Elasticity-Inspired Deformers for Character Articulation

Ladislav Kavan* Olga Sorkine

ETH Zurich

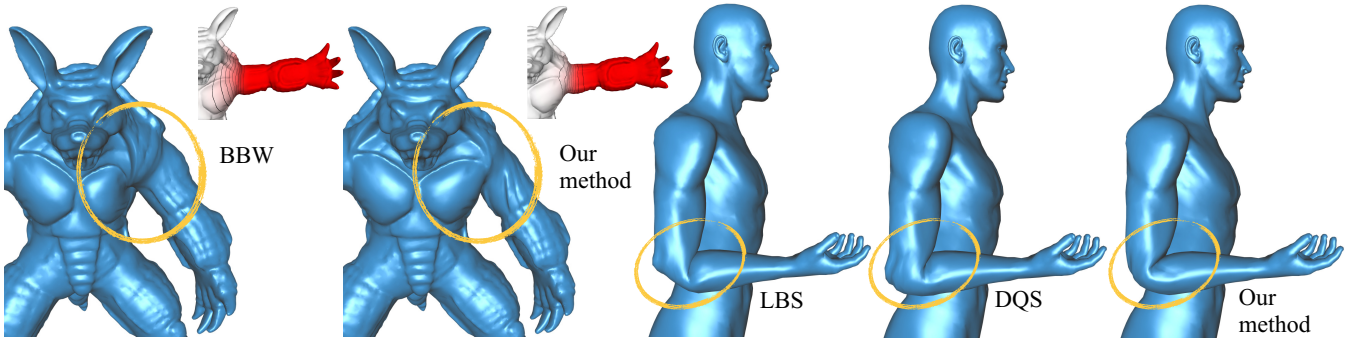

Figure 1: *Previous automatic weighting schemes, such as bounded biharmonic weights (BBW) are deformation-agnostic, whereas our method optimizes weights to minimize shape distortion. Combined with our joint-based deformers, we achieve higher-quality results than both linear blend (LBS) and dual quaternion skinning (DQS). Our technique requires no additional user input and its speed is competitive to linear skinning.* © *2012 The Authors*

Abstract

Current approaches to skeletally-controlled character articulation range from real-time, closed-form skinning methods to offline, physically-based simulation. In this paper, we seek a closed-form skinning method that approximates nonlinear elastic deformations well while remaining very fast. Our contribution is two-fold: (1) we optimize skinning weights for the standard linear and dual quaternion skinning techniques so that the resulting deformations minimize an elastic energy function. We observe that this is not sufficient to match the visual quality of the original elastic deformations and therefore, we develop (2) a new skinning method based on the concept of *joint-based deformers*. We propose a specific deformer which is visually similar to nonlinear variational deformation methods. Our final algorithm is fully automatic and requires little or no input from the user other than a rest-pose mesh and a skeleton. The runtime complexity requires minimal memory and computational overheads compared to linear blend skinning, while producing higher quality deformations than both linear and dual quaternion skinning.

CR Categories: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Animation

Keywords: Skeletal shape deformation, skinning, elasticity.

Links: DL PDF WEB VIDEO

*e-mail: ladislav.kavan@gmail.com

1 Introduction

Real-time approaches for character articulation, known as skinning or skeletal subspace deformation, are necessary for interactive animation interfaces and applications such as computer games and crowd simulation. Direct geometric methods, e.g., linear or dual quaternion skinning deliver the required speed, but often at the cost of compromising quality. Physically-based models produce high-quality, realistic deformations at the expense of much more complex computation. In this paper, we study the question of how to design a real-time, direct skinning method that delivers visually similar results to offline elastic simulation, albeit without collisions.

This problem is also motivated by requests from professional rigging artists who enjoy the improvements offered by dual quaternion skinning, but dislike the joint-bulging artifacts (Fig. 1), that require them to apply manual fix-ups. Physically-based methods deliver high-quality deformations automatically, but even the latest highly optimized algorithms [McAdams et al. 2011] are not fast enough for interactive posing. In this paper, we consider a common physically-based model – an elastic solid with rigid bones embedded inside. An input skeletal pose induces deformations through minimization of a nonlinear elastic energy, subject to rigid bone constraints. While being only a gross simplification of real anatomy, this model yields intuitive, high-quality deformations applicable to both realistic and stylized characters.

Our initial idea to improve the results of closed-form skinning techniques was to find weights that minimize nonlinear elastic energy over a range of skeletal poses. While achieving better results than deformation-agnostic weights, such as bounded biharmonic weights [Jacobson et al. 2011], we found that the results still contained objectionable visual artifacts when compared to elastic simulation. Indeed, rigging artists also experimentally confirmed that the space of traditional skinning deformations is not rich enough to approximate the nonlinear elastic behavior well. We identify the culprit to be the isotropy of linear and dual quaternion blending operators, causing these methods to treat all directions in material-space as equal. This contrasts physics, which, even with homogeneous elasticity, results in quite different deformations along the direction of the bone (which remains rigid) and in the orthogonal directions (where squash and stretch occur).

To overcome this problem, we design a new closed-form skinning model based on the concept of joint-based *deformers*. In general, a *deformer* is a spatial mapping $\Gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that depends on some input parameters. An arbitrary character rig can be described as a single deformer that has all the rig parameters (such as joint angles) as inputs. Since directly designing such all-embracing deformers would be difficult, linear blend skinning approximates the global spatial deformations as a linear combination of local affine transformations, corresponding to transformations of individual bones. While accurately predicting motion near bone midpoints, the deformations around joints can only be controlled indirectly by modifying the skinning weights.

In this paper, we propose to focus on deformations near joints in the first place, introducing a general concept of *joint-based deformers*. Each joint-based deformer is associated with a pair of adjacent bones (parent and child) and is assumed to preserve the position and orientation of both bones. Unlike linear blend skinning, a joint-based deformer is parameterized by the relative rotation between the parent and child bones and therefore can describe complex nonlinear spatial deformations. The final result is a linear combination of joint-based deformers, assuming that all significant nonlinearities have been resolved at the deformer level.

