Real-time Skeletal Skinning with Optimized Centers of Rotation

Binh Huy Le* Jessica K. Hodgins†

Disney Research

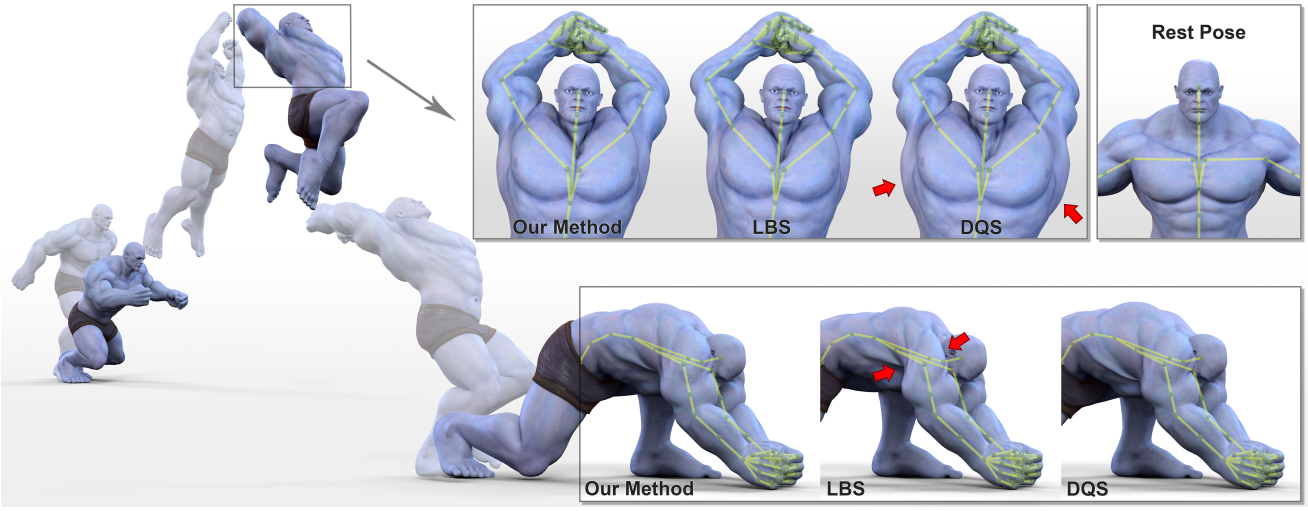


Figure 1: *The deformations created by our new direct skinning method on several key poses of an animation. With the same setup, our method reduces the bulging artifact of dual quaternion skinning (DQS) on the chest and the candy wrapper artifact of linear blend skinning (LBS) on the shoulder of the model. The artifacts are indicated by red arrows.*

Abstract

Skinning algorithms that work across a broad range of character designs and poses are crucial to creating compelling animations. Currently, linear blend skinning (LBS) and dual quaternion skinning (DQS) are the most widely used, especially for real-time applications. Both techniques are efficient to compute and are effective for many purposes. However, they also have many well-known artifacts, such as collapsing elbows, candy wrapper twists, and bulging around the joints. Due to the popularity of LBS and DQS, it would be of great benefit to reduce these artifacts without changing the animation pipeline or increasing the computational cost significantly. In this paper, we introduce a new direct skinning method that addresses this problem. Our key idea is to pre-compute the optimized center of rotation for each vertex from the rest pose and skinning weights. At runtime, these centers of rotation are used to interpolate the rigid transformation for each vertex. Compared to other direct skinning methods, our method significantly reduces the artifacts of LBS and DQS while maintaining real-time performance and backwards compatibility with the animation pipeline.

Keywords: skinning, skeletal animation, blending, real-time animation

Concepts: •**Computing methodologies** → **Animation;**

*e-mail: binh.le@disneyresearch.com

†e-mail: jkh@disneyresearch.com

SIGGRAPH '16 Technical Paper,, July 24 - 28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925959

1 Introduction

Good skinning algorithms are of fundamental importance for animating characters, such as humans or animals. Skinning methods must generate high-quality, detailed deformations around the joints of the characters from a small set of controllers. Among the many available skinning methods, skeletal-based algorithms are the most widely used. Generally, they employ a simplified version of the anatomical skeleton as the controller. The character is deformed by rotating the bones around its joints. Each bone propagates its transformation to the limb that it supports. The skin deformation typically appears near the joints where an area of the surface is influenced by the transformations of two or more bones.

Two skeletal-based models, *linear blend skinning* (LBS) [Magnenat-Thalmann et al. 1988] and *dual quaternion skinning* (DQS) [Kavan et al. 2008], are implemented in most game engines, virtual reality engines, and 3D animation software. Both LBS and DQS are direct methods with closed-form solutions where each vertex transformation is computed as a weighted-blend of bone transformations. Both LBS and DQS only require users to provide per-vertex skinning weights and per-frame bone transformations. These inputs are intuitive and users can easily manipulate the weights using painting tools. While the calculations of the deformations with LBS and DQS are different, their implementations are very similar and the most effective ways of calculating the vertex transformations utilize GPU vertex shaders. Due to this simplicity and efficiency, LBS

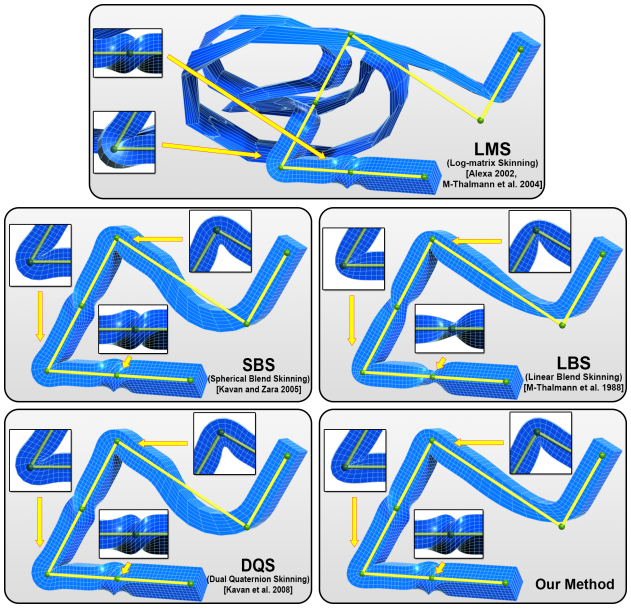


Figure 2: *This toy example demonstrates the deformation quality of different skinning methods for bending and twisting. In clockwise order, the five joint manipulations are:* 135° *twisting,* 120° *bending,* 90° *twisting,* 90° *bending, and* 90° *bending with translation. Compared to previous skinning methods, our method offers deformations without collapsing or bulging artifacts.*

and DQS are the de-facto standard for real-time applications and even for off-line rendering, where performance is less critical. LBS and DQS can still play an important role as the foundation for high-quality script-based rigging systems [Autodesk 2016], corrective methods [Kry et al. 2002; Vaillant et al. 2013; Vaillant et al. 2014], or reduced-space physical simulators [Müller and Chentanez 2011].

Although LBS and DQS offer great performance, they also create artifacts in deformation quality, particularly for certain poses. Two well-known artifacts of LBS are *elbow collapse* (at the bent joints in Fig. 2) and *candy wrapper* (at the twisted joints in Fig. 2) due to the volume loss caused by the linear blending. DQS eliminates these two artifacts, but it introduces a *joint bulging artifact* (at bent joints in Fig. 2) because DQS interpolates rotations around the center of rotation at the joints [Kim and Han 2014].

In this paper, we provide a solution to the skinning problem that avoids these artifacts and can be a drop-in replacement to LBS or DQS in a standard animation pipeline. Our approach relies on two assumptions: the local transformations should be rigid (orthogonal), and vertices with similar skinning weights should follow similar transformations. We define the desired deformation as a correction of the LBS result where we fit the best rigid transformation that brings vertices with similar skinning weights from the rest pose to the LBS-deformed pose.

