process, which increases the quality of the final result (Fig. 1).

In our framework, surface time-varying details are encoded and preserved by a non-linear probabilistic technique. In contrast to related approaches dealing with human skin deformations [19]∼[21], our method is even able to model deformations of loose apparel. Considering that the underlying subspace of deformations is inherently non-linear, we believe that a non-linear dimensionality technique is appropriate to compactly represent these deformations. Among the non-linear dimensionality reduction approaches, Gaussian Process Latent Variable Models (GPLVM [11]) has been shown to robustly generalize well from small training sets and it does not tend to over-fit as other techniques. Recently, a variety of GPLVM approaches have been widely used for learning human motion either using a dynamic representation [22] or a shared latent structure [23]. These techniques were also used to model large dimensional

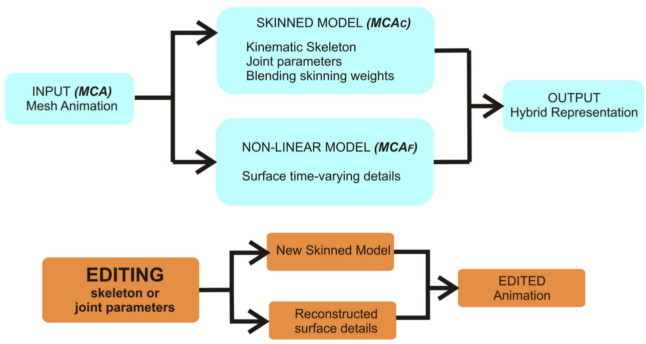


Fig. 2 Overview of our method: an input mesh-based character animation is decomposed into coarse (MCA_C) and fine (MCA_F) deformation components. This enables the user to edit the underlying skeleton-based representation and the time-varying details of the input are faithfully reconstructed in the final animation.

data, such as silhouettes [24], voxel data [25] and even simple deformable models [26]. However, to the best of our knowledge, such technique was never been used to learn time-varying surface details for more complex models like in our system.

3. Overview

An overview of our approach is shown in Fig. 2. The input to our method is an animated mesh sequence comprising of N_{FR} frames. The mesh-based character animation ($MCA = [M, p_t]$) is represented by a sequence of triangle mesh models $M = (V = vertices, T = triangulation)$ and position data $p_t(v_i) = (x_i, y_i, z_i)_t$ for each vertex $v_i \in V$ at all time steps t.

Our framework is inspired by Botsch and Kobbelt [27], where a new representation for mesh editing is proposed using a multiresolution strategy. In contrast to their method, our system can be applied to a sequence of spatio-temporal coherent meshes and it allows the manipulation of the entire animation by decomposing it into coarse (MCA_C) and fine (MCA_F) deformation components. A model for coarse deformations is created by automatically fitting a kinematic skeleton to the input and by calculating the joint parameters and blending skinning weights such that the input animation is reproduced as close as possible, Sect. 4..

Unfortunately, only a skeleton-based model is not able to represent the fine time-varying details of the input. In order to encode such details, a GPLVM-based technique is used to learn the motion-dependent fine non-rigid details, Sect. 5.. The combination of both models not only enables the conversion of the input mesh-based character animation in a new hybrid representation, but it also enables its manipulation preserving the important time-varying details, Sect. 6..

4. Skeleton-based Representation

Giving an input mesh-based character animation MCA, a skinned model (MCA_C) is created to reproduce the coarse deformation component of the input animation. This is done by automatically fitting a kinematic skeleton to the input character model (i.e. triangle mesh at first frame of the animation) and by calculating the joint parameters (θ) and blending skinning weights such that MCA_C reproduces MCA approximately.

Our goal is to deal with human-like characters. Therefore, we include prior knowledge in our framework by means of a known kinematic skeleton structure, Fig 3(left). Our kinematic structure contains $N_{JOINTS} = 18$ joints connecting bone segments and its joint hierarchy is presented in Fig. 3(right). We parameterize the skeleton by the translation of the root joint and three angular degrees-of-freedom for all other joints. We fit our kinematic skeleton to the input character model by using the method proposed in [13]. We also use the approach proposed in [13] to compute appropriate blending skinning weights to connect the character model to the underlying kinematic skeleton.