We show how to construct a simple joint-based deformer that resembles full elastic energy minimization while departing as little as possible from the established skinning techniques. The main idea is to decompose the joint rotation into its swing and twist components, where swing is the rotation of the bone axis and twist is the rotation along this axis. For twist we use 2D rotation interpolation and the swing is blended linearly. By decoupling bending from torsion, this simple deformer avoids the candy-wrapper artifacts of linear blend skinning as well as the bulging artifacts of dual quaternion skinning. This is done without any alterations to the rigid bones metaphor. The weights of our swing/twist deformer, used to control the spatial distribution of bending and torsion, are optimized to minimize the elastic energy over a default range of joint rotations.

The setup of our joint-based deformers is fully automatic for a given rest-pose mesh, skeleton, and the resolution of our volumetric discretization. This makes our approach as easy to use as current automatic skinning systems, while delivering superior quality comparable to physically-based simulation.

2 Related Work

Arguably the most accurate approach to character articulation involves simulation of the underlying anatomy [Lee et al. 2009]. This is, however, quite difficult to author and tune. A simpler physics-based method is to approximate the body with homogeneous elastic material where bones serve as rigid constraints that specify the pose [Capell et al. 2002; Capell et al. 2005]. Related techniques have been studied in the more general context of variational surface deformation [Botsch and Sorkine 2008]. Unfortunately, even the recent highly optimized methods [McAdams et al. 2011] are still too slow for interactive posing, even though this is partly due to collision and contact handling. One possible solution is to employ reduced-order physics [Barbič and Zhao 2011; Kim and James 2011] or subspace energy minimization [Jacobson et al. 2012]. Our technique is orthogonal to these methods, as we focus on designing a suitable deformation subspace itself.

Closed-form skinning techniques, such as linear blend skinning [Magnenat-Thalmann et al. 1988], feature very low time complexity, which is necessary in real-time applications such as games. Unfortunately, linear skinning suffers from the well-known candy-wrapper artifacts. Multi-linear techniques [Wang and Phillips 2002; Merry et al. 2006] suppress (but not completely eliminate) this problem by using additional skinning weights. These extra weights are learned from input examples, using regularization to prevent overfitting. Dual quaternion skinning [Kavan et al. 2008] replaces linear blending with a nonlinear transformation blending method, eliminating the candy-wrapper artifacts. However, a side effect of dual quaternion blending is a bulging artifact while bending. To address this issue, software such as Autodesk Maya™allows the blending of linear and dual quaternion skinning together, which has the side effect of reintroducing the candy-wrapper shrinkage while twisting.

In practice, linear or dual quaternion skinning weights are often painted manually by specialized rigging artists. Because this is a tedious process, automatic skinning techniques [Baran and Popović 2007; Wareham and Lasenby 2008; Chen et al. 2011; Jacobson et al. 2011] are becoming increasingly popular. These methods produce shape-aware skinning weights, but optimize only for smoothness of the resulting deformations. Our method finds weights that minimize nonlinear elastic energy of the induced deformations. In addition to being smooth and shape-aware, our new weights are therefore also deformation-aware.

One important approach to improve skinning quality uses additional example shapes to add data-driven details, or "residuals" [Lewis et al. 2000; Sloan et al. 2001]. Manual creation of examples can be difficult; Kry and colleagues [2002] use physically-based simulation to automatically generate the example shapes and compress the residuals using PCA to reduce memory costs. An alternative approach is to learn regression models for individual triangle rotations and use Poisson stitching to obtain the final vertex positions [Weber et al. 2007; Wang et al. 2007]. While an efficient GPU implementation is possible [Wang et al. 2007], it requires a multi-pass approach involving dense matrix multiplication. Instead of using examples, our method works directly with deformation energies. We demonstrate that this approach is more efficient than first computing examples by energy-minimization and subsequent weight fitting, both in terms of speed and the resulting quality.

Certain special types of deformers have been discussed in previous work. Specifically, deformers based on intrinsic $SO(3)$ [Mohr and Gleicher 2003] and $SE(3)$ [Kavan et al. 2009] interpolation eliminate the candy-wrapper artifacts but do not remove the bulging problems while bending. Sweep [Hyun et al. 2005] or spline-based deformers [Yang et al. 2006; Forstmann et al. 2007; Gregory and Weston 2008] produce higher-quality results but are more complex and feature many parameters that need to be tuned manually. The concept of flexible bones also requires different rigging workflows, which can be advantageous with sketch-based interfaces [Kho and Garland 2005]. In contrast, our approach is fully automatic and does not depart from the common rigid-bones metaphor. Recently, Jacobson and Sorkine [2011] enhanced linear and dual quaternion skinning with additional effects: stretch and secondary twist, distributed along the entire length of a bone. Our method focuses on joint-induced deformations and assumes rigid bones. If stretchable and twistable bones are desired, their technique can be combined with our method.

Cage-based deformation presents an interesting alternative to skeletally-controlled articulations [Joshi et al. 2007]. Ju and colleagues [2008] make this approach more practical for rigging by introducing reusable cage-based deformers. Other important properties of character deformation include volume preservation and avoiding self-intersections [Angelidis and Singh 2007; Rohmer et al. 2009]. Our elastic energy also encourages local volume preservation, however our simple swing/twist deformer does not guarantee intersection-free results.

3 Method

The input to our method is the rest-pose mesh of a character and a corresponding skeleton. We begin by describing the reference nonlinear elasticity formulation that produces the "ideal" skeletal deformations for our purposes (Sec. 3.1). We derive skinning weights for standard linear blend skinning (LBS) and dual quaternion skinning (DQS) so as to optimally approximate the ideal deformation behavior within those skinning deformation spaces (Sec. 3.2). As the standard skinning deformations are not expressive enough even with the optimized weights, we propose our new joint-based deformers in order to introduce a richer deformation space with the desired properties. We first discuss the general concept of joint-based deformers (Sec. 3.3) before explaining our specific joint-based deformer (called the *swing/twist deformer*) developed to resolve the artifacts of both LBS and DQS (Sec. 3.4).

