Example-based Body Model Optimization and Skinning

Philipp Fechteler, Anna Hilsmann and Peter Eisert

Visual Computing, HU Berlin, Germany
Vision & Imaging Technologies Dep., Fraunhofer HHI, Germany

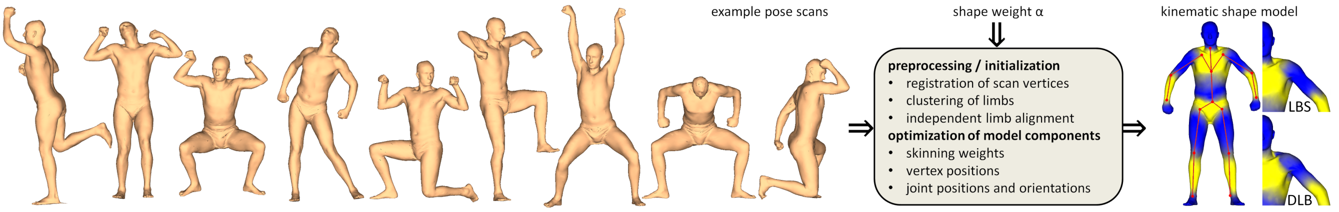

Figure 1: *Schematic overview of the kinematic shape model optimization framework.*

Abstract

In this paper, we present an example-based framework for the generation of a realistic kinematic 3D human body model that optimizes shape, pose and skinning parameters. For enhanced realism, the skinning is realized as a combination of Linear Blend Skinning (LBS) and Dual quaternion Linear Blending (DLB) which nicely compensates the deficiencies of using only one of these approaches (e.g. candy wrapper, bulging artifacts) and supports interpolation of more than two joint transformations. The optimization framework enforces two objectives: resembling both shape and pose as closely as possible by iteratively minimizing the objective function with respect to (a) the vertices, (b) the skinning weights and (c) the joint parameters. Smoothness is ensured by using a weighted Laplacian besides a typical data term in the objective function, which introduces the only parameter to be specified. With experimental results on publicly available datasets we demonstrate the effectiveness of the resulting shape model, exposing convincing naturalism. By using examples for the optimization of all parameters, our framework is easy to use and does not require sophisticated parameter tuning or user intervention.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Animatable Computer Graphics models of humans are omnipresent in todays society, e.g. in computer games, movie productions, telecommunications, human-computer interfaces. While very realistic body modeling techniques exist, these usually need a lot of manual effort and fine tuning. The aforementioned applications, however, raise the demand for approaches that are easy-to-use while at the same time producing sufficient quality. The scientific literature of recent years reveals that a lot of effort has been and still is put into this topic. An overview of recent work can be found in [JDKL14].

Due to its computational and analytical simplicity, Linear Blend Skinning (LBS) is a widely used skinning technique, despite its well known visual artifacts (candy wrapper, edge collapse etc). Various approaches to enhance the realism have been undertaken. One direction is to keep the compact skinning function structure based on rigid bone transformations by using non-linear interpolation functions, multiple linear ones or combinations of them [KCO09, KCvO08, KS12]. Further increase in realism is achieved by modeling also subject specific shape deformations [ASK*05, HSS*09, PMRMB15] at the cost of drastically increased complexity and loosing the benefits of a compact skinning function. On the other side, the realism of compact skinning functions can be increased by optimizing the skinning parameters to certain objectives [BP07, LD12, CLC*13, HTRS10, LD14], in contrast to let an artist model a character by hand. Usually, the objective function is chosen to reflect predefined or artificial assumptions on the skeleton structure and deformation behavior, e.g. smoothness, elasticity etc.

In this paper, we present a method for automatic generation of animatable models with realistic deformation behavior. Our work

DOI: 10.2312/egsh.20161001

aims at a compact representation (in comparison to those exploiting databases, e.g. Pose Space approaches) which possesses visual realism, i.e. animated movements should look as natural as possible. Additionally, we aim for a model generation process that is simple to use, without sophisticated parameter tuning or any manual intervention. Instead of using assumptions or artificial models for the optimization, we use a set of examples and optimize shape, pose and skinning parameters to fit examples as closely as possible. For increased realism, we present a generalization of the skinning function presented in [KS12] which allows realistic blending of more than two joint transformations. A schematic overview is shown in Fig. 1.