Contributions. Our analysis shows that we can simplify the bone transformations in the equations so that the most computationally expensive step, i.e., finding the optimum *center of rotation* (CoR) for each vertex, can be precomputed from the rest pose and skinning weights, without knowledge of the bone transformations for each frame of the animation. As a result, we can cache the CoRs and directly calculate per-vertex transformations at runtime. Compared to previous direct skinning methods [Magnenat-Thalmann et al. 1988; Kavan and Žára 2005; Alexa 2002; Magnenat-Thalmann et al. 2004; Kavan et al. 2008], our method offers three main advantages:

- **Deformation Quality.** Our method imposes orthogonal constraints to prevent the elbow collapsing and candy wrapper artifacts of the LBS model. Our method also reduces the joint-bulging artifact of DQS by estimating an optimized CoR for each vertex. In Fig. 2, we show the improvement in deformation quality over other direct skinning methods.
- **Backward Compatibility.** Our model uses the same setup as other skeletal-based skinning models, including LBS and DQS: the rest pose, the skinning weights, and the bone transformations. Therefore, it can be seamlessly integrated into existing animation pipelines.
- **Performance.** Our method can fully utilize current graphics hardware (GPUs). With the cached CoRs, we can directly compute the deformed positions in the vertex shaders, similar to LBS and DQS. In particular, our runtime implementation is simply one LBS step on CoRs followed by one quaternion blending step (Section 3.2).

2 Related Work

We review the literature on skinning by considering three categories: geometric skinning, physical simulation and data-driven approaches. The approach described in this paper falls into the category of direct methods for geometric skinning, so we focus primarily on those methods.

Geometric skinning deforms a surface representation of the model to best represent the current pose of the character. These representations facilitate simple, high performance algorithms that are suitable for real-time applications. Geometric skinning methods can be further divided into two categories: *direct methods* and *indirect methods*.

Direct methods explicitly compute the deformed position of each vertex from control parameters and a rest configuration of the model. The first direct skinning model is known by many names: *joint-dependent local deformation*, *skeleton subspace deformation*, or *linear blend skinning* (LBS). It is credited to Magnenat-Thalmann and colleagues [1988]. In their paper, they introduced two important concepts: (1) *bones* which support the nearly rigid parts of the model and whose rigid transformations are used to control the animation, and (2) *skinning weights* which generate smooth deformations by blending bone transformations and then applying the result to each vertex. In the original work, the simple linear blending of 4×4 transformation matrices generates noticeable volume loss artifacts. These have come to be known as the elbow collapsing artifact (when joints are bent by almost 180°) and the candy wrapper artifact (when joints are twisted by a significant rotation) (Fig. 2).

Non-linear techniques, such as *log-matrix skinning* (LMS) [Alexa 2002; Magnenat-Thalmann et al. 2004], *spherical blend skinning* (SBS) [Kavan and Žára 2005], or *dual quaternion skinning* (DQS) [Kavan et al. 2008], prevent volume loss by blending transformations in the orthogonal space but create bulges near bent joints (Fig. 2). This bulging artifact occurs because all vertices near a joint share the same center of rotation (CoR), which is located exactly at the joint [Kim and Han 2014]. In contrast, we compute different CoRs for different vertices. Fig. 7 shows CoRs correponding to the toy example in Fig. 2. In this simple example, our CoRs behave similarly to the *curve skeleton* [Yang et al. 2006; Forstmann et al.

2007; Öztireli et al. 2013] without requiring the specific shape of the skeleton.

One common approach to improving the deformation quality of direct methods is to add more dimensions. Adding extra bones (helper bones) at joints reduces volume loss by dividing large twist/bend angles into smaller angles across multiple bones [Mohr and Gleicher 2003; Mukai 2015]. Using multidimensional skinning weights allows input transformations to be decomposed into different components, where each component is blended separately with one dimension of skinning weights [Wang and Phillips 2002; Merry et al. 2006; Jacobson and Sorkine 2011; Kavan and Sorkine 2012]. Cage-based deformers [Ju et al. 2005; Joshi et al. 2007; Lipman et al. 2008; Jacobson et al. 2011] also fall into in this category because each near-rigid part of the model is controlled by many surrounding cage vertices instead of a single transformation. Although adding more dimensions helps to resolve artifacts, an overhead is incurred as the extra controllers (bone transformations) have to be animated and extra skinning weights have to be computed or painted.

Indirect methods typically compute the skinning deformation in multiple passes. Correction methods [Vaillant et al. 2013; Vaillant et al. 2014] perform a direct skinning (DQS) in the first pass and then fix artifacts in a second pass by considering intersections or the global elasticity of the deformed model produced by the first pass. Poisson stitching techniques [Sumner and Popović 2004; Sumner et al. 2005; Wang et al. 2007; Weber et al. 2007] ignore smooth interpolations in the first pass by directly applying one rigid transformation per triangle, and then, in a second pass, solving the large, sparse Poisson equation to reconstruct a smooth surface from the rough, disconnected mesh produced by the first pass. Multiple passes can also be used to solve a non-linear optimization to produce an as-rigid-as-possible deformation [Sorkine and Alexa 2007; Jacobson et al. 2012a]. In general, indirect methods use multiple passes to solve vertex-vertex relationships, which are ignored by direct methods. Although these extra relationships improve the deformation quality, they make it more difficult to create a high-performance implementation on graphics hardware because significant communication between vertex computations is required.

Physical simulations provide many important visual effects that are missing from geometric methods. These include collisions [McAdams et al. 2011], jiggling [Liu et al. 2013], skin sliding [Li et al. 2013], and wrinkling [Rémillard and Kry 2013]. Simulation packages are included in professional tools, such as Maya Muscle[1] or Weta Digital's Tissue System[2]. However, setting up a character with an anatomical model and appropriate material properties requires significant effort [Teran et al. 2005; Lee et al. 2009]. In addition, physical simulation approaches require input of external forces, which is not native in the traditional animation pipeline [Capell et al. 2007; Hahn et al. 2012; Hahn et al. 2013]. Simulation-based approaches are typically not real-time due to the complexity of computing each time step and the small time steps required.

Data-driven methods use examples in the shape space to model the deformation behavior of the character model [Sloan et al. 2001]. Blendshape techniques [Lewis and Anjyo 2010; Seo et al. 2011] are the most commonly implemented data-driven methods. They compute the deformation as a linear combination of basic poses. Pose-space deformation [Lewis et al. 2000] uses non-linear shape blending with the parameters computed by linear least squares. Corrective shape methods also offer great flexibility because example shapes can be learned from simulation data, captured data, or user-provided data [Kry et al. 2002; Sumner et al. 2005; Anguelov et al. 2005; Park and Hodgins 2006; Wang et al. 2007; Park and Hodgins 2008; Feng et al. 2008; Kavan et al. 2011; Loper et al. 2014; Tsoli et al. 2014].

[1] www.autodesk.com
[2] www.wetafx.co.nz

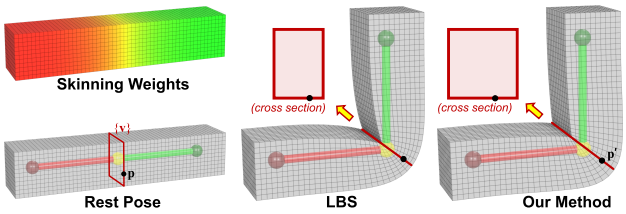


Figure 3: *An illustration of our method: Using the skinning weights and the rest pose (left), we first perform LBS (middle). LBS produces a collapsing artifact at a point* $\mathbf{p}$ *(black dot) near the joint due to the non-orthogonal output transformation matrix. We re-orthogonalize this transformation by finding the best rigid transformation* $[\mathbf{R_p}\ \mathbf{t_p}]$ *to match a set of vertices* $\{\mathbf{v}\}$ *(red lines) with similar skinning weights with* $\mathbf{p}$ *from the rest pose to the LBS-deformed pose. Applying* $[\mathbf{R_p}\ \mathbf{t_p}]$ *to* $\mathbf{p}$ *generates the final result that removes the collapsing artifact (right). In this toy example,* $\{\mathbf{v}\}$ *forms a cross section of the bar as all points on it have the same skinning weights.*

3 Method

We will first formulate the problem in the continuous domain and present the solution in Sections 3.1. Then, we will outline the discretized calculation on triangle meshes in Section 3.2. Finally, we will present the runtime algorithm in Section 3.3.