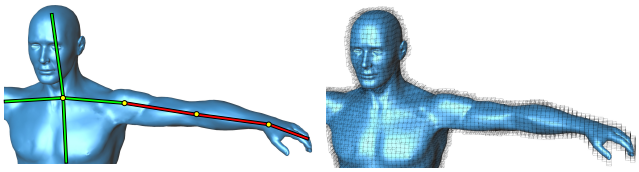

Figure 2: *Example skeleton and a volumetric discretization.*

3.1 Elastic Energy for Skeletal Deformation

Inspired by McAdams and colleagues [2011], we discretize nonlinear elasticity using a regular voxel grid enclosing the given rest-pose mesh. Although tetrahedral meshing can achieve higher accuracy with the same number of primitives, it is difficult to construct while conforming to the given mesh surface especially if self-intersections or non-manifold issues are present in the input model. The skeleton is a collection of points representing joints, with bones corresponding to line segments connecting the joints (Fig. 2 left). Voxels intersecting a bone are assumed to be rigidly attached to the bone. Since we aim for a fully-automatic procedure, we chose the simple geometric elasticity model by Chao and colleagues [2010]. Let m be the number of voxels, n the number of vertices in the voxel grid, and $\mathbf{v} \in \mathbb{R}^{3n}$ denote the 3D positions of those vertices in a deformed configuration. In matrix form, the discrete elastic energy of [Chao et al. 2010] can be written as:

$$E(\mathbf{v}) = \frac{1}{2}\mathbf{v}^\mathsf{T}\mathbf{L}\mathbf{v} - \min_{\mathbf{r}\in\mathcal{Q}_m}(\mathbf{r}^\mathsf{T}\mathbf{K}\mathbf{v}) \tag{1}$$

where $\mathbf{L}$ is the Laplace operator on the voxel grid, $\mathbf{r} \in \mathbb{R}^{9m}$ are the stacked coordinates of m best-fitting rotation matrices, $\mathcal{Q}_m = SO(3) \times \ldots \times SO(3) = SO(3)^m$, and $\mathbf{K}$ is a matrix of differential rest-pose coordinates. Formally, if $\mathbf{B}_i$ is an oriented incidence matrix corresponding to i-th voxel and $\tilde{\mathbf{V}} \in \mathbb{R}^{n\times 3}$ are rest-pose coordinates of voxel vertices, $\mathbf{L}$ and $\mathbf{K}$ can be defined as:

$$\mathbf{L} = \left(\sum_{i=1}^{m}\mathbf{B}_i\mathbf{B}_i^\mathsf{T}\right)\otimes\mathbf{I}_3, \quad \mathbf{K} = \begin{pmatrix}\tilde{\mathbf{V}}^\mathsf{T}\mathbf{B}_1\mathbf{B}_1^\mathsf{T}\\ \vdots \\ \tilde{\mathbf{V}}^\mathsf{T}\mathbf{B}_m\mathbf{B}_m^\mathsf{T}\end{pmatrix}\otimes\mathbf{I}_3 \tag{2}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$ is an identity matrix. Different material properties could be specified by changing matrices $\mathbf{L}$ and $\mathbf{K}$, e.g., weighting each voxel by a different stiffness value. However, we use only the uniform homogeneous model (Eq. 2). Further details can be found in the literature [Chao et al. 2010; Jacobson et al. 2012].

Intuitively, $E(\mathbf{v})$ finds the best-fitting rotation for each voxel and measures the squared distance between its current (deformed) configuration and its rest pose transformed by the best-fitting rotation. Similar models, belonging to the broader class of corotational methods, are popular in physics-based simulation for their simplicity and robustness [Müller et al. 2005; Rivers and James 2007]. The "ideal" deformation is computed by finding $\mathbf{v}$ that minimizes $E(\mathbf{v})$ subject to the constraints that the positions of bone-intersecting voxels are given (specified by a desired skeleton pose).

3.2 Elastic Energy-minimizing Weights

Let us first study deformations induced by rotations of a single joint, e.g., left shoulder (Fig. 2 left). As all other joints are fixed, this corresponds to rotating the whole skeletal subtree starting at this joint (the red bones in Fig. 2, while the green ones are fixed). Direct minimization of Eq. 1 requires iteration and is too slow for runtime. Our initial goal is to generate similar deformations by finding appropriate weights for linear or dual quaternion skinning. In the described setting we need two weight functions, $\mathbf{w}_1 \in \mathbb{R}^n$ for the identity transformation of the fixed bones and $\mathbf{w}_2 \in \mathbb{R}^n$ for the rotation transformation of the rotating bones. Denote by $\mathbf{w} = (\mathbf{w_1}^\mathsf{T}\ \mathbf{w_2}^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{2n}$ the concatenated vector of the weights. The weights computed on the voxel grid are subsequently mapped to mesh vertices by trilinear interpolation.

To optimize the weights, we need representative rotation samples $\mathbf{S}_1, \ldots, \mathbf{S}_k \in SO(3)$, i.e., poses of the joint, and we simultaneously minimize the deformation energy (Eq. 1) *in weight space* for all these sample poses. While arbitrary sampling is possible, in our final method we use only a very simple set of samples, identical for all joints. Specifically, for swing weights, we use just 4 sample rotations $\mathbf{S}_1, \ldots, \mathbf{S}_4$, corresponding to ± 90 degrees rotations in x- and y-axes. For twist weights, we use only 2 rotation samples representing ± 90 degrees rotation about the z-axis.