To sum up, the contribution of this paper is two-fold: (a) we present an easy-to-use and completely data-driven optimization framework that optimizes all components of an enhanced (b) skinning function.

2. Skinning Function

The choice of the skinning function is crucial for the quality of the resulting animation realism. In [KS12], a compact skinning function is presented. This skinning function combines Linear Blend Skinning (LBS) and Dual Quaternion Skinning (DQS) in order to compensate for their deficiencies (Candy-wrapper, bulging artifacts etc.). LBS is used to model swing motion and DQS is used for twist motions. This approach provides convincing animation results. However, it is limited to blending only two joint transformations, e.g. two consecutive joints.

The skinning function we use in this work is enhanced by combining LBS with Dual quaternion Linear Blending (DLB) [KCvO08]. The usage of DLB instead of DQS allows the interpolation between more than two joints. This is important in cases where the kinematic dependency chain is split into multiple branches, e.g. legs, arms and head. Since DLB approximates closely the characteristics of DQS for two joints, our skinning function produces visually indistinguishable results for these cases as in [KS12] (especially because we optimize all components), but supports further modeling flexibility.

In order to transform a vertex v of the model from its rest pose to a certain target pose (e.g. for a specific example scan), each unit dual quaternion representing the rotation with respect to the joint's local coordinate system is decomposed into its twist and swing components. According to DLB, a vertex specific twist transformation is calculated by adding up the unit dual quaternions representing the twist components weighted with their vertex specific skinning weight for twisting, followed by a normalization step. After applying the combined twist transformation, standard LBS is applied using the unit dual quaternions representing the swing components of the joints.

The complete pose aligning transformation $\mathcal{T}()$ uses this DLB/LBS-based skinning interpolation $\mathcal{S}()$, followed by the rigid transformation $\mathcal{K}()$ representing the transformations of all previous joints in the kinematic chain, followed by global rigid alignment $\mathcal{G}()$.

$$v' = \mathcal{T}(v) = \mathcal{G}(\mathcal{K}(\mathcal{S}(v)))$$

3. Preprocessing

The optimization framework we present in Section 4 optimizes a shape model to resemble the given training data as closely as possible. The training data consists of a set of registered 3D scans of an actor in different poses. There are various approaches available for the registration of 3D scans. The datasets we use are [ASK*05] and [BHKH13].

The shape model to be optimized consists of:

3D mesh: a set of vertices with a defined topology, sometimes called bind shape - initially set randomly to any of the training set poses

Skinning weights: two sets of float values for each vertex (one for LBS and one for DLB) - initially set to binary weights corresponding to the result of a k-means clustering as presented in [LD12] (rigid kinematics without any smooth interpolation)

Joint positions and orientations: a set of 3-vectors and unit quaternions - initially set to the intersections of the limbs resulting from the k-means clustering and local coordinate systems with the x-axis pointing to the next joint in the kinematic chain

Additionally, the optimization framework needs as input the initial configuration for each joint as well as the global alignment (rotation and translation) for each example scan. This is addressed by treating each joint/limb of each scan independently as an Orthogonal Procrustes alignment problem [Akc03] which is solved efficiently via Singular Value Decomposition.