Let $\mathbf{\Omega} \subset \mathbb{R}^3$ be the (continuous) rest pose shape and $\mathbf{W} \subset \mathbb{R}^m$ be the skinning weights of all vertices $\mathbf{p} \in \mathbf{\Omega}$, where m denotes the number of bones and $\mathbf{w}_{\mathbf{p}j} \in \mathbb{R}$ denotes the skinning weight of bone j on vertex $\mathbf{p}$. Let $[\mathbf{R}_j\ \mathbf{t}_j]$ be the transformation matrix of bone j ($j = 1..m$), where $\mathbf{R}_j \in \mathbb{R}^{3\times3}$ is the rotation matrix and $\mathbf{t}_j \in \mathbb{R}^3$ is the translation vector.

We need to compute the rigid (orthogonal) transformation matrix $[\mathbf{R_p}\ \mathbf{t_p}]$ for the rest pose vertex $\mathbf{p}$ from bone transformations $[\mathbf{R}_j\ \mathbf{t}_j]$ ($\forall j = 1..m$), the rest pose $\mathbf{\Omega}$, and the skinning weights $\mathbf{W}$. The rotation $\mathbf{R_p}$ is directly computed by ***q**uaternion **l**inear **inter**polation* (QLERP) [Kavan and Žára 2005].

The translation $\mathbf{t_p}$ is indirectly inferred based on the assumption that a vertex $\mathbf{v}$ with similar skinning weights $\mathbf{w_v} \simeq \mathbf{w_p}$ will have similar transformations to $[\mathbf{R_p}\ \mathbf{t_p}]$. We can use a set of all vertices $\{\mathbf{v}\}$ to compute a better translation from an initial LBS deformation. This translation is the one that best matches $\mathbf{p}$ and all $\{\mathbf{v}\}$ from the rest pose to the LBS-deformed pose. Because this matching produces a single rigid transformation for a group of vertices with similar weights, it can prevent collapsing, as illustrated in Fig. 3. In Section 3.1, we will show that solving this matching problem allows us to pre-compute the *center of rotation* (CoR) for vertex $\mathbf{p}$.

Similarity Function. We define the similarity $s(\mathbf{w_p}, \mathbf{w_v})$ between two skinning weights of vertices $\mathbf{p}$ and $\mathbf{v}$ as the sum of the pairwise terms in Eq. (1), where the product of weights $\mathbf{w}_{\mathbf{p}j}\mathbf{w}_{\mathbf{p}k}\mathbf{w}_{\mathbf{v}j}\mathbf{w}_{\mathbf{v}k}$ represents the contributions of bone j and bone k and the difference $(\mathbf{w}_{\mathbf{p}j}\mathbf{w}_{\mathbf{v}k} - \mathbf{w}_{\mathbf{p}k}\mathbf{w}_{\mathbf{v}j})$ represents the distance in the normalized skinning weight space. The reason we use this difference as the distance metric is that even if the input skinning weights are affine (the

sum of skinning weights at each vertex is 1), the sum of skinning weights on bones j and k ($\mathbf{w}_{\mathbf{p}j} + \mathbf{w}_{\mathbf{p}k}$ and $\mathbf{w}_{\mathbf{v}j} + \mathbf{w}_{\mathbf{v}k}$) might not be. Thus, we consider two pairs of skinning weights $(\mathbf{w}_{\mathbf{p}j}, \mathbf{w}_{\mathbf{p}k})$ and $(\mathbf{w}_{\mathbf{v}j}, \mathbf{w}_{\mathbf{v}k})$ to be similar if $\mathbf{w}_{\mathbf{p}j} : \mathbf{w}_{\mathbf{p}k} \approx \mathbf{w}_{\mathbf{v}j} : \mathbf{w}_{\mathbf{v}k}$ or $\mathbf{w}_{\mathbf{p}j}\mathbf{w}_{\mathbf{v}k} \approx \mathbf{w}_{\mathbf{p}k}\mathbf{w}_{\mathbf{v}j}$. σ is the parameter that controls the width of the exponential kernel.

$$s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v}) = \sum_{\forall j \neq k} \mathbf{w}_{\mathbf{p}j}\mathbf{w}_{\mathbf{p}k}\mathbf{w}_{\mathbf{v}j}\mathbf{w}_{\mathbf{v}k}\, \mathrm{e}^{-\frac{(\mathbf{w}_{\mathbf{p}j}\mathbf{w}_{\mathbf{v}k} - \mathbf{w}_{\mathbf{p}k}\mathbf{w}_{\mathbf{v}j})^2}{\sigma^2}} \tag{1}$$

3.1 Solution in the Continuous Domain

For a vertex $\mathbf{p}$, we consider all vertices $\mathbf{v}$ with similar skinning weights, where the similarity is defined by $s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})$ in Eq. (1). We find the best translation $\mathbf{t}_\mathbf{p}$ by minimizing the weighted sum of squared errors on all vertices $\mathbf{v}$ in Eq. (2), where we want the transformation $[\mathbf{R}_\mathbf{p}\ \ \mathbf{t}_\mathbf{p}]$ to bring the rest pose $\forall \mathbf{v} \in \mathbf{\Omega}$ (source vertices) as close as possible to the LBS-deformed pose $\forall \widetilde{\mathbf{v}}$ (target vertices).

$$\mathbf{t}_\mathbf{p} = \arg\min_\mathbf{t} \int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v}) \left\|\mathbf{R}_\mathbf{p}\mathbf{v} + \mathbf{t} - \widetilde{\mathbf{v}}\right\|_2^2 \mathrm{d}\mathbf{v} \tag{2a}$$

$$\text{where: } \widetilde{\mathbf{v}} = \sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\mathbf{v} + \mathbf{t}_j\right) \tag{2b}$$

The minimization in Eq. (2) is a linear least squares problem with the solution obtained by the normal equation (3a). Substituting $\widetilde{\mathbf{v}}$ in Eq. (2b) and rearranging the equations yields:

$$\begin{aligned}
\mathbf{t}_\mathbf{p} &= \frac{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\left(\widetilde{\mathbf{v}} - \mathbf{R}_\mathbf{p}\mathbf{v}\right)\mathrm{d}\mathbf{v}}{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\,\mathrm{d}\mathbf{v}} && \text{(3a)}\\
&= \frac{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\left(\sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\mathbf{v} + \mathbf{t}_j\right) - \mathbf{R}_\mathbf{p}\mathbf{v}\right)\mathrm{d}\mathbf{v}}{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\,\mathrm{d}\mathbf{v}}\\
&= \sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\frac{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\mathbf{v}\,\mathrm{d}\mathbf{v}}{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\,\mathrm{d}\mathbf{v}} + \mathbf{t}_j\right)\\
&\qquad - \mathbf{R}_\mathbf{p}\frac{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\mathbf{v}\,\mathrm{d}\mathbf{v}}{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\,\mathrm{d}\mathbf{v}}\\
&= \sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\mathbf{p}^* + \mathbf{t}_j\right) - \mathbf{R}_\mathbf{p}\mathbf{p}^* && \text{(3b)}\\
&\qquad \text{where: } \mathbf{p}^* = \frac{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\mathbf{v}\,\mathrm{d}\mathbf{v}}{\int_{\mathbf{v}\in\mathbf{\Omega}} s(\mathbf{w}_\mathbf{p}, \mathbf{w}_\mathbf{v})\,\mathrm{d}\mathbf{v}} && \text{(3c)}
\end{aligned}$$

We can infer some properties from Eq. (3b) and Eq. (3c):

- The first term in Eq. (3b) ($\sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\mathbf{p}^* + \mathbf{t}_j\right)$) corresponds to applying LBS on $\mathbf{p}^*$.
- $\mathbf{p}^*$ plays the role of the CoR of the source vertices (the rest pose) and $\sum_{j=1}^{m} w_{\mathbf{p}j}\left(\mathbf{R}_j\mathbf{p}^* + \mathbf{t}_j\right)$ plays the role of the CoR of the target vertices (LBS-deformed pose).
- Most important, the computation of CoR $\mathbf{p}^*$ in Eq. (3c) is independent of the bone transformations; therefore, it can be pre-computed and cached.

