Representing and Manipulating Mesh-based Character Animations

Edilson de Aguiar
DECOM / CEUNES / UFES
São Mateus, ES, Brazil
Email: edilson.de.aguiar@gmail.com

Norimichi Ukita
Nara Institute of Science and Technology
Nara, Japan
Email: ukita@is.naist.jp

***Abstract*—We propose a new 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 different performance capture approaches.**

***Keywords*-Animation; Character animations; Mesh animations; Mesh Editing;**

I. 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], [5], [6], [7], [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 [10], [2]. However, these editing methods are not able to preserve fine time-varying details during the manipulation process, as for instance the waving of the 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 different performance capture methods. As seen in Fig. 1 and in the results (Sect. VI), 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 integrated in the conventional animation pipeline.

The main contributions of our paper are:

- 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. II reviews the most relevant related work and Sect. III briefly describes our overall approach. Thereafter, Sect. IV details the method to convert a mesh-based character animation into the skeleton-based format and Sect. V describes how the time-varying details are learned using a non-linear probabilistic technique. Experiments and results are shown in Sect. VI and the paper concludes in Sect. VII.

II. Related Work

Creating animations for human subjects is still a time-consuming and expensive task. In the traditional framework,

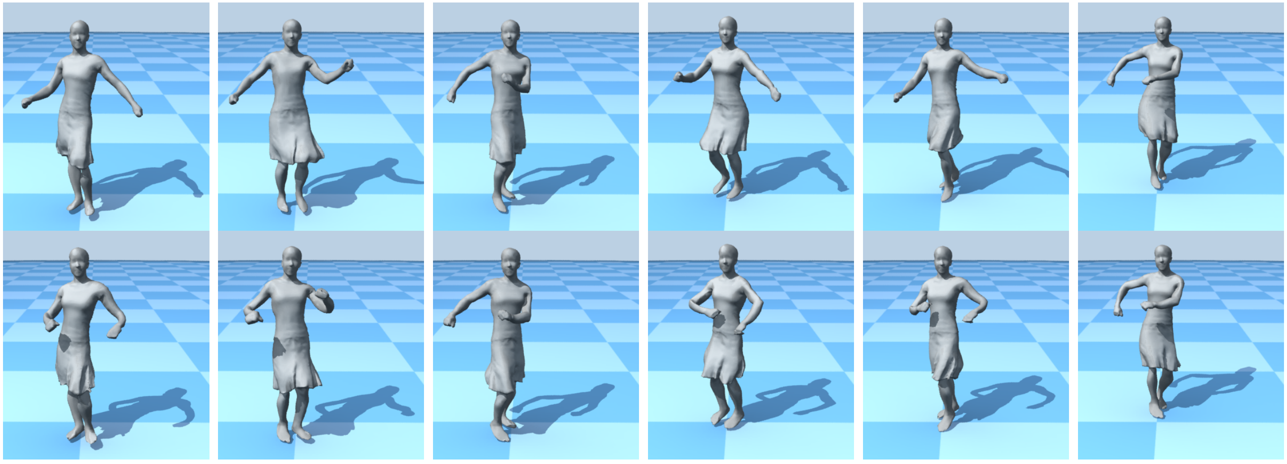


Fig. 1. Our approach represents an input mesh-based character animation (top row - particular frames) into a new hybrid representation that simplifies the editing process and preserves important time-varying details, i.e. dynamics of the skirt (bottom row - edited frames).

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], [14]. The skeleton model is animated by assigning motion parameters to the joints and the geometry and skeleton are connected via skinning (see [15] for an overview).

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 [16], [17], surface capture [18], [19] 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], [5], [6], [7], [8], but unfortunately these methods cannot 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).

Example-based skinning methods attempt to improve simple linear deformation by adding or correcting surface details from a given set of examples. In case the animation edits are not too large or complex, pose-space deformation [20], [21], weighted pose space deformations [22], and related papers would be able to provide reasonable results. Similar techniques have been developed for face animation [23], [24] as well. 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 [25], [26], [27], 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], [28]) 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 [29] or a shared latent structure [30]. These techniques were also used to model large dimensional data, such as silhouettes [31], voxel data [32] and even simple deformable models [33]. However, to the best of our knowledge, such technique has never been used to learn time-varying surface details for more complex models like in our system.