4. Optimization

In the optimization we enforce that the model should resemble as close as possible the example scans after skinning-transformation. This can be refined into two guiding principles:

Conforming pose: the transformed model should resemble the targeted scan pose by having minimal distances between corresponding vertices

Conforming shape: the surface of the transformed model should resemble the surface of the targeted scan pose by having similar vertex positions relative to their neighbors

For S example scans with V vertices these two principles are transformed into an objective function consisting of two terms: a typical data term and a weighted mesh Laplacian [Sor06]:

$$Q = \sum_{\substack{i=1\ldots S\\ j=1\ldots V}} \left| v_j^{\text{scan}_i} - \mathcal{T}(v_j, i) \right|^2 + \alpha \left| \mathcal{L}\left(v_j^{\text{scan}_i}\right) - \mathcal{L}\left(\mathcal{T}(v_j, i)\right) \right|^2$$

with $\alpha \geq 0$ being the shape weight parameter, $v_j^{\text{scan}_i}$ being the j^{th} vertex of the i^{th} example scan, $\mathcal{T}(v_j, i)$ being the j^{th} vertex transformed with the transformation parameters for the i^{th} example scan and $\mathcal{L}()$ being the uniform Laplacian function.

This objective function is sufficient to generalize the skinning model parameters in order to generate realistically appearing movements as demonstrated in Section 5. The minimization of this objective function is grouped into three optimization steps that are iterated until convergence: Skinning weights, vertex locations and joint locations and orientations.

Optimization of skinning weights via coordinate descent
For each vertex of the model, the DLB/LBS-based skinning interpolation $\mathcal{S}()$ requires two sets of weights: one for DLB, and one for LBS. Each of the two sets w_{LBS} and w_{DLB} contains a weight for each joint in the range of 0..1, specifying the influence of the joint's transformation on the vertex.

The optimization of the skinning weights with respect to Q is achieved by employing the coordinate descent algorithm [JDKL14]. This algorithm operates on a currently valid configuration and updates only one variable at a time. Applied to our skinning weight optimization problem, we generate several slightly varied skinning weight candidates for each vertex and apply the update which results in the best gain with respect to the objective Q. This is repeated until convergence. The gain for a skinning weight candidate is calculated/updated efficiently by exploiting the fact that changes in skinning weights affect only the vertex's and its neighbors' contribution to the objective Q.

Optimization of vertex locations via least squares fitting
The optimal location of the model's vertices with respect to the objective Q is calculated directly by setting up a suitable linear equation system and solving for the least squares solution. With all components fixed to constant values, the complete pose alignment transformation $\mathcal{T}()$ can be brought into the form of an invertible transformation, consisting of a $3{\times}3$-matrix $T_{\mathcal{T}}$ and a 3-vector $t_{\mathcal{T}}$:

$$v' = \mathcal{T}(v) = T_{\mathcal{T}}v + t_{\mathcal{T}}$$

Note, that as a consequence of LBS $T_{\mathcal{T}}$ is not a rotation matrix. This reformulation in turn allows to stack the equations for all vertices of all example scan poses into one equation system matrix. Following the same principle, the equations for the Laplacian term of the objective Q can be appended to the equation system matrix. The model's vertex positions which minimize objective Q are calculated directly by using a sparse solver like the PARDISO solver from the Intel MKL library.

Optimization of joints via Gauss-Newton algorithm
The orientation and location of the model's joints are jointly optimized with the actual pose parameters (joint angles & global rigid alignment) for all example scans by employing the Gauss-Newton algorithm. For this purpose, we calculate the partial derivatives of the objective Q with respect for these variables at the current configuration. In order to calculate valid rotations conforming to SO(3), we use the exponential map representation of rotations in the same way as in [Ude98]. Solving for the optimal parameter updates can be significantly accelerated by exploiting that the pose parameters for a certain example scan are independent to all other parameters. Because of the highly non-linear nature of the objective Q we include a simple line search to determine the best step width while updating the parameters.

Figure 2: *Optimized shape model with (left) and without (right) optimizing the joint parameters, and the corresponding input target pose scan (middle).*

5. Experimental results

Throughout all our experiments we used a shape weight of $\alpha = 100$. Lower values turned out to produce a non-smooth surface of the resulting shape model and bigger values increased the computation time for skinning weights, but not the quality of the result.

Since optimization of the joints are rarely considered in the literature, we compare in Fig. 2 the results of optimizing only the shape model's vertices and skinning weights versus also the joint's location and orientation. It is clearly visible that the result with optimized joints looks much more realistic and resembles much better the targeted input scan pose.