3.2 Computation on Triangle Meshes

Discretization. Let T be the set of all triangles $t_{\alpha\beta\gamma}$ that represent an input model $\mathbf{\Omega}$. We compute the integration Eq. (3c) over all triangles $t_{\alpha\beta\gamma}$ on T. For each triangle, we use the average value of the skinning weights on three vertices to approximate the similarity $s(\cdot,\cdot)$ (computed by Eq. (1)). Because the triangle mesh is a piecewise linear function, the CoR of vertex $\mathbf{v}_i$ can be approximated as:

$$\mathbf{p}_\mathbf{i}^* = \frac{\sum_{t_{\alpha\beta\gamma}\in T} s(\mathbf{w}_i, \frac{\mathbf{w}_\alpha+\mathbf{w}_\beta+\mathbf{w}_\gamma}{3})\frac{\mathbf{v}_\alpha+\mathbf{v}_\beta+\mathbf{v}_\gamma}{3}a_{\alpha\beta\gamma}}{\sum_{t_{\alpha\beta\gamma}\in T} s(\mathbf{w}_i, \frac{\mathbf{w}_\alpha+\mathbf{w}_\beta+\mathbf{w}_\gamma}{3})a_{\alpha\beta\gamma}} \tag{4}$$

where $a_{\alpha\beta\gamma}$ denotes the area of triangle $t_{\alpha\beta\gamma}$, $\mathbf{v}_\alpha$ and $\mathbf{w}_\alpha$ denote the position and skinning weight of vertex α, respectively (and similarly for vertices i, β, and γ). To avoid large numerical error on the approximation of the similarity $s(\cdot,\cdot)$, we subdivide the input triangle mesh so that the l_2 skinning weight distance $||\mathbf{w}_i - \mathbf{w}_j||_2 < \epsilon$ for all edges e_{ij}. The subdivision is done by recursively bisecting an edge if the l_2 skinning weight distance between two vertices is greater than ϵ.

Integration. We observe that a naïve implementation to compute $\mathbf{p}_\mathbf{i}^*$ by summing all triangles can take up to an hour for a model with approximately 30,000 vertices (using our unoptimized C++ single-thread implementation on a consumer PC). Instead, we can greatly accelerate the performance (as reported in Table 1) by employing the following techniques:

- *Approximate nearest neighbor (ANN) search.* For each vertex i with skinning weight $\mathbf{w}_i$, we first query all vertices j with a similar skinning weight, i.e., vertices j, so that $||\mathbf{w}_i - \mathbf{w}_j||_2 < \omega$. We use these vertices j as the seeds for the next step. Our ANN search is accelerated by clustering all vertices in the skinning weight space using maximized minimum distance point sets [Schlömer et al. 2011] as the centers of clusters. The clusters are used to quickly reject distant vertices. We use approximately $\sqrt{n}$ clusters, where n is the number of vertices.
- *Smooth skinning weights assumption.* Typically, the input skinning weights are smooth on the surface of the models. Using this assumption, we perform *breadth-first search* on the triangle adjacency graph, starting from all triangles adjacent to the seed vertices in the previous step. We stop expanding the search if the similarity (Eq. (1)) is smaller than a threshold ε.
- *Parallel implementation.* Because the CoR of each vertex can be computed independently, we can compute all CoRs in parallel. We randomly shuffle the order of calculations on the list of vertices to balance the workload between different computing cores.

3.3 Runtime Algorithm

The pseudocode to compute the skinning deformation is given in Algorithm 1. There are two main steps:

- QLERP [Kavan and Žára 2005] from line 1 to line 6 computes the rotation matrix. The linear interpolation of unit quaternions uses the $\oplus$ operator (line 5) to resolve the quaternion antipodality [Kavan et al. 2008], where both quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. This $\oplus$ operator flips one quaternion to a consistent direction with the second before performing the sum, where the consistency between two quaternions is defined as the sign of their vector dot product.

Algorithm 1 Skeletal Skinning with Optimized Centers of Rotation

Input: n vertices, vertex i includes:
- Rest pose position $\mathbf{v}_i \in \mathbb{R}^3$
- Skinning weights $\mathbf{w}_i \in \mathbb{R}^m$
- CoR $\mathbf{p}_i^* \in \mathbb{R}^3$ computed by Eq. (1) and Eq. (4)

m bones, bone j transformation is $[\mathbf{R}_j \ \mathbf{t}_j] \in \mathbb{R}^{3\times 4}$

Output: Deformed position $\mathbf{v}_i' \in \mathbb{R}^3$ for all vertices $i = 1..n$

1: **for each** bone j **do**
2: Convert rotation matrix $\mathbf{R}_j$ to unit quaternion $\mathbf{q}_j$
3: **end for**
4: **for each** vertex i **do**
5: $\mathbf{q} \leftarrow w_{i1}\mathbf{q}_1 \oplus w_{i2}\mathbf{q}_2 \oplus \ldots \oplus w_{im}\mathbf{q}_m$
where: $\mathbf{q}_a \oplus \mathbf{q}_b = \begin{cases} \mathbf{q}_a + \mathbf{q}_b & \text{if } \mathbf{q}_a \cdot \mathbf{q}_b \geq 0 \\ \mathbf{q}_a - \mathbf{q}_b & \text{if } \mathbf{q}_a \cdot \mathbf{q}_b < 0 \end{cases}$
($\mathbf{q}_a \cdot \mathbf{q}_b$ denotes the vector dot product)
6: Normalize and convert $\mathbf{q}$ to rotation matrix $\mathbf{R}$
7: LBS: $[\widetilde{\mathbf{R}}\ \widetilde{\mathbf{t}}] \leftarrow \sum_{j=1}^{m} w_{ij}[\mathbf{R}_j \ \mathbf{t}_j]$
8: Compute translation: $\mathbf{t} \leftarrow \widetilde{\mathbf{R}}\mathbf{p}_i^* + \widetilde{\mathbf{t}} - \mathbf{R}\mathbf{p}_i^*$ (Eq. (3b))
9: $\mathbf{v}_i' \leftarrow \mathbf{R}\mathbf{v}_i + \mathbf{t}$
10: **end for**

- LBS in line 7 computes the transformation interpolation of the CoR $\mathbf{p}_i^*$.

These two steps are similar to SBS [Kavan and Žára 2005], except that we do not need to compute CoRs at runtime.

Scaling and Shearing. As our method imposes the orthogonal constraint on output transformations applied on vertices, it does not naturally support scaling and shearing of the input bone transformations. But similar to DQS [Kavan et al. 2008], scaling and shearing can be handled in two phases where scaling/shearing components are linearly blended in the first phase and then, our method is applied in the second phase to handle rotation and translation.