Thereafter, for each frame of the input animation, joint parameters θ are estimated such that the reconstructed skinned model MCA_C best reproduces the input poses in MCA. Starting from the root of the hierarchy and stepping down to the leaves, this is achieved by optimizing the root translation and the angles for each joint in order to minimize the average square deviations between the vertices in the skinned model and the vertices in the input pose for each frame. We perform this optimization for each joint subsequently following the skeleton's hierarchy. In contrast to [2], [10], this simple strategy is fast and it is more robust against artifacts due to the non-rigid components of the input animation. Although we are not as general as the related work regarding the estimation of the underlying skeleton structure, in our experiments, our automatic approach is able to correctly convert animations of different human subjects wearing a variety of clothing styles, Fig. 6.

Our final skinned model MCA_C closely matches the input animation. However, non-rigid time-varying details can not be accurately reproduced in this representation. In the next section, a new method is used to learn such time-varying details which enables the faithful reconstruction and manipulation of the input.

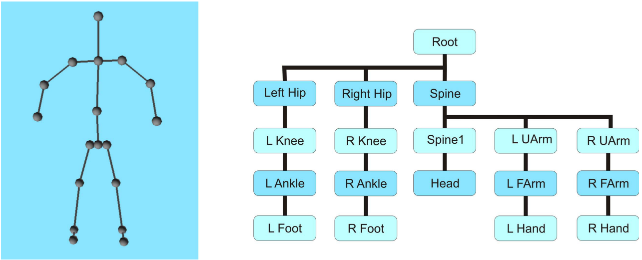


Fig. 3 Prior knowledge is incorporated in our framework by means of a known kinematic skeleton structure (left) containing 18 joints organized hierarchically (right).

5. Learning Time-varying Surface Details

We use a non-linear probabilistic technique to efficiently learn the complex time-varying details of the input, which is inherently non-linear, from a small number of samples (i.e. frames). This is achieved by learning the difference between the input mesh animation and its corresponding skinned model representation. This algorithm design is important because it makes our representation more stable (i.e. by using the coarse skinned animation) and it enables a more detailed and accurate reproduction of the input (see Fig. 5(a) and Sect. 6.). Another advantage is that while absolute coordinate values of neighboring vertices may be completely different, the fine deformations tend to be similar for neighboring vertices, which improves the performance of our learning scheme.

Giving the mesh animation MCA and its skinned model MCA_C, we create the details by subtracting for each vertex v_i its original position $\boldsymbol{p}_t(v_i)$ in MCA from its position $\boldsymbol{ps}_t(v_i)$ in MCA_C at time step t: $\boldsymbol{d}_t(v_i) = \boldsymbol{p}_t(v_i) - \boldsymbol{ps}_t(v_i)$. Finally, $\boldsymbol{y}_{M_t} = [\boldsymbol{d}_t(v_1), \cdots, \boldsymbol{d}_t(v_N)]^T$ is a component of MCA_F at time t.

The skeletal motion (i.e. joint parameters) is linked to the fine deformations of the input model using a shared latent structure of GPLVM, *Shared Gaussian Process Latent Variable Models* (SGPLVM) [23], via a low-dimensional latent space X. In conjunction with *Gaussian Process Dynamical Models* (GPDM) [22], the smoothness of the temporal transitions of the latent variables $\boldsymbol{x}$ is enforced. Latent variables are mapped from the high-dimensional observation spaces $f_S(\boldsymbol{x}) : X \rightarrow Y_S$ and $f_M(\boldsymbol{x}) : X \rightarrow Y_M$, where Y_S represents D_S-dimensional joint parameters and Y_M represents the D_M-dimensional time-varying details, as illustrated in Fig. 4. Inherence of the Gaussian Process technique allows us to optimize our latent space increasing its generalization.

The estimation of the mapping functions $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$ is briefly described in the following. In SGPLVM [23], d-dimensional latent variables $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ corresponding to N given samples in Y_S and Y_M (denoted by $\bar{\boldsymbol{Y}}_S$ and $\bar{\boldsymbol{Y}}_M$, respectively) are acquired by maximizing the joint likelihood of $\bar{\boldsymbol{Y}}_S$ and $\bar{\boldsymbol{Y}}_M$ with respect to $\boldsymbol{X}$. In this optimization, the similarity between components of $\boldsymbol{X}$ (i.e., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ where $i \neq j$) is evaluated by a non-linear kernel function. In our particular case, the similarity is determined in accordance with our sampling data, namely mesh ($\bar{\boldsymbol{y}}_{M_i}$ and $\bar{\boldsymbol{y}}_{M_j}$) and skeleton ($\bar{\boldsymbol{y}}_{S_i}$ and $\bar{\boldsymbol{y}}_{S_j}$). We use RBF for the non-linear kernel function and scaled conjugate gradient (SCG) for the optimization of the non-linear mapping functions $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$.