Linear blend skinning for the i-th sample rotation can be described as $\mathbf{v}_i = \mathbf{M}_i\mathbf{w}$, where $\mathbf{v}_i \in \mathbb{R}^{3n}$ are the resulting deformed voxel vertices and $\mathbf{M}_i$ is a sparse matrix with the following structure:

$$\mathbf{M}_i = \left(\begin{array}{ccc|ccc} \tilde{\mathbf{v}}_1 & & & \mathbf{S}_i\tilde{\mathbf{v}}_1 & & \\ & \ddots & & & \ddots & \\ & & \tilde{\mathbf{v}}_n & & & \mathbf{S}_i\tilde{\mathbf{v}}_n \end{array}\right) \in \mathbb{R}^{3n\times 2n}$$

where $\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_n \in \mathbb{R}^3$ are rest-pose voxel vertices. We seek $\mathbf{w}$ that minimizes our energies summed over all k sample poses:

$$\begin{aligned}\tilde{E}(\mathbf{w}) &= \sum_{i=1}^{k} E(\mathbf{M}_i\mathbf{w}) \\ &= \sum_{i=1}^{k}\left(\frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{M}_i^\mathsf{T}\mathbf{L}\mathbf{M}_i\mathbf{w} - \min_{\mathbf{r}_i\in\mathcal{Q}_m}(\mathbf{r}_i^\mathsf{T}\mathbf{K}\mathbf{M}_i\mathbf{w})\right) \\ &= \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{Y}\mathbf{w} - \sum_{i=1}^{k}\min_{\mathbf{r}_i\in\mathcal{Q}_m}(\mathbf{r}_i^\mathsf{T}\mathbf{K}\mathbf{M}_i\mathbf{w})\end{aligned} \tag{3}$$

where we denoted $\mathbf{Y} = \sum_{i=1}^{k}\mathbf{M}_i^\mathsf{T}\mathbf{L}\mathbf{M}_i$. The requirement of the vertices of bone-intersecting voxels being rigidly attached to their bones can be expressed by setting the respective weights to 1. Appending the partition-of-unity constraints (required for translation-invariance), we obtain our final set of linear equality constraints.

To find the weights $\mathbf{w}$ minimizing $\tilde{E}(\mathbf{w})$, we employ the local/global block coordinate descent advocated by Sorkine and Alexa [2007]. In the local step ($\mathbf{w}$ fixed), we solve the Procrustes problem to find the best-fitting rotations $\mathbf{r}_1, \ldots, \mathbf{r}_k$. This step is

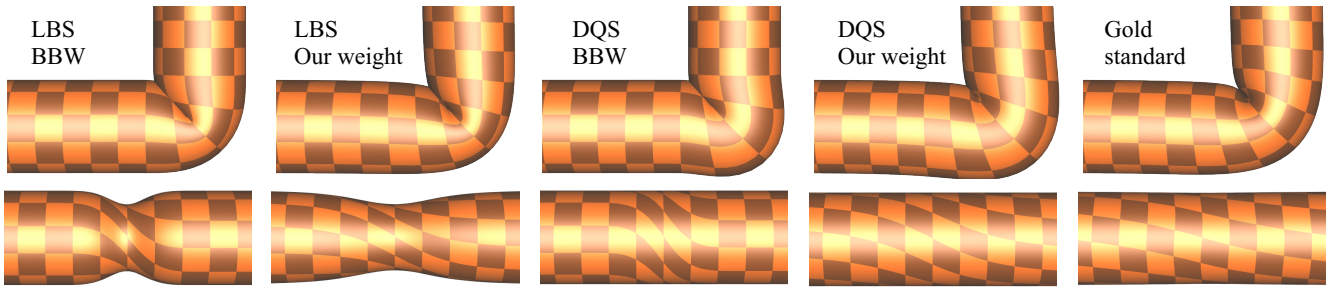

Figure 3: *Comparison of linear blend skinning (LBS) with bounded bi-harmonic weights (BBW) and our weights to dual quaternion skinning (DQS) and a gold standard (full energy minimization); 90 degrees bend (top row) and 90 degrees twist (bottom row). While our weights do better, the bend-bulging artifacts of DQS and the candy-wrapper artifacts of LBS are still visible even at these non-extreme rotations.*

embarrassingly parallel and we employ the highly optimized 3×3 SVD routines developed by McAdams and colleagues [2011]. In the global step ($\mathbf{r}_1, \ldots, \mathbf{r}_k$ fixed), we minimize a quadratic objective subject to linear equality constraints using PARDISO [Schenk and Gartner 2006], taking advantage of prefactorization, since the system matrix does not change between iterations. In our examples, the block coordinate descent typically converges in less than a couple dozen iterations. In comparison, a straightforward l_2 fit, obtained by first computing the energy-minimizing deformations $\mathbf{v}_i$ using Eq. 1 and then solving for $\mathbf{w}$ to minimize the l_2 error $\|\mathbf{v}_i - \mathbf{M}_i\mathbf{w}\|$ requires more iterations to converge. Also, each iteration is slower and the final weight fitting introduces additional projection error (see Fig. 13 and further discussion in Sec. 4).

Finding optimal weights for dual quaternion skinning is slightly more complicated than for LBS because the skinning transformation is nonlinear, i.e., instead of $\mathbf{v}_i = \mathbf{M}_i\mathbf{w}$, we have $\mathbf{v}_i = \Psi_i(\mathbf{w})$. This leads to the following objective:

$$\sum_{i=1}^{k} \left(\frac{1}{2} \Psi_i(\mathbf{w})^\mathsf{T} \mathbf{L} \Psi_i(\mathbf{w}) - \min_{\mathbf{r}_i \in \mathcal{Q}_m} (\mathbf{r}_i^\mathsf{T} \mathbf{K} \Psi_i(\mathbf{w})) \right) \qquad (4)$$

We can adopt the same local/global strategy as before. The local step remains unchanged, but in the global step we no longer have a quadratic objective. Therefore, we employ a nonlinear solver [Byrd et al. 2006], taking advantage of the fact that the derivatives of DQS are easy to compute analytically [Kavan et al. 2009]. Unfortunately, even with energy-minimizing weights, we observe that neither LBS nor DQS achieve deformations resembling minimization of the original energy (Eq. 1), see Fig. 3. Even when optimizing our weights separately for all four cases (bending/twisting, LBS/DQS), we can see an improvement over bounded biharmonic weights, but the candy-wrapper artifacts of LBS, as well as the bend-bulging artifacts of DQS are still visible (Fig. 3).