III. 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 [34], 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-temporally coherent meshes and it allows the manipulation of the entire animation by decomposing it into coarse (MCA_C) and fine (MCA_F)

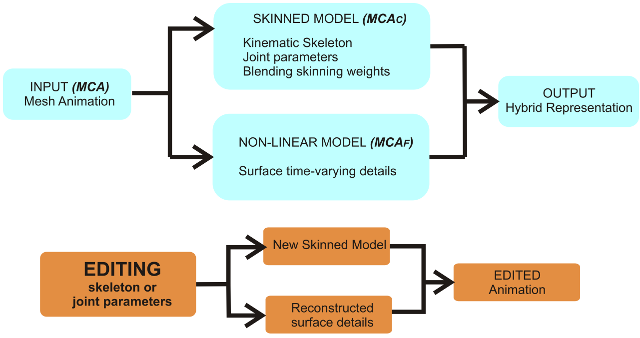


Fig. 2. Overview of our method: an input mesh-based character animation is decomposed into coarse (MCA_C) and fine (MCA_F) deformation components. This enables the user to edit the underlying skeleton-based representation and the time-varying details of the input are faithfully reconstructed in the final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

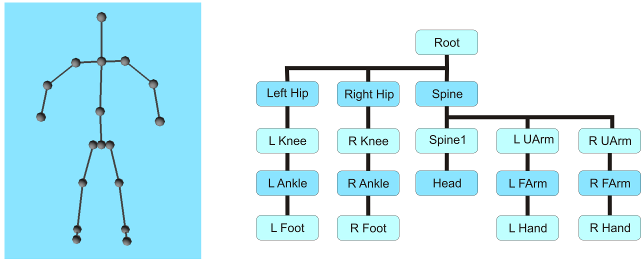


Fig. 3. Prior knowledge is incorporated in our framework by means of a known kinematic skeleton structure (left) containing 18 joints organized hierarchically (right).

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. IV.

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. V. 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. VI.

IV. 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 mesh 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 parametrize 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 input mesh 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 mesh 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 mesh pose for each frame. We perform this optimization for each joint subsequently following the skeleton's hierarchy. In contrast to [10], [2], 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 cannot 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.

V. Learning Time-varying Surface Details

We use a non-linear probabilistic technique to efficiently learn the surface 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. VI). 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) [30], via a low-dimensional latent space X, as illustrated in Fig. 4. In conjunction with the idea of Gaussian Process Dynamical Models (GPDM) [29], our latent space encoding is not only subject to the structure of the high dimensional data, but it is also subject to the dynamics in this data, which enforces smoothness of the temporal transitions of the latent variables. Our shared variant approach is used since we assume that the skeletal motion and the mesh details have a common underlying temporally coherent behavior. Skeletal pose Y_S and mesh details Y_M are related to the shared latent variables with a pair of forward mapping functions $f_S(\boldsymbol{x}) : X \rightarrow Y_S$ and $f_M(\boldsymbol{x}) : X \rightarrow Y_M$, where Y_S represents a D_S-dimensional joint parameter vector and Y_M is the D_M-dimensional time-varying detail vector.

The estimation of the mapping functions $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$ is briefly described in the following. In SGPLVM [30], 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 details ($\bar{\boldsymbol{y}}_{M_i}$ and $\bar{\boldsymbol{y}}_{M_j}$) and skeletal motion ($\bar{\boldsymbol{y}}_{S_i}$ and $\bar{\boldsymbol{y}}_{S_j}$). We use radial basis function (RBF) to define the non-linear kernel function and scaled conjugate gradient (SCG) for the optimization of $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$.