We have evaluated our optimization framework together with the enhanced skinning function with the publicly available datasets SCAPE [ASK*05] (71 example poses with 12k vertices and 25k

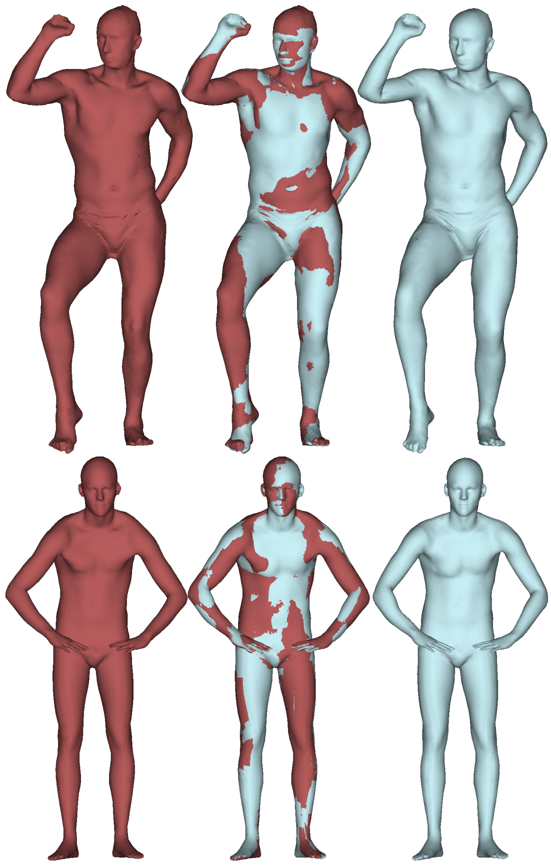

Figure 3: *Example results from the optimization framework for SCAPE (upper row) and MPI Faust (lower row), with target/input mesh (left) and result/model mesh (right), and both on top of each other (middle).*

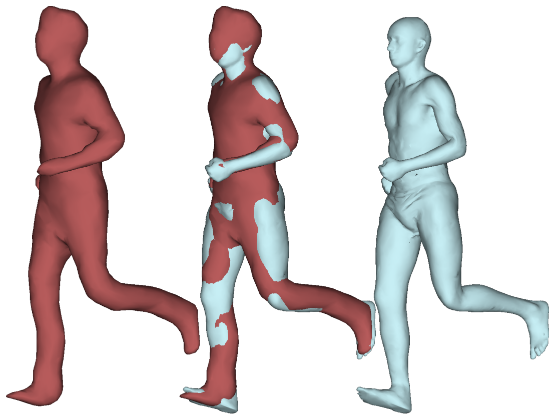

Figure 4: *Example frame showing the SCAPE optimized shape model (right) animated to the CVSSP dataset (left) using motion transfer techniques and both put on top of each other (middle) for better visualization of the matching accuracy.*

triangles) and MPI Faust [BRLB14] (10 example poses with ∼7k vertices, ∼14k triangles). Examples are shown in Fig. 3. It is clearly visible that the optimized shape model closely resembles the targeted example pose from the input training set. Additionally, we encourage the reader to take a look at the accompanied video to see the animated results with interpolated poses.

In order to demonstrate the natural appearance of animating the optimized shape model, we extracted suitable animation parameters for the shape model learned from the SCAPE dataset from the publicly available CVSSP Performance Capture sequence [BHKH13] (416 frames, 2667 vertices, 5330 triangles, captured at 25 FPS). The animation parameter extraction is done by fitting the kinematic shape model first to a CVSSP frame of similar pose. Then, one frame after another is added to the alignment process, which optimizes the pose parameters independently for each frame, based on one common set of vertex correspondences. A sample frame is shown in Fig. 4 and the full sequence is shown in the accompanied video. This result video demonstrates the natural appearance of the resulting animation.

6. Conclusions

In this paper we presented an optimization framework to automatically generate 3D shape models of human actors from examples. We target at generating a compact model with visual realism during animation and an easy-to-use model creation method.