The bend-bulging is caused by the fact that dual quaternions transform all vertices with the same weight rigidly, and therefore the natural squish of an elastic material cannot be reproduced. For example, let us consider the middle cross-section of the bending cylinder from Fig. 3, i.e., all vertices with both weights equal 0.5. Dual quaternions cannot squish this circle while bending, because all of its vertices are transformed by one rigid body transformation, leading to the bulging artifact. Similarly, LBS cannot avoid shrinking this circle while twisting, leading to the candy wrapper artifact. To address this issue, software packages such as Autodesk Maya™ support linear blending of LBS and DQS. We also experimented with this approach as well as with several alternatives, specifically, non-normalized DQS and Animation Space [Merry et al. 2006]. Unfortunately, we were unable to avoid all artifacts.

We argue that the problems stem from the isotropic nature of transformation blending, i.e., treating all directions as equal, which makes it impossible to differentiate between the rather rigid direction implied by the bone and the much more supple directions orthogonal to the bone. In the following section we discuss how to overcome this limitation in a general way while retaining the standard skeletal animation framework and the rigid-bones metaphor.

3.3 Joint-based Deformers

This section presents the general concept of joint-based deformers, as an alternative to the conventional "bone-based" deformers, such as LBS or DQS. In the interest of generality, this section does not depend on the energy-minimizing weights (Sec. 3.2). Our specific joint-based deformer, utilizing the energy-minimizing weights, will be discussed in Sec. 3.4.

The basic idea is to associate with each joint a quickly computable function (*deformer*) that describes deformations in the vicinity of this joint. By definition, a joint-based deformer is a mapping $\Gamma : SO(3) \times \mathbb{R}^3 \to \mathbb{R}^3$, which for any given rotation $\mathbf{Q}$ and point $\mathbf{x}$ returns a deformed point $\Gamma(\mathbf{Q}, \mathbf{x})$. It is helpful to study the joint-based deformers in a common canonical coordinate system. Specifically, we choose the joint location to coincide with the origin and the z-axis oriented in the direction of the child bone (Fig. 4).

Figure 4: *Local coordinate systems and weights d_j of our joint-based deformers (here the shoulder and the elbow).*

To apply joint-based deformers in skinning, we need to embed them in the standard forward kinematics framework. We assume the local coordinate frames of all joints are chosen according to our convention, i.e., with the z-axis oriented in the child bone direction. Let $\mathbf{R}_j$ represent the rest-pose transformation of joint j, $\mathbf{Q}_j$ its current rotation, and $\mathbf{A}_j$ the composed transformation in the animated pose, such that $\mathbf{A}_j\mathbf{Q}_j$ is the transformation of joint j in the current pose. Then, $\mathbf{A}_j\mathbf{Q}_j\mathbf{R}_j^{-1}$ corresponds to the standard skinning transformation (input of linear or dual quaternion skinning). The composition of a joint-based deformer Γ_j with a kinematic skeleton can be then expressed as $\mathbf{A}_j\Gamma_j(\mathbf{Q}_j, \mathbf{R}_j^{-1}\tilde{\mathbf{p}})$, where $\tilde{\mathbf{p}} \in \mathbb{R}^3$ is the rest-pose location of a surface mesh vertex. Because each deformer is only accurate in the vicinity of its joint, we obtain the final result by taking a linear combination:

$$\mathbf{p} = \sum_{j=1}^{D} d_j(\tilde{\mathbf{p}}) \mathbf{A}_j \Gamma_j(\mathbf{Q}_j, \mathbf{R}_j^{-1}\tilde{\mathbf{p}}) \qquad (5)$$

where $d_j : \mathbb{R}^3 \to [0,1]$ are partition-of-unity deformer weights describing the influence of deformer Γ_j on each point in space and D is the number of deformers.

By using trivial deformers $\Gamma_j(\mathbf{Q}_j, \mathbf{x}) = \mathbf{Q}_j\mathbf{x}$, we obtain the standard linear blend skinning, which completely relies on the weights to control the resulting deformations. In our approach, we instead control the deformations near joints directly using Γ_j, with weights d_j only taking care of smooth transitioning from one deformer (e.g., shoulder) to another (e.g., elbow). The blending therefore occurs near the middle of a bone, in contrast to standard skinning where most blending happens near the joints. Because bones are typically rigid, the deformations induced by adjacent joint-based deformers are usually fairly similar near the middle of the bone, and therefore the choice of blending weights d_j is much less critical than with linear blend skinning. We use point-based bounded biharmonic weights [Jacobson et al. 2011] located at joints (Fig. 4).

3.4 Swing/Twist Deformer

We seek the simplest possible joint-based deformer that avoids the issues with linear and dual quaternion skinning. Observing that LBS works well for bending and DQS nicely handles twisting (Fig. 3), we design a joint-based deformer to combine their strengths. Assuming our canonical coordinate system, our deformer first decomposes the input rotation $\mathbf{Q}$ into a swing (rotation about an axis in the xy-plane) and a twist (rotation about the z-axis, corresponding to the child bone direction), such that $\mathbf{Q} = \mathbf{Q}_{\text{swing}}\,\mathbf{Q}_{\text{twist}}$. This decomposition is a simple closed-form operation and in quaternion representation requires only few arithmetic operations [Baerlocher 2001].