GPDM [22], which consists of an observation space Y (i.e. Y_S or Y_M) and its latent space X, is defined by two mappings. The first mapping is from the latent space X to the observation space and the second one is from a point at $t-1$ to a point at t in X, $f_D(\boldsymbol{x})$, as illustrated in Fig. 4. Similarly to SGPLVM, these mapping functions are acquired by maximizing the joint likelihood of $\boldsymbol{Y}$ and $\boldsymbol{X}_{t+1}$ with respect to $\boldsymbol{X}$ and $\boldsymbol{X}_t$, where $\boldsymbol{X}_{t+1} = [\boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ and $\boldsymbol{X}_t = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N-1}]$.

In our framework, the shared space $\boldsymbol{X}$ under the dynamics constraint, is acquired by maximizing the product of the joint likelihoods evaluated in SGPLVM and GPDM. In contrast to previous work, where the initialization of $\boldsymbol{X}$ is achieved by canonical correlation analysis (CCA) [28] or averaging the top eigenvectors of PCA [23], in our method, $\boldsymbol{X}$ is initialized by computing the principal components of $\boldsymbol{Y}_S$ and then we optimize the product of the joint likelihoods. Since $D_S \ll D_M$, this approach results in a better initialization and optimization for $\boldsymbol{Y}_M$.

The goal of this learning scheme is to encode time-varying details of the input mesh animation using the joint parameters. In general, a given joint angle configuration might correspond to multiple surface details. GPDM allows us to properly model this situation and obtain an improved latent space by mapping the data with similar details but different motions to different latent variables in X. In order to leverage this advantage, a temporal history of the input skeletal motion is mapped from Y_S to X and then to Y_M. In our implementation, a concatenation of the joint parameters for two frames is employed: $\boldsymbol{y}_{S_t} = [\theta_t, \theta_{t-1}]^T$, where θ_t denotes the skeletal joint parameters at time t. Please note that only joint angles are used for learning (i.e. translation parameters are discarded) and that in our

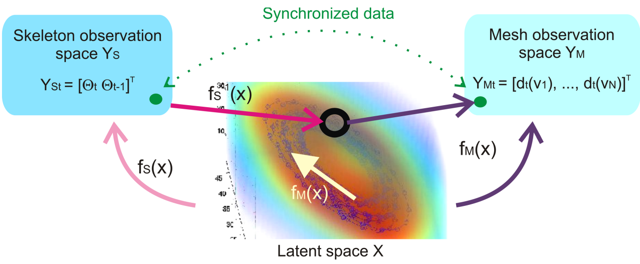


Fig. 4 The relation between joint parameters and surface details is learned using a shared latent space with dynamical constraints. Our model can generalize to different input configurations, as seen by the color-coded variance (blue=low → red=high).

experiments, we achieved better results by discarding the joint angles for the root joint as well.

While the latent space X is optimized by embedding with Gaussian Process, a mapping function from $\boldsymbol{y}_S$ to $\boldsymbol{x}$ ($f^{-1}(\boldsymbol{y}_S) : \boldsymbol{Y}_S \rightarrow \boldsymbol{X}$) is not provided by the above mentioned process. In this work, after the latent space X is optimized, the mapping $\boldsymbol{Y}_S \rightarrow \boldsymbol{X}$ is obtained by a regression function, which is also learned by Gaussian Process [29].

Using $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$, a new model is generated as follows: first the coarse deformation $\boldsymbol{ps}(v_i)$ is estimated from $\boldsymbol{y}_S$ using our skinned model, Sect. 4.. Thereafter, the joint parameters for time t and $t-1$ are mapped to Y_M via X: $\boldsymbol{y}_M = f_M(f_S^{-1}(\boldsymbol{y}_S))$ and the time-varying details $\boldsymbol{d_t}(v_i)$ are calculated. Both terms are added together and the pose for the model is reconstructed. In our experiments, the dimension of the latent space and the number of iterations for the SCG technique are set to be 4 and 100, respectively. These values enable convergence and they are a good trade-off between training speed and accuracy of the final model.