4 Results and Comparisons

We used five models, described in Table 1, to demonstrate our results and compare our method with others. The skinning weights for the first four models were computed by *bounded biharmonic weights with controlled extrema* [Jacobson et al. 2012b] and the skinning weights for the cloth model were computed by Maya's closest distance bind with a dropoff rate of 2.0. In all comparisons, we used the same skeleton animation for all the methods. In our visualizations, joints are represented by green spheres, bones are shown by yellow lines, and centers of rotation (CoRs) are represented by red dots.

Smooth skinning weights, such as *bounded biharmonic weights* [Jacobson et al. 2012b], do not generate a plausible twisting deformation when the twist appears at the joint (as shown at the top panel of Fig. 4). To distribute the twist along the bone, we split the bone by adding one extra joint at the middle of the bone and modify the original skinning weights by splitting the weight corresponding to the original bone $\mathbf{w}_j$ to form a new pair $\mathbf{w}_{j'}$ and $\mathbf{w}_j$. The weight $\mathbf{w}_{j'}$ corresponding to the new bone j' is computed by multiplying the original weight $\mathbf{w}_j$ with the projection weight e_{proj} (Eq. (8) in [Jacobson and Sorkine 2011]) and the original weight $\mathbf{w}_j$ is then subtracted by this value (illustrated in Fig. 5). The improvement with extra twist joints is shown in the bottom panel of Fig. 4.

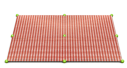
 Goliath	 man	 boy	 horse	 cloth
$n = 39158$ $f = 78312$ $m = 77$	$n = 20846$ $f = 41616$ $m = 76$	$n = 22459$ $f = 44360$ $m = 28$	$n = 8435$ $f = 16860$ $m = 31$	$n = 2601$ $f = 5000$ $m = 9$
$t = 15.7$ (s)	$t = 5.6$ (s)	$t = 18.0$ (s)	$t = 10.9$ (s)	$t = 0.7$ (s)

Table 1: *The models and the pre-computation time of our method on these models (horse model courtesy of Sumner and Popović [2004]). n denotes the number of vertices, f denotes the number of triangles, m denotes the number of bones, and t denotes the pre-computing time. The running time (in seconds) was recorded on a computer with an eight-core 2.40GHz CPU.*

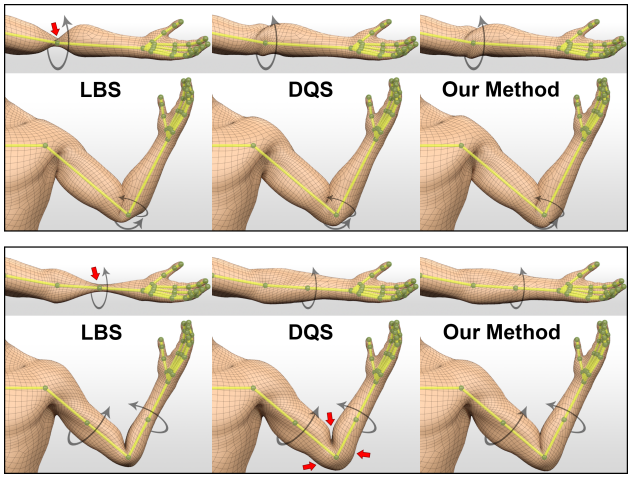


Figure 4: ***Top panel:*** *the traditional skeleton setup with bounded biharmonic weights produces an unnatural twisting deformation near the joints.* ***Bottom panel:*** *adding extra joints at the middle of the upper and lower arms can improve the quality without modifying skinning models. Even with the extra joints, LBS and DQS still exhibit candy wrapper and joint bulging artifacts (indicated by red arrows).*

Our simple solution is more effective than adding one extra dimension to the skinning weights [Jacobson and Sorkine 2011; Kavan and Sorkine 2012] because: first, it does not require modification of the data structure and the algorithm of the skinning models, and second, it has lower overhead because we only need a few twistable bones in practice. We find that adding four extra bones for the upper and lower arms is generally sufficient. This is a small number compared to always having to compute one extra twist deformer per bone [Jacobson and Sorkine 2011; Kavan and Sorkine 2012].

We found that our method is not sensitive to parameter selection. In our experiments, it produced visually indistinguishable results with any $\epsilon \leq \sigma \leq 0.1$, $\omega \geq 0.1$, and $\varepsilon \leq 10^{-6}$. We recall that σ is the parameter of the similarity function (1), ϵ is the subdivision threshold, ω is the ANN search threshold, and ε is the search cutoff threshold (ϵ, ω, and ε are described in §3.2). Any smaller values of ϵ, σ, and ε or larger value of ω only increases the execution time of the pre-computation step. For this reason, we use $(\epsilon, \sigma, \omega, \varepsilon) = (0.1, 0.1, 0.1, 10^{-6})$ as the parameters for all experiments. The pre-computing time with these parameters is reported in Table 1.

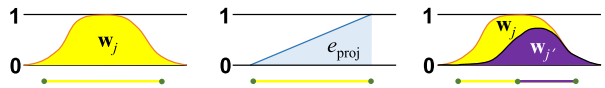


Figure 5: *The illustration of weight splitting for adding a twist joint. The new skinning weight $\mathbf{w}_{j'}$ (purple region) is computed by multiplying the original skinning weight $\mathbf{w}_j$ (yellow region on the left) with the projection weight e_{proj} (blue region) [Jacobson and Sorkine 2011] . The original bone is illustrated by the yellow line and the new bone is illustrated by the purple line at the bottom.*

LBS generates *candy wrapper* and *joint collapsing* because the linear blending produces shearing transformations that cause volume loss. In general, the joint collapsing (as shown in Fig. 8 and Fig. 9) is not as visually distracting as the candy wrapper, where a joint is twisted (as shown in Fig. 4 and Fig. 6). Our formulation (illustrated in Fig. 3) avoids this problem because it imposes an orthogonal (rigidness) constraint on the cross section that helps to preserve the volume. As shown in Fig. 4 and Fig. 6, OPS can completely remove the candy wrapper artifact on vertices influenced by two bones.

For vertices influenced by many joints, our method still exhibits a slight volume loss (as shown at the chest of the Goliath model in Fig. 6) because there is no unique CoR for more than two bones. We can investigate this issue by visualizing the CoRs (Fig. 7), which are distributed near the medial axes of the models. The CoRs form a one-dimensional curve for vertices influenced by two bones (vertices on the limbs). Because the curve has zero volume, applying LBS tranformations on these CoRs (line 7 in Algorithm 1) does not reduce the volume of the curve. As the result, all vertices rotating around this curve preserve the volume of the model. In contrast, CoRs of vertices influenced by more than two bones (vertices on the body of the models, especially at the chest and the hip) form a three-dimensional object which might lose volume after LBS transformations.

Other non-linear skinning methods (LMS, SBS, and DQS) can resolve collapsing artifacts, but they introduce a *bulging artifact* because all vertices near a joint can only rotate around a single CoR (the joint) [Kim and Han 2014]. In contrast, our method avoids bulging by computing per-vertex CoRs at the rest pose and transforming them smoothly at runtime. The CoRs transformations between different poses are visualized in Fig. 7.

LBS and DQS calculations are quite simple and robust. LMS is not robust to a twist/bend larger than 90° due to the non-injective inverse [Bloom and Blow 2004] (Fig. 8 and Fig. 9). SBS generates non-smooth deformation if two neighboring vertices are influenced by different sets of bones [Kavan et al. 2008] (Fig. 11 and Fig. 12). Although our method performs a non-trivial pre-computing step, we observe no robustness issue in these cases.

With different poses (Fig. 9), different skinning weights (Fig. 10), or a model with multiple components (Fig. 11 and Fig. 13), our method consistently produces better deformation quality than other direct skinning methods.

5 Discussion

We have presented a new direct skinning method that addresses the common artifacts of linear blend skinning (LBS) and dual quaternion skinning (DQS). Our main contribution is a method to precompute the per-vertex centers of rotation (CoRs) from only the rest pose and the skinning weights. These CoRs improve the deformation quality while still maintaining comparable performance with previous methods. Our method uses the standard skeletal-based skinning setup, so the computational cost to replace legacy LBS and DQS in the animation pipeline is insignificant.