Let us assume we have two skinning weights, $s : \mathbb{R}^3 \to [0,1]$ for the swing (bending) distribution and $t : \mathbb{R}^3 \to [0,1]$ for the twist. For a given mesh vertex expressed in local joint coordinates $\tilde{\mathbf{x}} = \mathbf{R}_j^{-1}\tilde{\mathbf{p}}$, the result of our swing/twist (sw/tw) deformer is:

$$\Gamma^{\text{sw/tw}}(\mathbf{Q}, \tilde{\mathbf{x}}) = \Big((1 - s(\tilde{\mathbf{x}}))\ \mathbf{I} + s(\tilde{\mathbf{x}})\mathbf{Q}_{\text{swing}} \Big) \mathbf{Q}_{\text{twist}}^{t(\tilde{\mathbf{x}})}\, \tilde{\mathbf{x}} \quad (6)$$

This equation first applies spherical interpolation of the twist, followed by linear interpolation of the swing. Because $\mathbf{Q}_{\text{twist}}$ is always a rotation about the z-axis, we only need a simple 2D rotation interpolation (instead of general SLERP). We precompute the swing weight s as our energy-minimizing weight (Sec. 3.2) for linear blend skinning, with rotation samples covering swing rotations only. Specifically, we use just 4 sample rotations $\mathbf{S}_1, \ldots, \mathbf{S}_4$, corresponding to ± 90 degrees rotations in x- and y-axes. We found this simple set of samples works surprisingly well. The twist weight t can be computed with only 2 rotation samples representing ± 90 degrees rotation about the z-axis. While the swing/twist deformer could also be used with bounded biharmonic weights (for both s and t), the elastic energy-minimizing weights minimize shape distortion (Fig. 5).

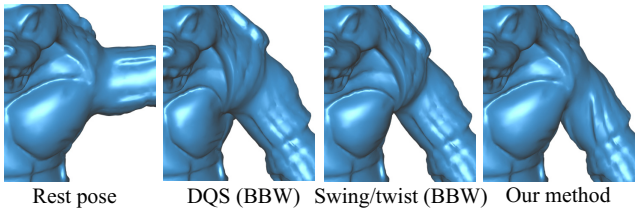

Figure 5: *The swing/twist deformer improves upon dual quaternion skinning (DQS) even with bounded biharmonic weights (BBW). Our method combines the swing/twist deformer with elastic energy-minimizing weights, resulting in minimal shape distortion.*

3.5 Final Algorithm

In this section we summarize our resulting method based on swing/twist deformers with energy-minimizing weights (Alg. 1). Note that we compute the bounded biharmonic weights (BBW) using the same regular grid as in our method, because the tetrahedralization utilized by Jacobson and colleagues [2011] is not directly applicable to meshes containing self-intersections or non-manifold edges. We produce our voxelization using Min's implementation [2012] of Nooruddin and Turk's method [2003], which is very efficient and robustly handles imperfect meshes.

Algorithm 1

Input: Rest-pose mesh, skeleton, voxel size (resolution).
Output: Vertex weights d_j, s_j, t_j for each joint j.

```
Establish voxel grid G [Min 2012]
T = trilinear interpolation from G vertices to mesh vertices
for each joint j do
    d̃_j = Point-based BBW(G) [Jacobson et al. 2011]
    s̃_j = EMW(G; R_x(−90), R_x(90), R_y(−90), R_y(90))
    // EMW ... Energy-Minimizing Weights (Sec. 3.2)
    // R_axis(α) ∈ SO(3) ... sample rotations
    t̃_j = EMW(G; R_z(−90), R_z(90))
    d_j = T d̃_j,  s_j = T s̃_j,  t_j = T t̃_j
end for
```

With the precomputed weights, the runtime component is simple and fast. We first evaluate the swing/twist deformers $\Gamma_j^{\text{sw/tw}}$ for all joints j, using the weights s_j and t_j (Eq. 6). Subsequently, we blend the results using the deformer weights d_j (Eq. 5), obtaining the final deformed vertex positions.

4 Results

In this section, as well as in the accompanying video, we compare our final method (Sec. 3.5) to bounded biharmonic weights (BBW) [Jacobson et al. 2011] with both linear blend skinning (LBS) and dual quaternion skinning (DQS).

Deformation quality. We distinguish three main types of artifacts. The first is the well-known candy-wrapper artifact of LBS (Fig. 6). DQS removes the candy-wrapper artifacts by replacing linear trajectories with spherical ones, which introduces second type of artifact, i.e., bulging while bending (Fig. 7). These two kinds of artifacts can also occur simultaneously when both bending and twisting are present (Fig. 1). Our swing/twist deformer successfully removes both the candy-wrapper and the bend-bulging artifacts.

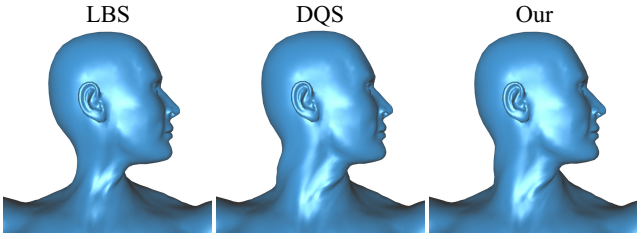

Figure 6: *The candy-wrapper artifacts of linear blend skinning do not occur in dual quaternion skinning or our method.*

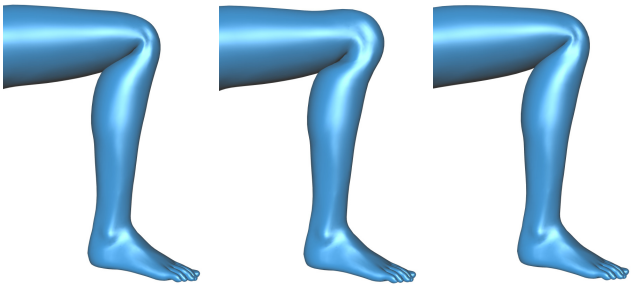

Figure 7: *A bulging artifact of dual quaternion skinning does not occur with linear blend skinning or our method.* © *2012 The Authors*

