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

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 compensate 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 nonlinear interpolation functions, multiple linear ones or combinations of them [KCC09, Kcv08, 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...
aims at a compact representation (in comparison to those exploit-
databases, e.g. Pose Space approaches) which possesses visual
realism, i.e. animated movements should look as natural as pos-
sible. Additionally, we aim for a model generation process that is
to simple to use, without sophisticated parameter tuning or any man-
tual 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 possi-
ble. For increased realism, we present a generalization of the skin-
ning function presented in [KS12] which allows realistic blending
or 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) skin-
ning 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 func-
tion 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 re-
sults. However, it is limited to blending only two joint transfor-
mations, 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 multi-
ple branches, e.g. legs, arms and head. Since DLB approximates
closely the characteristics of DQS for two joints, our skinning func-
tion 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 transform-
ation is calculated by adding up the unit dual quaternions rep-
resenting the twist components weighted with their vertex specific
skinning weight for twisting, followed by a normalization step. Af-
fter applying the combined twist transformation, standard LBS is
applied using the unit dual quaternions representing the swing com-
ponents of the joints.

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

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

3. Preprocessing
The optimization framework we present in Section 4 optimizes a
shape model to resemble the given training data as closely as pos-
sible. 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 [BHKK13].

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 cor-
responding 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 result-
ing 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 treat-
ing 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 corre-
sponding 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 trans-
formed into an objective function consisting of two terms: a typical
data term and a weighted mesh Laplacian [Sor06]:

\[
Q = \sum_{j=1,...,S} \left| v_{j}^{\text{scăn}} - T(v_{j}, i) \right|^2 + \alpha \mathcal{L} \left( v_{j}^{\text{scăn}} - \mathcal{L}(T(v_{j}, i)) \right)^2
\]

with \( \alpha \geq 0 \) being the shape weight parameter, \( v_{j}^{\text{scăn}} \) being the \( j^{\text{th}} \)
vertex of the \( j^{\text{th}} \) example scan, \( T(v_{j}, i) \) being the \( j^{\text{th}} \) vertex trans-
formed with the transformation parameters for the \( j^{\text{th}} \) example scan
and \( \mathcal{L}(\cdot) \) being the uniform Laplacian function.

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

© 2016 The Author(s)

Optimization of skinning weights via coordinate descent
For each vertex of the model, the DLB/LBS-based skinning inter-
polation $S()$ requires two sets of weights: one for DLB, and one
for LBS. Each of the two sets $\mathbb{W}_{\text{DLB}}$ and $\mathbb{W}_{\text{LBS}}$ 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 algo-
rithm [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 up-
date 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 $T()$ can be brought into the form of an invertible
transformation, consisting of a $3\times3$-matrix $T_T$ and a $3$-vector $t_T$:

\[
\mathbf{v'} = T(\mathbf{v}) = T_T \mathbf{v} + t_T
\]

Note, that as a consequence of LBS $T_T$ is not a rotation matrix.
This reformulation in turn allows to stack the equations for all ver-
tices 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 cal-
culated 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 opti-
mized 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 up-
dates 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 objec-
tive $Q$ we include a simple line search to determine the best step
width while updating the parameters.

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 lit-
erature, 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 op-
timized 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 [ASK05] (71 example poses with 12k vertices and 25k

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

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).
of more than two joint transformations. This nicely compensates for typical skinning artifacts and allows Linear Blending (DLB) with Linear Blend Skinning (LBS), which closely resembles the shape of the resulting model.

We target at generating a compact model with visual realism during the 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


© 2016 The Author(s)