6. Experiments and Results

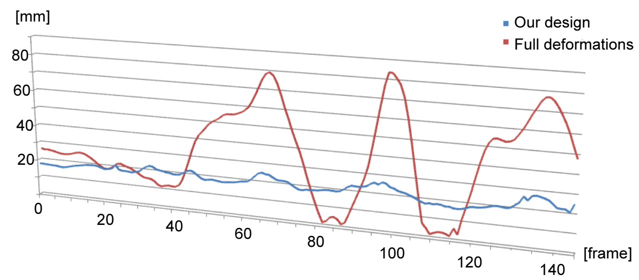
Our approach has been tested on several mesh-based animation sequences generated from performance capture methods that are publicly available [3], [30]. The animations contain walking sequences as well as dancing and fighting sequences. The input meshes were generated at a resolution of around $N_{VERT} = 7000 - 10000$K vertices and the animation sequences range from $N_{FR} =$ 70-400 frames long. In order to evaluate the performance of different algorithmic alternatives, we first ran a series of experiments.

In our first experiment, we verified the efficiency of our system's design by comparing the performance of our non-linear probabilistic model to learn the full range of deformations in contrast to only encoding the time-varying details in Sect. 5.. By encoding coarse and fine deformations, our non-linear model is able to reproduce the input, but unfortunately it is not able to generalize well to different pose configurations. Fig. 5(a) shows the result when we use a model trained with the full deformations (red line) and one trained with only the fine deformations to reconstruct the swing sequence [3]. The graph shows the average distance error between the corresponding vertices of the human-size input animation and our reconstructions. This demonstrates our correct choice by using a non-linear model to encode only the fine time-varying deformations, as described in Sect. 5..

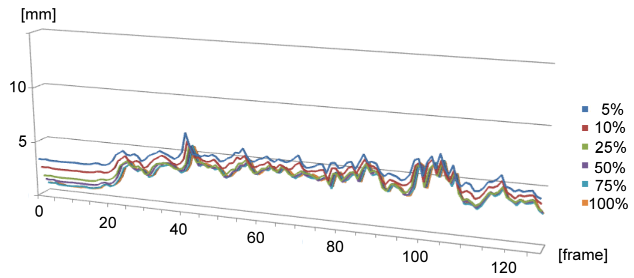
Our second experiment was used to determine the best combination of representations to be applied to our GPLVM-based approach, Sect. 5.. Motion capture data can be represented by euler angles, quaternions or exponential maps. We applied all three representations to our method and in our tests exponential maps performed better. We also tested two common representations for positional data using the samba sequence: vertex displacements in xyz space (XYZ) and differential coordinates [1] (DIF). In our experiments, both mesh representations gave similar results. Therefore, giving the fast generation of XYZ, in contrast to DIF where a linear system needs to be solved for each frame, for the remainder of this paper we use the combination *exponential maps* and XYZ to generate the results.

Given the high dimension space of the input data (i. e. $D_S = 3 \times (N_{JOINTS} - 1) \times N_{FR}$ and $D_M = 3 \times N_{VERT} \times N_{FR}$), our third and last experiment analyzes the performance of our system to handle it, as well as lower dimensional spaces (i.e. mesh resolutions) generated by simplifying the original one. We generated a simplified version of the input animation by decimating the character triangle mesh at the first frame using a surface mesh simplification technique proposed in [31]. We maintain the temporal connectivity in the control mesh animation by saving the sequence of edge collapses for the simplified character model and by applying the same sequence of operations for all meshes in the input sequence. Thereafter, we apply our framework to generate our hybrid representation and perform some manipulations using the control mesh animation. At the end, a radial basis function approach, proposed in [30], is used to reconstruct the fine resolution models based on the sequence of edited control meshes.