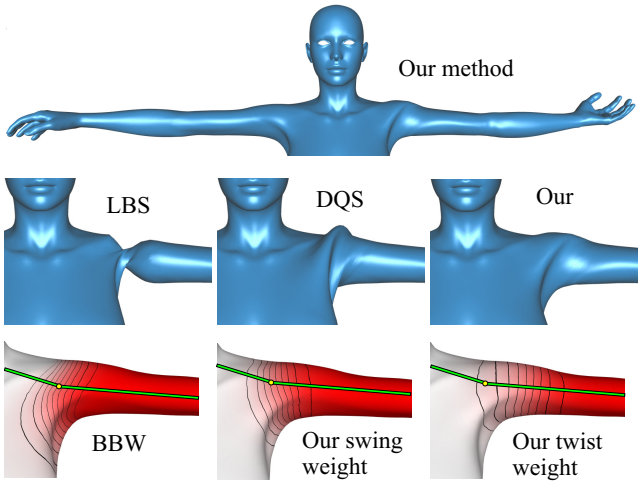

Figure 8: *Our method robustly handles large shoulder rotations, while dual quaternions look unnatural due to overly large influence of the corresponding bounded biharmonic weight (BBW). Our twist weight results in more natural torsion distribution.* © *2012 The Authors*

The third kind of artifact stems from deformation-agnostic weight distribution, apparent with bounded biharmonic weights (Fig. 1 left, Fig. 5, and Fig. 8). Our weights minimize elastic deformation energy and therefore also the shape distortion. Another example is shown in Fig. 9. The biharmonic operator diffuses uniformly in all directions, and thus both the calf and the foot bones influence the heel by approximately the same amount, resulting in unsightly deformations of the heel. Our energy-minimizing weight rigidly attaches the heel to the foot, producing more natural results. In some situations, a too generous spread of bounded biharmonic weights exacerbates the dual quaternion bulging issues (Fig. 10). The combination of swing/twist deformers with elastic energy-minimizing weights produces natural results in all cases.

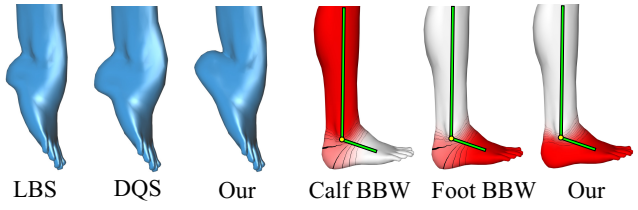

Figure 9: *Bounded biharmonic weights (BBW) used with LBS and DQS overlap in the heel region, resulting in unnatural deformations. Our weights correctly concentrate the bending in the ankle region.* © *2012 The Authors*

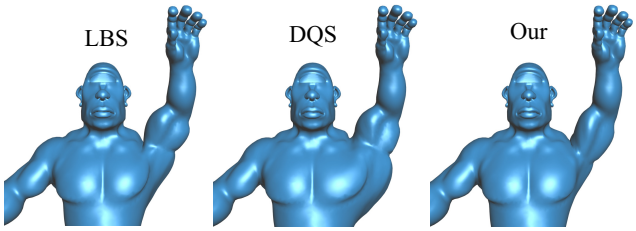

Figure 10: *The bulging artifact of dual quaternion skinning is amplified by the relatively large support of the corresponding weight.* © *2012 The Authors*

A nice property of our energy-minimizing weights is that the solution is not very sensitive to the density of the joint rotations sampling. For given axes of rotation, increasing the number of samples yields almost the same results (Fig. 11). While the twist weight optimization should in theory account for the skinning nonlinearity (Eq. 4), in practice we used only the much faster LBS scheme (Eq. 3) to generate our results, except for the bending cylinder in Fig. 3. We observed that the linear model typically converges to very similar weights as the slower nonlinear scheme.

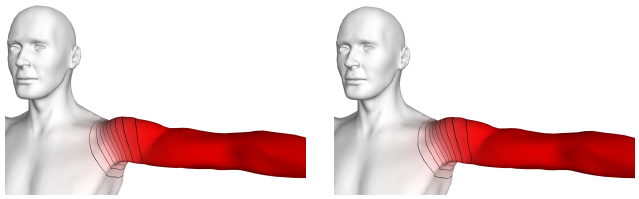

Figure 11: *Optimized bending weight using 4 samples (left) is almost identical to one using 32 samples (right).* © *2012 The Authors*

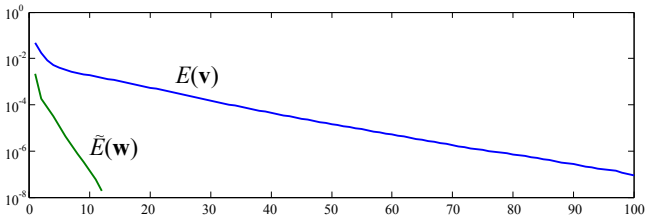

Figure 12: *Relative error (y-axis) of the full nonlinear elastic energy $E(\mathbf{v})$ and the reduced energy $\tilde{E}(\mathbf{w})$ vs. number of iterations.* © *2012 The Authors*

As mentioned in Sec. 3.2, an obvious alternative to our energy-minimizing weights would be to first precompute example shapes using non-reduced energy minimization, followed by a least-squares (l_2) fit of skinning weights [James and Twigg 2005]. However, this approach is sub-optimal in all respects: direct minimization of $E(\mathbf{v})$ (Eq. 1) requires more iterations to converge than the minimization of $\tilde{E}(\mathbf{w})$ (Eq. 3), see Fig. 12. Furthermore, each iteration is more expensive because the global step contains more variables and has to be evaluated k times, whereas in Eq. 3 all samples collapse to one matrix $\mathbf{Y}$. Finally, the subsequent least-squares fit introduces additional l_2 projection error, which can be perceptibly non-smooth even when adding non-negativity constraints as suggested by James and Twigg [2005] (Fig. 13). Our method does not use any inequality constraints, but we still obtain non-negative and visually smooth weights in all our test cases. However, we do not have theoretical guarantees of these properties.