GPDM [29], 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 Y, and the second one is from a point at $t-1$ to a point at t in X, $f_D(\boldsymbol{x})$, as also 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 $\boldsymbol{X}_t$, respectively, 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 latent 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) [35] or averaging the top eigenvectors of the principal components [30], in our method, $\boldsymbol{X}$ is initialized by using only the principal components of $\boldsymbol{Y}_S$. Thereafter, 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. Dynamics constraint with 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

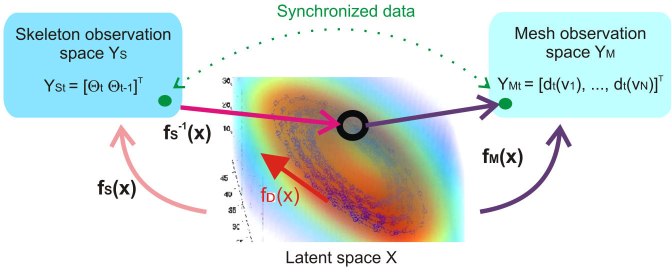


Fig. 4. The relation between joint parameters and surface details is learned using a shared latent space with dynamical constraints. Our model can generalize to different input configurations, as seen by the color-coded variance (blue=high $\rightarrow$ red=low).

joint angles are used for learning and that in our experiments, we achieved better results by discarding the joint angles for the root joint.

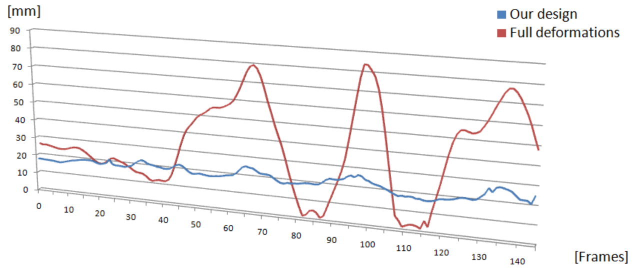
While the latent space X is optimized by embedding with Gaussian Process, a mapping function from $\boldsymbol{y}_S$ to $\boldsymbol{x}$ ($f^{-1}(\boldsymbol{y}_S) : \boldsymbol{Y}_S \rightarrow \boldsymbol{X}$) is not provided by the above mentioned process. In our work, after the latent space X is optimized, the mapping $\boldsymbol{Y}_S \rightarrow \boldsymbol{X}$ is obtained by a regression function, which is also learned by Gaussian Process [28].

Using $f_S(\boldsymbol{x})$ and $f_M(\boldsymbol{x})$, a new mesh model is generated as follows: first the coarse deformation $\boldsymbol{ps}(v_i)$ is estimated from the joint parameters at time t using our skinned model, Sect. IV. Thereafter, $\boldsymbol{y}_{S_t} = [\theta_t, \theta_{t-1}]^T$ is mapped to $\boldsymbol{y}_{M_t} = [\boldsymbol{d}_t(v_1), \cdots, \boldsymbol{d}_t(v_N)]^T$ 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 framework.

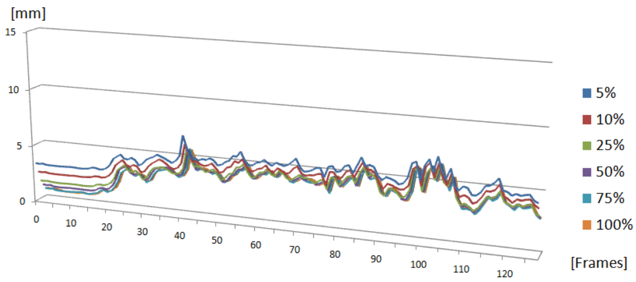
VI. Experiments and Results

Our approach has been tested on several mesh-based animation sequences generated from performance capture methods that are publicly available [36], [3]. The animations contain walking, marching 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.

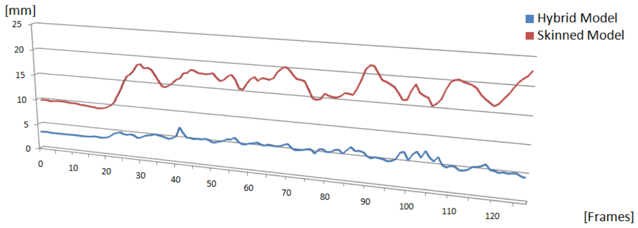
In our first experiment, we verified the efficiency of our system's design by comparing the performance of our non-linear probabilistic model to learn the full range of deformations in contrast to only encoding the time-varying details in Sect. V. By encoding coarse and fine deformations, our non-linear 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



(a)



(b)



(c)

Fig. 5. Experiments for the human-size samba sequence [3]: (a) The graph shows that our system design is able to reproduce the swing sequence more accurately. (b) A multiresolution approach can be used in our framework to deliver the same level of quality and decrease 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.