We have considered alternative choices for the main steps of our algorithm and we explore the advantages and disadvantages of these alternative solutions in the discussion below.

An alternative to the approach presented in Section 3.1 of only finding the translation would be to compute both translation and rotation by jointly optimizing the objective function in Eq. (2). This formulation becomes the *Orthogonal Procrustes* problem [Kabsch 1978; Horn et al. 1988], where the optimum rotation can be solved by performing *singular value decomposition* (SVD) on a 3×3 cross-product matrix. This matrix can be cached as the CoR in Section 3. This solution does not require representing the bone rotations in quaternions, which could save some storage and data transfer. However, it has two limitations. First, it is costly to cache nine floats (for the cross product matrix) and perform SVD at runtime. Second, the optimum solution, which minimizes the sum of square distances (2a), does not work well for large rotations because distances are computed on straight line segments, as illustrated on the left side of Fig. 14.

Compared to other rotation interpolation methods [Shoemake 1985; Buss and Fillmore 2001; Alexa 2002], QLERP is only an approximate method. However, it offers two main advantages: first, it is quite simple and effective, and second, it can handle multiple rotations. As shown by Kavan and Žára [2005], the upper bound error of QLERP is only 8.15° for a 180° twist and the practical error is typically much smaller. While our implementation resolves antipodality per-vertex (Section 3.3), better performance can be gained by solving it per-bone when the bone hierarchy (skeleton) is given [Kavan et al. 2008].

Instead of performing per-triangle integration (Section 3.2), we can perform per-vertex integration with the infinitesimal $\mathrm{d}\mathbf{v}$ to be computed as the *Voronoi* area of vertex $\mathbf{v}$. However, per-vertex integration is more difficult because the local patches are polygonal (Voronoi regions around vertices). We found that first-order integration (with linear interpolation) gives visually good results on triangle meshes.

Generally, any similarity function with the form of the inverse distance in the skinning weights space is suitable. However, a function with a very fast dropoff might affect the robustness of our method. For example, the *Gaussian radial basis function* $s(\mathbf{w_p}, \mathbf{w_v}) = \mathrm{e}^{-\frac{||\mathbf{w_p}-\mathbf{w_v}||_2^2}{\sigma^2}}$ does not work well on vertices influenced by more than two bones (e.g. vertices near branching joints in the skeleton) because there might only be a small number of vertices with similar weights that fall into the hyper-sphere centered at $\mathbf{w_{\bar{p}}}$ in the high-dimensional non-zero skinning weight space.

Our method still has some of common limitations of direct skinning methods. It does not handle collisions, so the produced deformation might self intersect. It also omits dynamics and global constraints, so our results lack effects such as jiggling or skin sliding. For the best performance, our method requires an optimized set of skinning weights. This requirement is a challenge for a weight-painting tool because the pre-computed CoRs need to be updated on the fly with the weight changes. In the future, we will investigate dedicated tools to set up good skinning weights for this approach.

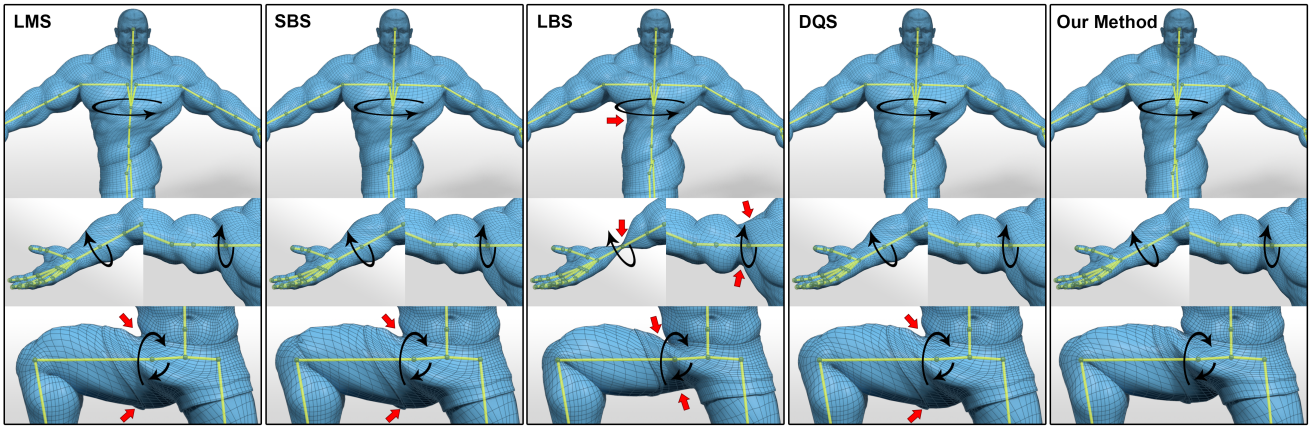


Figure 6: *A comparison on the Goliath model with big muscles: LBS shows a strong candy wrapper artifact on twisting while non-linear methods (LMS, SBS, and DQS) show a bulging artifact at the thigh. Notice how the thigh bone is off-center and there is no crease at the hip. In these examples, our method generates more natural deformation, even at challenging places like the shoulder or the hip.*

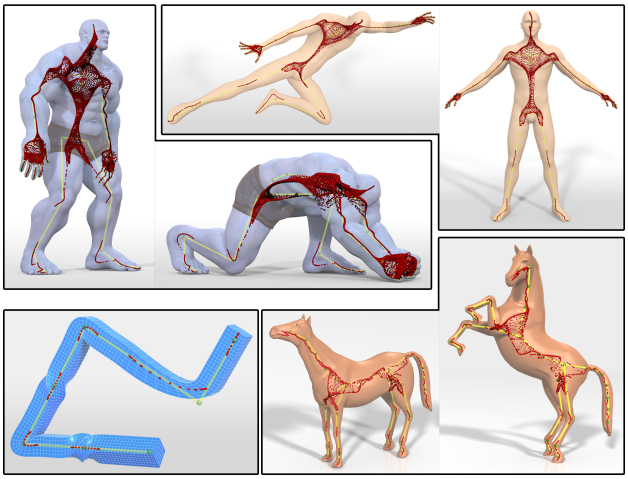


Figure 7: *Our centers of rotation (CoRs) are indicated by red dots. We only visualize CoR for vertices with more than one non-zero skinning weight, in which their pairwise similarities (Eq.* (1)*) exist so that their CoRs can be computed.*

Acknowledgements

We thank Moshe Mahler and Spencer Diaz for their artistic modeling and the animation of Goliath, man, and boy models. We thank Elizabeth Carter for proofreading the paper. We also thank Robert Sumner and Jovan Popović [2004] for making the horse model available.

References

ALEXA, M. 2002. Linear combination of transformations. *ACM Trans. Graph. 21*, 3 (July), 380–387.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (July), 408–416.

AUTODESK, 2016. Creating a character rig by Maya learning channel. https://www.youtube.com/playlist?list=PL8hZ6hQCGHMXKqaX9Og4Ow52jsU_Y5veH. (accessed January 19, 2016).

BLOOM, C., AND BLOW, J., 2004. Errors and omissions in Marc Alexa's "Linear combination of transformations". http://www.cbloom.com/3d/techdocs/lcot_errors.pdf. (accessed January 19, 2016).

BUSS, S. R., AND FILLMORE, J. P. 2001. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph. 20*, 2 (Apr.), 95–126.

CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2007. Physically based rigging for deformable characters. *Graph. Models 69*, 1 (Jan.), 71–87.

FENG, W.-W., KIM, B.-U., AND YU, Y. 2008. Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph. 27*, 3 (Aug.), 91:1–91:9.