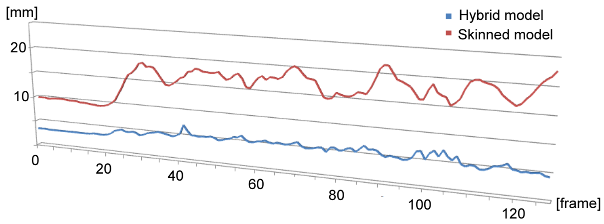
We tested the performance of our system in six different resolution levels: full resolution or 100%, 75%, 50%, 25%, 10% and 5%. As seen in the graph in Fig. 5(b),



(a)



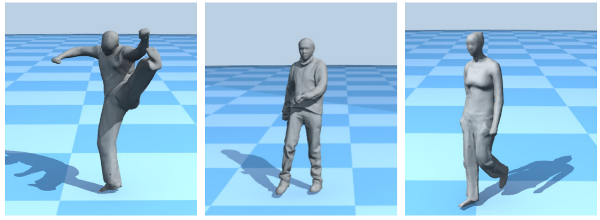
(b)



(c)

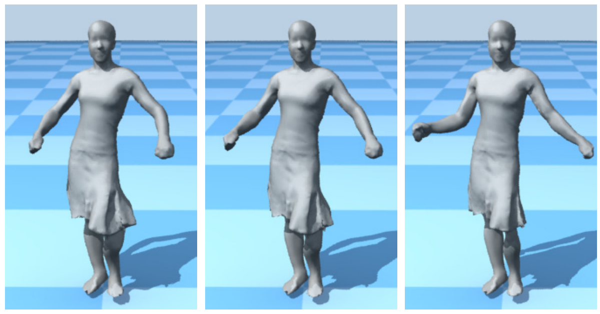
Fig. 5 Experiments for the human-size samba sequence [3]: (a) The graph shows that our system design, where coarse and fine deformations are decomposed, is able to reproduce the input swing sequence more accurately. (b) A multiresolution approach can be used in our framework delivering the same level of quality for the final animation, while decreasing computational power and storage resources. (c) Graph comparing the reconstruction accuracy of our skinned model (red line) and our hybrid representation (blue line) demonstrating the advantage of our algorithm.

the reconstruction accuracy of our system for the challenging samba sequence is similar in all resolution levels. In the accompanying video, we can also see that visually there is not much difference in the final result when we manipulate the control mesh or the full fine resolution animation. Therefore, in order to make our approach more efficient, decreasing its overall processing time, we decided to perform the editing process following a multiresolution strategy using the control mesh at a resolution of 5% ($N_{FR} = 501$). Please note that our system can still be applied to any resolution level and that for all sequences we tested, the time-varying

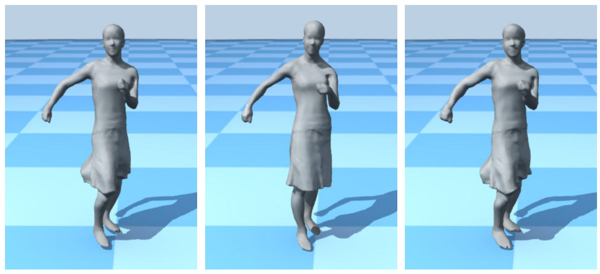


(a)

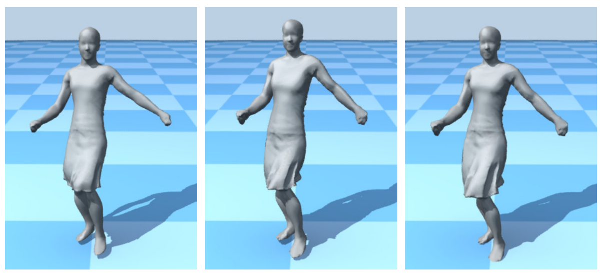
(b)



(c)



(d)



(e)

Fig. 6 Results of our editing mesh-based character animations: input mesh animations (a) and edited animations (b). Our framework allows a more faithful manipulation. For a single frame, different surface details for the skirt can be reconstructed based on the underlying skeletal motion (c – different three frames). We are also able to modify the proportions of the underlying skeleton, simplifying the retargeting of a mesh animation (d and e – left: original frame, middle: skinning results, right: hybrid results).

details of the input animations were preserved during the process. We see this multiresolution scheme as an advantage of our framework as it allows the reduction of processing time and storage without decreasing the overall quality of the animation.

The performance of our framework to automatically convert an input mesh-based character animation to our new hybrid representation is shown in Fig 5(c). By using only the skinned model, as in related approaches [2], [10], time-varying details are not preserved and the reconstruction is not accurate (red line in Fig. 5(c)). Our hybrid solution preserves the details of the input animation which yields a more accurate reconstruction of the input (blue line in Fig. 5(c)). This property of our new representation is specially useful during the manipulation of the input animation, where the important time-varying details are preserved, and a more realistic visual look for the resulting edited animation can be achieved, Fig. 6.