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. V.

Our second experiment was used to determine the best combination of representations to be applied to our GPLVM-based approach, Sect. V. Motion capture data can be represented by euler angles, quaternions or exponential maps. We applied all three representations to our method, and in our experiments exponential maps performed better. We also tested two common representations for positional data: vertex displacements in xyz space (*XYZ*) and differential coordinates [1] (*DIF*). In our experiments, both mesh representations give 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. $D_S = 2 \times 3 \times (N_{JOINTS} - 1)$ and $D_M = 3 \times N_{VERT}$), 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 procedure. 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 [36], 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 mesh resolution or 100% of the number of vertices, 75%, 50%, 25%, 10% and 5%. As seen in the graph in Fig. 5(b), 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% ($N_{VERT} = 350-500$). Please note that our system can still be applied to any resolution level and that for all sequences we tested, the time-varying details of the input animations were preserved during the process. We see this multiresolution scheme as an additional 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 [10], [2], 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 faithful reconstruction of the input (blue line in Fig. 5(c)). Fig. 6(e) also shows that our approach reproduces better the original dynamic details in the skirt in comparison with linear blending skinning. This property of our new representation is specially useful during the manipulation of the entire input animation.

The advantages of our hybrid representation are presented in Fig. 1, 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 realistically in the final edited animation. Fig. 6(a,b) shows a particular frame for several input sequences and the respective edited result using our hybrid representation. As presented in Fig. 6(c), using our framework, we are able to change the motion parameters of the underlying skeleton and generate convincing deformations for the skirt. We are also able to change the input skeleton dimensions, which enables us to even retarget the input animation to a different character proportion, Fig. 6(d).

The running time of our algorithm is dominated by the training phase of the GPLVM-based technique (around $30min$ for 100 frames). This step is done only once at the beginning for each sequence and, thereafter, the editing operations run in real-time. Our 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 cannot 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, the level of editing is limited and we are not able to generate details for motions that are too far from the original input. Scalability might become an issue as well. For general edits and motions the database will need to be larger. However, applying gaussian process methods to large data sets is challenging since it involves non-linear optimizations. We believe this issue can be minimized by adjusting the dimension of the latent space and the number of iterations during the optimization in Sect. V.

In constrast to related linear techniques [20], [21], [22], our representation allows a better generalization of the input, while suppressing the bad effects of noise. The use of a non-linear technique also enables us to learn not only the structure of the high dimensional data but also the dynamics present on it. For instance, one skeleton data at a moment might correspond to multiple mesh data (different shapes). Our approach uses GPDM that allows us to obtain a nice latent space where observation data (i.e. skeletons and meshes) with the same shape but different motions are mapped accordingly.

Currently, linear blending skinning was used to create the skinned model in Sect. IV, but we believe that similar results can be achieved with a more advanced skinning method [15] and we leave this for future work. We are using a basic GPLVM implementation, and improvements of this basic technique [37] can increase the performance of our method even further. Important parameters in Sect. V (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.

VII. Conclusions

We presented a 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.

Acknowledgment

The authors would like to thank FAPES (Espirito Santo Research Foundation Agency) for the financial support of this research.