FORSTMANN, S., OHYA, J., KROHN-GRIMBERGHE, A., AND MCDOUGALL, R. 2007. Deformation styles for spline-based skeletal animation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 141–150.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Trans. Graph. 31*, 4 (July), 72:1–72:8.

HAHN, F., THOMASZEWSKI, B., COROS, S., SUMNER, R. W., AND GROSS, M. 2013. Efficient simulation of secondary motion in rig-space. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 165–171.

HORN, B. K. P., HILDEN, H. M., AND NEGAHDARIPOUR, S. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A 5*, 7 (Jul), 1127–1135.

JACOBSON, A., AND SORKINE, O. 2011. Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. Graph. 30*, 6 (Dec.), 165:1–165:8.

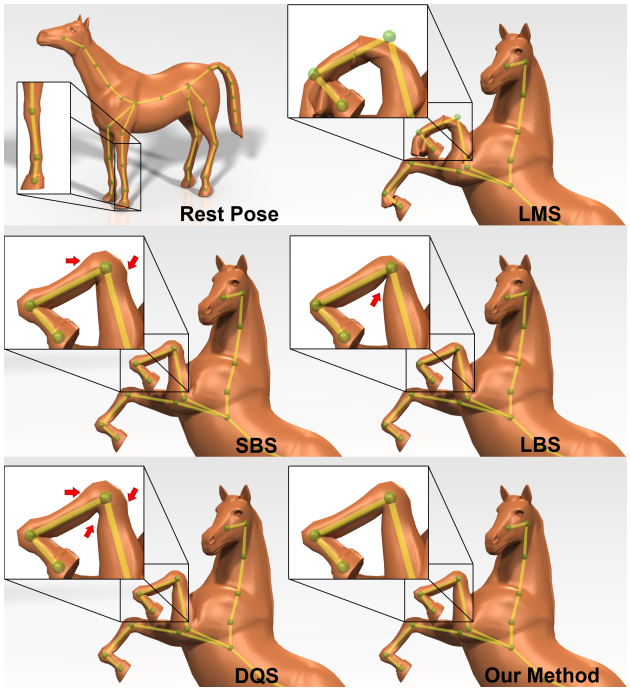


Figure 8: *Joint bending comparison on the horse model: LMS is not robust because of the large bending angles (more than 90°). SBS and DQS generate similar results with bulges. Because of this bulging, DQS could not form a crease on the inside of the bent joint. LBS shows no bulge but it reduces the cross section at the joint (elbow collapsing). Artifacts are indicated by red arrows.*

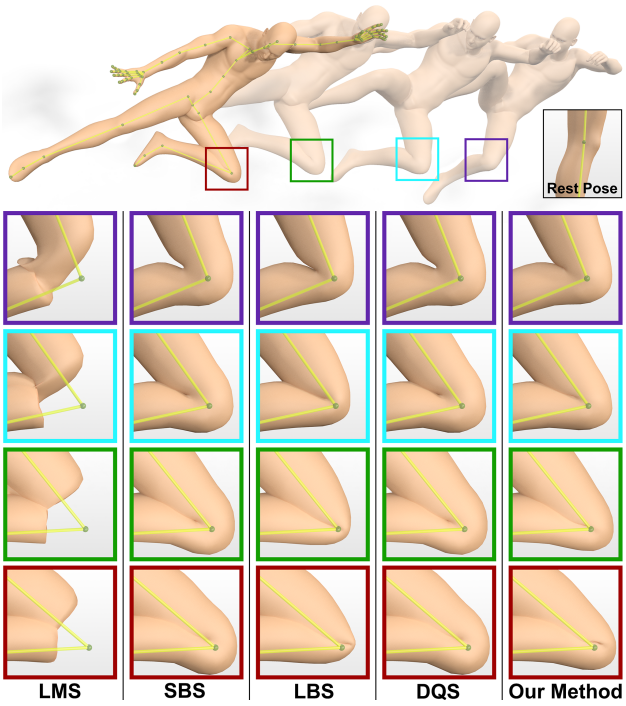


Figure 9: *Some keyframes of the man animation show different knee joint bending angles. As the bend increases, skinning artifacts appear to be more serious with all skinning methods: LMS deviates further from a good solution due to the robustness issue, SBS and DQS show a larger bulge, LBS folds the knee too much and it makes the skeleton appears further to the center of the leg, and our method shows the most pleasing deformation among all methods. Detailed crops at the bottom are color-coded corresponding to the keyframes above.*

JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4 (July), 78:1–78:8.

JACOBSON, A., BARAN, I., KAVAN, L., POPOVIĆ, J., AND SORKINE, O. 2012. Fast automatic skinning transformations. *ACM Trans. Graph. 31*, 4 (July), 77:1–77:10.

JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *Comput. Graph. Forum 31*, 5 (Aug.), 1577–1586.

JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph. 26*, 3 (July).

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (July), 561–566.

KABSCH, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A 34*, 827–828.

KAVAN, L., AND SORKINE, O. 2012. Elasticity-inspired deformers for character articulation. *ACM Trans. Graph. 31*, 6 (Nov.), 196:1–196:8.

KAVAN, L., AND ŽÁRA, J. 2005. Spherical blend skinning: A real-time deformation of articulated models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, ACM, I3D '05, 9–16.

KAVAN, L., COLLINS, S., ŽÁRA, J., AND O'SULLIVAN, C. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (Nov.), 105:1–105:23.

KAVAN, L., GERSZEWSKI, D., BARGTEIL, A. W., AND SLOAN, P.-P. 2011. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph. 30*, 4 (July), 93:1–93:10.

KIM, Y., AND HAN, J. 2014. Bulging-free dual quaternion skinning. *Comput. Animat. Virtual Worlds 25*, 3-4, 321–329.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. Eigenskin: Real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 153–159.

LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28*, 4 (Sept.), 99:1–99:17.

LEWIS, J. P., AND ANJYO, K.-I. 2010. Direct manipulation blendshapes. *IEEE Comput. Graph. Appl. 30*, 4 (July), 42–50.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 165–172.

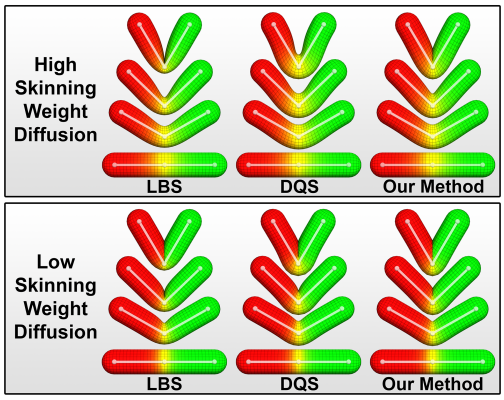


Figure 10: *A comparison with different skinning weight diffusions. Our method always reduces the LBS collapsing and DQS bulging.*

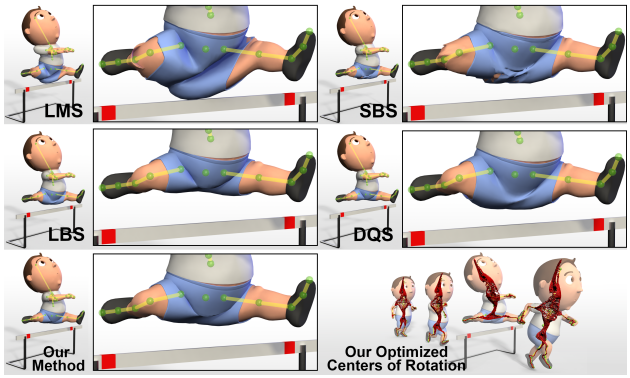


Figure 11: *Extreme bends at the hip socket joints amplify the bulging artifact of non-linear methods (LMS, SBS, and DQS). While our method reduces bulging, it does not reduce the volume of the thighs as LBS does. SBS shows discontinuities because of the bad centers of rotation estimation.*