The advantages of our hybrid representation are presented in Fig. 1 amd Fig. 6 and in the accompanying video. In Fig. 1, the motions of the arms, torso and head of the girl dancing samba are edited and the skirt waves faithfully in the final result, increasing the realism of the edited animation. Fig. 6(a,b) shows the input animation and the editing result using our hybrid representation. Using our framework, we are able to change the motion parameters of the underlying skeleton and generate convincing deformations for the skirt, Fig. 6(c). We are also able to change the input skeleton dimensions, Fig. 6(d,e), which enables us to even retarget the input animation to a different character proportion.

The running time of our algorithm is dominated by the training phase of the GPLVM-based technique (around $30min$ for 100 frames) and by the joint parameter estimation in Sect. 4. ($2.5s$/frame). These steps are done only once at the beginning for each sequence and, thereafter, the editing operations run in real-time. These timings were obtained with an Intel Core Duo Laptop at 2.4 GHz. Another advantage of our approach is its ability to compress a mesh-based character animation without losing its time-varying details. Using our lowest multiresolution level (5%), the input animation is compressed to around 5%-10% of its original size.

Despite our method's ability to reproduce and manipulate the input animation, there are a few limitations to be considered. Our current framework is targeted to kinematically-based subjects and therefore it would not perform as well as other methods in the literature [5], [7] for extreme non-rigid deformations, like purely deforming cloth. Currently, the time-varying details can not be directly edited and they are reconstructed based on the motion of the underlying skeleton. Although our system allows the animators to edit the input motion, giving the limited amount and variety of training data available, we are not able to generate details for motions that are too far from the original input. However, we see this as a more general limitation of any learning method and believe that by increasing the variety of training data our framework will be able to deal with more general edits and motions.

Currently, linear blending skinning was used to create the skinned model in Sect. 4., but we believe that similar results can be achieved with a more advanced skinning method [14] and we leave this for future work. We are using a basic GPLVM implementation, but we believe that improvements of this basic technique [32] can increase the performance of our method even further. Important parameters in Sect. 5. (i.e number of iterations, number of latent variables) were found experimentally and kept constant for all sequences. We would like to investigate better ways to determine such parameters in the future as well. Nevertheless, we described a simple framework to represent and manipulate a mesh-based character animation using its underlying kinematic structure and incorporating the reconstruction of its time-varying details.

7. Conclusions

We presented a simple and fast system to represent and manipulate an input sequence of animated characters preserving its important time-varying details. By decomposing the input animation into coarse and fine deformation components, a skinned model and a GPLVM-based technique are used to reproduce the input and to enable its meaningful manipulation. Our new hybrid representation maintains the flexibility of mesh-based methods while it allows for practical manipulations using the conventional animation tools.

As future work, we would like to extend our approach to reconstruct a time-varying non-linear probabilistic model to handle more complex edits and even different input motions. For handling the complex animations, more sophisticated modelings below would be useful:

- Modeling with additional constraints for smooth mapping (e.g. smooth bidirectional mapping [33] and constraints with regard to the topology of motionsc [34]).

- Using rank priors for automatically selecting the dimension of a latent space [32].

The source codes of GPLVM were provided by courtesy of Neil Lawrence.

References

[1] M. Botsch and O. Sorkine: "On linear variational surface deformation methods", IEEE TVCG, **14**, 1, pp. 213–230 (2008).

[2] E. de Aguiar, C. Theobalt, S. Thrun and H.-P. Seidel: "Automatic conversion of mesh animations into skeleton-based animations", Computer Graphics Forum, **27**, 2, pp. 389–397 (2008).

[3] D. Vlasic, I. Baran, W. Matusik and J. Popović: "Articulated mesh animation from multi-view silhouettes", ACM TOG, **27**, 3, p. 97 (2008).

[4] S. Kircher and M. Garland: "Editing arbitrarily deforming surface animations", SIGGRAPH '06, pp. 1098–1107 (2006).

[5] W. Xu, K. Zhou, Y. Yu, Q. Tan, Q. Peng and B. Guo: "Gradient domain editing of deforming mesh sequences", ACM TOG, **26**, 3, p. 84 (2007).

[6] R. W. Sumner, J. Schmid and M. Pauly: "Embedded deformation for shape manipulation", SIGGRAPH '07, ACM, p. 80 (2007).