References

[1] M. Botsch and O. Sorkine, "On linear variational surface deformation methods," *IEEE TVCG*, vol. 14, no. 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*, vol. 27, no. 2, pp. 389–397, 2008.
[3] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM TOG*, vol. 27, no. 3, p. 97, 2008.
[4] S. Kircher and M. Garland, "Editing arbitrarily deforming surface animations," in *SIGGRAPH '06*, 2006, pp. 1098–1107.
[5] W. Xu, K. Zhou, Y. Yu, Q. Tan, Q. Peng, and B. Guo, "Gradient domain editing of deforming mesh sequences," *ACM TOG*, vol. 26, no. 3, p. 84, 2007.
[6] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," in *SIGGRAPH '07*. ACM, 2007, p. 80.
[7] S. Kircher and M. Garland, "Free-form motion processing," *ACM TOG*, vol. 27, no. 2, pp. 1–13, 2008.
[8] I. Baran, D. Vlasic, E. Grinspun, and J. Popović, "Semantic deformation transfer," *ACM TOG*, vol. 28, no. 3, p. 36, 2009.
[9] D. James and C. Twigg, "Skinning mesh animations," *ACM TOG*, vol. 24, no. 3, 2005.
[10] S. Schaefer and C. Yuksel, "Example-based skeleton extraction," in *SGP '07*, 2007, pp. 153–162.
[11] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 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," in *Proc. of CVPR '05*. IEEE, 2005, pp. 782–788.
[13] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Trans. Graph.*, vol. 26, no. 3, p. 72, 2007.
[14] G. Bharaj, T. Thormählen, H.-P. Seidel, and C. Theobalt, "Automatically rigging multi-component characters," *Comp. Graph. Forum (Proc. Eurographics 2012)*, vol. 30, no. 2, 2011.
[15] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Geometric skinning with approximate dual quaternion blending," *ACM TOG.*, vol. 27, no. 4, pp. 1–23, 2008.
[16] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," in *Proc. of SIGGRAPH '04*. ACM, 2004, pp. 399–405.
[17] M. Ben-Chen, O. Weber, and C. Gotsman, "Spatial deformation transfer," in *Proc. of SCA*. ACM, 2009, pp. 67–74.
[18] J. Starck and A. Hilton, "Surface capture for performance based animation," *IEEE Computer Graphics and Applications*, vol. 27(3), pp. 21–31, 2007.
[19] P. Huang, A. Hilton, and J. Starck, "Human motion synthesis from 3d video," in *Proc. of CVPR '09*. IEEE, 2009.
[20] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 165–172.

[21] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman, “Context-aware skeletal shape deformation,” *Comput. Graph. Forum*, vol. 26, no. 3, pp. 265–274, 2007.
[22] T. Kurihara and N. Miyata, “Modeling deformable human hands from medical images,” in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer animation*, ser. SCA ’04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 355–363.
[23] B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross, “Pose-space animation and transfer of facial details,” in *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA ’08. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 57–66.
[24] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec, “Facial performance synthesis using deformation-driven polynomial displacement maps,” *ACM Trans. Graph.*, vol. 27, no. 5, pp. 121:1–121:10, 2008.
[25] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, “Scape: shape completion and animation of people,” *ACM TOG*, vol. 24, no. 3, pp. 408–416, 2005.
[26] 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,” in *Proc SCA’06*. Eurographics Association, 2006, pp. 147–156.
[27] S. I. Park and J. K. Hodgins, “Data-driven modeling of skin and muscle deformation,” *ACM TOG*, vol. 27, no. 3, pp. 1–6, 2008.
[28] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
[29] J. M. W., D. J. Fleet, and A. Hertzmann, “Gaussian process dynamical models for human motion,” *IEEE PAMI*, vol. 30, no. 2, pp. 283–298, 2008.
[30] A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao, “Learning shared latent structure for image synthesis and robotic imitation,” in *Proc. of NIPS’05*, 2005.
[31] R. Urtasun and T. Darrell, “Sparse probabilistic regression for activity-independent human pose inference,” in *CVPR*, 2008.
[32] N. Ukita, M. Hirai, and M. Kidode, “Complex volume and pose tracking with probabilistic dynamical models and visual hull constraints,” in *IEEE ICCV’09*, 2009.
[33] M. G. Salzmann and P. Urtasun, R.and Fua, “Local deformation models for monocular 3d shape recovery,” in *Proc. of CVPR ’08*. IEEE, 2008.
[34] M. Botsch and L. Kobbelt, “Multiresolution surface representation based on displacement volumes,” *Computer Graphics Forum*, vol. 22, no. 3, pp. 483–491, 2003.
[35] C. H. Ek, P. H. S. Torr, and N. D. Lawrence, “Gaussian process latent variable models for human pose estimation,” in *International Workshop on Machine Learning for Multimodal Interaction*, 2007.
[36] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, “Performance capture from sparse multi-view video,” in *ACM TOG*, 2008.
[37] A. Geiger, R. Urtasun, and T. Darrell, “Rank priors for continuous non-linear dimensionality reduction,” in *Proc. of CVPR ’09*. IEEE, 2009.