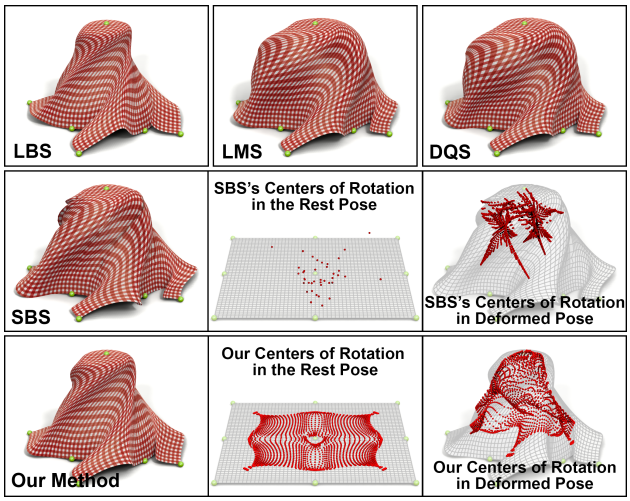


Figure 12: *A comparison of skinning a cloth model with non-hierarchical joints: LBS produces quite natural deformation because its linear interpolation takes the shortest straight line so stretch is minimized. Non-linear interpolations are not suitable because they produce more stretch (see how LMS and DQS produce a stretched texture deformation). Because the input model is planar, the centers of rotation (CoRs)are distributed on the plane that makes our method behave like LBS and produces a pleasing deformation. The CoRs are computed per-vertex so the deformation does not suffer from discontinuities like SBS, where the CoRs of SBS are limited in number.*

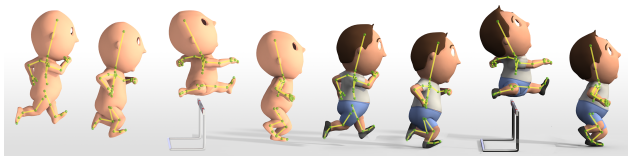


Figure 13: *Our method can process a model with multiple components. The boy model shown here includes the underlying skin layer and other components (hair, shirt, pant, and shoes). The eyes are not bound to the skinning model.*

LI, D., SUEDA, S., NEOG, D. R., AND PAI, D. K. 2013. Thin skin elastodynamics. *ACM Trans. Graph. 32*, 4 (July), 49:1–49:10.

LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. *ACM Trans. Graph. 27*, 3 (Aug.), 78:1–78:10.

LIU, L., YIN, K., WANG, B., AND GUO, B. 2013. Simulation and control of skeleton-driven soft body characters. *ACM Trans. Graph. 32*, 6 (Nov.), 215:1–215:8.

LOPER, M., MAHMOOD, N., AND BLACK, M. J. 2014. Mosh: Motion and shape capture from sparse markers. *ACM Trans. Graph. 33*, 6 (Nov.), 220:1–220:13.

MAGNENAT-THALMANN, N., LAPERRIÈRE, R., AND THALMANN, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface '88*, 26–33.

MAGNENAT-THALMANN, N., CORDIER, F., SEO, H., AND PAPAGIANAKIS, G. 2004. Modeling of bodies and clothes for virtual environments. In *Cyberworlds, 2004 International Conference on*, 201–208.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. 30*, 4 (July), 37:1–37:12.

MERRY, B., MARAIS, P., AND GAIN, J. 2006. Animation space: A truly linear framework for character animation. *ACM Trans. Graph. 25*, 4 (Oct.), 1400–1423.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. *ACM Trans. Graph. 22*, 3 (July), 562–568.

MUKAI, T. 2015. Building helper bone rigs from examples. In *Proceedings of the 19th ACM Symposium on Interactive 3D Graphics and Games*, 77–84.

MÜLLER, M., AND CHENTANEZ, N. 2011. Solid simulation with oriented particles. *ACM Trans. Graph. 30*, 4 (July), 92:1–92:10.

ÖZTIRELI, A. C., BARAN, I., POPA, T., DALSTEIN, B., SUMNER, R. W., AND GROSS, M. 2013. Differential blending for expressive sketch-based posing. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 155–164.

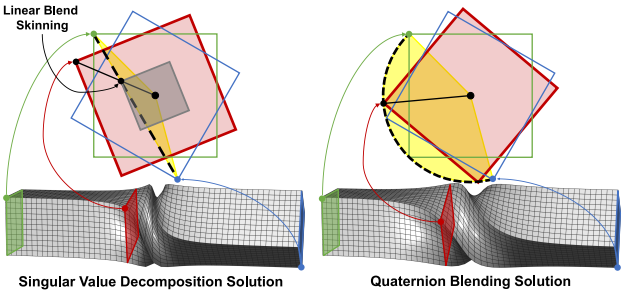


Figure 14: *The deformation skinning of a bar with two bones where the joint is twisted by* 150°*. Cross-section views are shown on top.* ***Left:*** *a rotation computed by performing singular value decomposition (SVD) does not interpolate smoothly near the joint because it is only a correction of the LBS (illustrated by a gray square cross-section), which interpolates positions on a line segment (illustrated by a dashed black line).* ***Right:*** *a rotation computed by* ***q****uaternion* ***l****inear int****erp****olation (QLERP) interpolates positions on a circular segment (illustrated by a dashed black arc).*

PARK, S. I., AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. *ACM Trans. Graph. 25*, 3 (July), 881–889.

PARK, S. I., AND HODGINS, J. K. 2008. Data-driven modeling of skin and muscle deformation. *ACM Trans. Graph. 27*, 3 (Aug.), 96:1–96:6.

RÉMILLARD, O., AND KRY, P. G. 2013. Embedded thin shells for wrinkle simulation. *ACM Trans. Graph. 32*, 4 (July), 50:1–50:8.

SCHLÖMER, T., HECK, D., AND DEUSSEN, O. 2011. Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, 135–142.

SEO, J., IRVING, G., LEWIS, J. P., AND NOH, J. 2011. Compression and direct manipulation of complex blendshape models. *ACM Trans. Graph. 30*, 6 (Dec.), 164:1–164:10.

SHOEMAKE, K. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '85, 245–254.

SLOAN, P.-P. J., ROSE, III, C. F., AND COHEN, M. F. 2001. Shape by example. In *Proceedings of the 2001 ACM Symposium on Interactive 3D Graphics*, 135–143.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 109–116.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug.), 399–405.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. *ACM Trans. Graph. 24*, 3 (July), 488–495.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 181–190.

TSOLI, A., MAHMOOD, N., AND BLACK, M. J. 2014. Breathing life into shape: Capturing, modeling and animating 3D human breathing. *ACM Trans. Graph. 33*, 4 (July), 52:1–52:11.

VAILLANT, R., BARTHE, L., GUENNEBAUD, G., CANI, M.-P., ROHMER, D., WYVILL, B., GOURMEL, O., AND PAULIN, M. 2013. Implicit skinning: Real-time skin deformation with contact modeling. *ACM Trans. Graph. 32*, 4 (July), 125:1–125:12.

VAILLANT, R., GUENNEBAUD, G., BARTHE, L., WYVILL, B., AND CANI, M.-P. 2014. Robust iso-surface tracking for interactive character skinning. *ACM Trans. Graph. 33*, 6 (Nov.), 189:1–189:11.

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 129–138.

WANG, R. Y., PULLI, K., AND POPOVIĆ, J. 2007. Real-time enveloping with rotational regression. *ACM Trans. Graph. 26*, 3 (July).

WEBER, O., SORKINE, O., LIPMAN, Y., AND GOTSMAN, C. 2007. Context-aware skeletal shape deformation. *Comput. Graph. Forum 26*, 3, 265–274.

YANG, X., SOMASEKHARAN, A., AND ZHANG, J. J. 2006. Curve skeleton skinning for human and creature characters: Research articles. *Comput. Animat. Virtual Worlds 17*, 3-4 (July), 281–292.