[7] S. Kircher and M. Garland: "Free-form motion processing", ACM TOG, **27**, 2, pp. 1–13 (2008).

[8] I. Baran, D. Vlasic, E. Grinspun and J. Popović: "Semantic deformation transfer", ACM TOG, **28**, 3, p. 36 (2009).

[9] D. James and C. Twigg: "Skinning mesh animations", ACM TOG, **24**, 3 (2005).

[10] S. Schaefer and C. Yuksel: "Example-based skeleton extraction", SGP '07, pp. 153–162 (2007).

[11] N. Lawrence: "Probabilistic non-linear principal component analysis with gaussian process latent variable models", J. Mach. Learn. Res., **6**, pp. 1783–1816 (2005).

[12] A. G. Kirk, J. F. O'Brien and D. A. Forsyth: "Skeletal parameter estimation from optical motion capture data", Proc. of CVPR '05, IEEE, pp. 782–788 (2005).

[13] I. Baran and J. Popović: "Automatic rigging and animation of 3d characters", ACM Trans. Graph., **26**, 3, p. 72 (2007).

[14] L. Kavan, S. Collins, J. Žára and C. O'Sullivan: "Geometric skinning with approximate dual quaternion blending", ACM TOG., **27**, 4, pp. 1–23 (2008).

[15] R. W. Sumner and J. Popović: "Deformation transfer for triangle meshes", Proc. of SIGGRAPH '04, ACM, pp. 399–405 (2004).

[16] M. Ben-Chen, O. Weber and C. Gotsman: "Spatial deformation transfer", Proc. of SCA, ACM, pp. 67–74 (2009).

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

[18] P. Huang, A. Hilton and J. Starck: "Human motion synthesis from 3d video", Proc. of CVPR '09, IEEE (2009).

[19] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers and J. Davis: "Scape: shape completion and animation of people", ACM TOG, **24**, 3, pp. 408–416 (2005).

[20] B. Allen, B. Curless, Z. Popović and A. Hertzmann: "Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis", Proc SCA'06, Eurographics Association, pp. 147–156 (2006).

[21] S. I. Park and J. K. Hodgins: "Data-driven modeling of skin and muscle deformation", ACM TOG, **27**, 3, pp. 1–6 (2008).

[22] J. M. W., D. J. Fleet and A. Hertzmann: "Gaussian process dynamical models for human motion", IEEE PAMI, **30**, 2, pp. 283–298 (2008).

[23] A. P. Shon, K. Grochow, A. Hertzmann and R. P. N. Rao: "Learning shared latent structure for image synthesis and robotic imitation", Proc. of NIPS'05 (2005).

[24] R. Urtasun and T. Darrell: "Sparse probabilistic regression for activity-independent human pose inference", CVPR (2008).

[25] N. Ukita, M. Hirai and M. Kidode: "Complex volume and pose tracking with probabilistic dynamical models and visual hull constraints", IEEE ICCV'09 (2009).

[26] M. G. Salzmann and P. Urtasun, R.and Fua: "Local deformation models for monocular 3d shape recovery", Proc. of CVPR '08, IEEE (2008).

[27] M. Botsch and L. Kobbelt: "Multiresolution surface representation based on displacement volumes", Computer Graphics Forum, **22**, 3, pp. 483–491 (2003).

[28] C. H. Ek, P. H. S. Torr and N. D. Lawrence: "Gaussian process latent variable models for human pose estimation", International Workshop on Machine Learning for Multimodal Interaction (2007).

[29] C. E. Rasmussen and C. K. I. Williams: "Gaussian Processes for Machine Learning", The MIT Press (2006).

[30] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel and S. Thrun: "Performance capture from sparse multi-view video", ACM TOG (2008).

[31] Q. Fang and D. Boas: "Tetrahedral mesh generation from volumetric binary and gray-scale images", IEEE International Symposium on Biomedical Imaging (2009).

[32] A. Geiger, R. Urtasun and T. Darrell: "Rank priors for continuous non-linear dimensionality reduction", Proc. of CVPR '09, IEEE (2009).

[33] N. D. Lawrence: "Local distance preservation in the gp-lvm through back constraints", Proc of ICML '06, ACM, pp. 513–520 (2006).

[34] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. Darrell and N. D. Lawrence: "Topologically-constrained latent variable models", ICML, pp. 1080–1087 (2008).