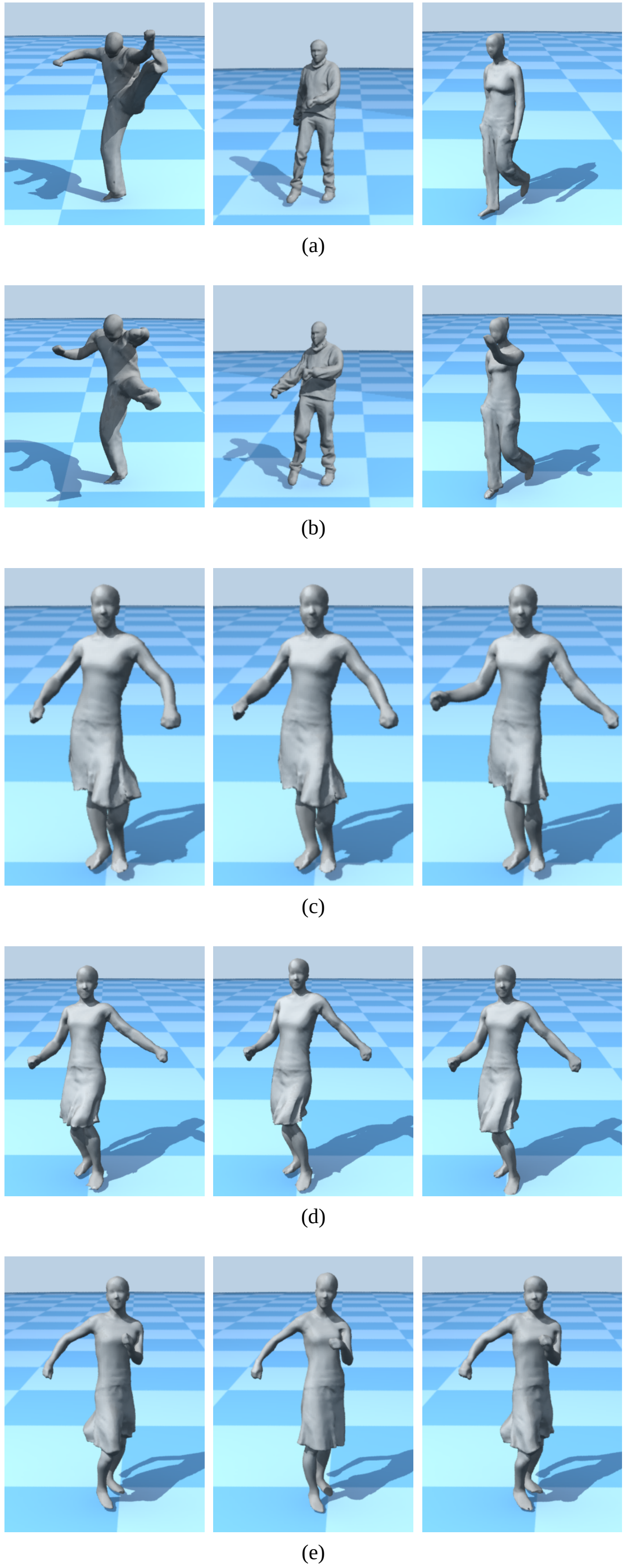


Fig. 6. Editing mesh-based character animations: input frames (a) and edited results (b). Our framework allows a more faithful manipulation of the fighting, marching and walking sequences, respectively. (c) For a single frame, different surface details for the skirt can be reconstructed based on the underlying skeletal motion. (d) We are also able to modify the proportions of the input mesh model (left), simplifying the retargeting of a mesh animation (middle=changing torso and right=changing legs). (e) Comparison between original frame (left), reconstructed frame using our skeleton-based representation (i.e. linear blending skinning) (middle) and reconstruction using our full approach. Our method reproduces better the details of the dynamics in the skirt from the original frame.