The framework iterates in an optimization loop over the optimization of all required shape model components: vertices, skinning weights and joint parameters. A mesh Laplacian enforces that the shape of the resulting model closely resembles the shape of the input example scans.

Realistic skinning is realized by a combination of Dual quaternion Linear Blending (DLB) with Linear Blend Skinning (LBS), which nicely compensates for typical skinning artifacts and allows blending of more than two joint transformations.

We evaluate the quality of the resulting optimized shape model on two publicly available datasets: SCAPE and MPI Faust. Additionally, we demonstrate the natural appearance of the resulting animation of the optimized shape model, by using motion transfer techniques on the publicly available Performance Capture sequence CVSSP to extract suitable animation parameters, and applying them to our optimized shape model.

7. Acknowledgments

The research that lead to this paper was supported in part by the European Commission under contract FP7-ICT-611761 ActionTV and H2020-ICT-687757 REPLICATE.

References

[Akc03] Akca D.: *Generalized Procrustes Analysis and its applications in Photogrammetry*. Tech. rep., ETH Zurich, 2003. 2

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: SCAPE: Shape Completion and Animation of People. *Proc. ACM SIGGRAPH* (2005). 1, 2, 3

[BHKH13] Budd C., Huang P., Klaudiny M., Hilton A.: Global Non-rigid Alignment of Surface Sequences. *Int. Journal of Computer Vision (IJCV)* (2013). 2, 4

[BP07] Baran I., Popovic J.: Automatic Rigging and Animation of 3D Characters. In *Proc. ACM SIGGRAPH* (2007). 1

[BRLB14] Bogo F., Romero J., Loper M., Black M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. IEEE CVPR* (2014). 4

[CLC*13] Chen Y., Lai Y.-K., Cheng Z.-Q., Martin R. R., Jin S.-Y.: A Data-Driven Approach to Efficient Character Articulation. In *Proc. Int. Conf. on Computer-Aided Design and Computer Graphics* (2013). 1

[HSS*09] Hasler N., Stoll C., Sunkel M., Rosenhahn B., Seidel H.-P.: A Statistical Model of Human Pose and Body Shape. In *Proc. EUROGRAPHICS* (2009). 1

[HTRS10] Hasler N., Thormählen T., Rosenhahn B., Seidel H.-P.: Learning skeletons for shape and pose. *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2010). 1

[JDKL14] Jacobson A., Deng Z., Kavan L., Lewis J.: Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH Courses* (2014). 1, 3

[KCO09] Kavan L., Collins S., O'Sullivan C.: Automatic Linearization of Nonlinear Skinning. In *Proc. Symposium on Interactive 3D Graphics and Games* (2009). 1

[KCvO08] Kavan L., Collins S., Žára J., O'Sullivan C.: Geometric Skinning with Approximate Dual Quaternion Blending. *Proc. ACM SIGGRAPH* (2008). 1, 2

[KS12] Kavan L., Sorkine O.: Elasticity-Inspired Deformers for Character Articulation. *Proc. ACM SIGGRAPH ASIA* (2012). 1, 2

[LD12] Le B., Deng Z.: Smooth Skinning Decomposition with Rigid Bones. *Proc. ACM SIGGRAPH* (2012). 1, 2

[LD14] Le B. H., Deng Z.: Robust and accurate skeletal rigging from mesh sequences. *Proc. ACM SIGGRAPH* (2014). 1

[PMRMB15] Pons-Moll G., Romero J., Mahmood N., Black M. J.: Dyna: A Model of Dynamic Human Shape in Motion. *Proc. ACM SIGGRAPH* (2015). 1

[Sor06] Sorkine O.: Differential Representations for Mesh Processing. *Computer Graphics Forum* (2006). 2

[Ude98] Ude A.: Nonlinear Least Squares Optimisation of Unit Quaternion Functions for Pose Estimation from Corresponding Features. In *Proc. Int. Conf. on Pattern Recognition* (1998). 3