Performance. Experimental evaluation of runtime performance is challenging, because the implementation specifics and hardware features of each target platform often impact the resulting speed more than the skinning algorithm itself [Frey and Herzeg 2011].

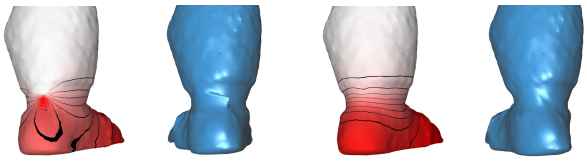

Figure 13: *A non-negative least-squares fit (left) does not guarantee smooth weights and results in a pinching artifact. Our weight (right) directly minimizes the deformation energy and avoids this problem.*

Examining Eq. 5, we see that the only overhead compared to linear blend skinning comes from evaluating the deformers. The local coordinates $\tilde{\mathbf{x}} = \mathbf{R}_j^{-1}\tilde{\mathbf{p}}$ are constant and thus can be precomputed. Because the number of deformers is much lower than the number of vertices, the time to perform the swing/twist decompositions is negligible and the main overhead comes from evaluating Eq. 6. Assuming the twist is represented by an angle, this formula requires 29 elementary arithmetic operations and 2 trigonometric functions (for the twist rotation). The trigonometric functions can be optionally replaced by the complex number analogue of SLERP, requiring only reciprocal square root. The memory overhead consists of storing the additional swing and twist weights per joint. It would be possible to optimize only one weight to simultaneously control both bending and torsion, however the optimal swing and twist weights generally differ (Fig. 8).

As can be seen in Tab. 1, the preprocessing time is typically dominated by solving the bounded biharmonic weights d for deformer blending (Fig. 4), computed using MOSEK [Andersen and Andersen 2000]. Precomputation of our energy-minimizing weights is faster because we do not use any inequality constraints. The voxel grid resolution was selected manually for each model in order to balance performance vs. accuracy.

	Verts	Tris	n	d	(s,t)	Total
Armadillo	43,243	86,482	40,725	46s	23s	18min
Female	45,659	91,208	20,425	11s	12s	6min
Male	16,093	32,182	44,820	37s	24s	20min
Ogre	28,670	57,336	33,973	29s	19s	13min

Table 1: *Model statistics: n is the number of voxel vertices, d is the time to precompute a bounded biharmonic weight per joint, (s,t) is the time to precompute both swing and twist energy-minimizing weights per joint, and* Total *is the total precomputation time.*

5 Conclusions and Future Work

We presented a conceptually new method for skinning articulated shapes motivated by approximating the behavior of elastic solids with embedded rigid bones. All our examples were obtained fully automatically, up to user-specified voxel resolution. Determining suitable voxel size automatically would be an interesting problem for future work. While our final method uses exclusively the swing/twist deformer (Sec. 3.4), our framework (Sec. 3.3) is general and allows a variety of different deformers to be explored and controlled by artists wishing to fine-tune the result. For example, if the joint-collapse or bulging effects are actually desirable, we can trivially design deformers implementing linear or dual quaternion skinning and use them in chosen parts of the model. More interestingly, we could design a deformer that, e.g. nonlinearly blends between linear and spherical skinning based on the angle between parent and child bones, so that bulging gradually increases as the joint rotates. In future work, we would like to explore the space of meaningful joint-based deformers and provide intuitive interfaces to the users. One limitation is we assume only linear combinations of the individual deformers, ignoring nonlinear joint coupling effects.

Unlike physics-based elasticity [McAdams et al. 2011], our system cannot guarantee self-intersection free deformations. Correct handling of collisions and contact without relying on runtime simulation is an interesting challenge for future work. In this paper, we focused on the overall, high-level skeletal shape deformations. Our method could be enhanced with example-based details and/or secondary dynamics using the well-known techniques [Lewis et al. 2000; Shi et al. 2008]. One technical aspect to look at is to consider integration over a continuous range of motion of the joint instead of using a discrete set of sample rotations. This would involve solving a finite-elements version of the Procrustes problem. Finally, we would also like to explore the effect of more accurate material models and actual volumetric bones on the resulting weights. In conclusion, we hope our method will help to bridge the gap between the accurate yet slow physics-based simulation and the simple and fast closed-form skinning methods.

6 Acknowledgements

We are grateful to Chris Evans and John Howe for art and rigging feedback. We thank Eftychios Sifakis for his open source fast 3×3 SVD code and Alec Jacobson, Martin Pražák, Stelian Coros, and Bernhard Thomaszewski for many useful discussions. We also thank Emily Whiting for her narration of the accompanying video and Kenshi Takayama and Katie Bassett for proofreading. This work was supported in part by an SNF award 200021-137879.

References

ANDERSEN, E. D., AND ANDERSEN, K. D. 2000. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*. Kluwer Academic Publishers, 197–232.

ANGELIDIS, A., AND SINGH, K. 2007. Kinodynamic skinning using volume-preserving deformations. In *Proc. Symp. Comput. Animation*, 129–140.

BAERLOCHER, P. 2001. *Inverse kinematics techniques for interactive posture control of articulated figures*. PhD thesis, EPFL.

BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3D characters. *ACM Trans. Graph. 26*, 3, 72.

BARBIČ, J., AND ZHAO, Y. 2011. Real-time large-deformation substructuring. *ACM Trans. Graph. 30*, 4, 91:1–91:7.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Trans. Vis. Comput. Graph. 14*, 1, 213–230.

BYRD, R. H., NOCEDAL, J., AND WALTZ, R. A. 2006. KNITRO: An integrated package for nonlinear optimization. In *Large Scale Nonlinear Optimization*, Springer Verlag, 35–59.

CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph. 21*, 3, 586–593.

CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2005. Physically based rigging for deformable characters. In *Proc. Symp. Comput. Animation*, 301–310.

CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph. 29*, 4, 38:1–38:6.

CHEN, C.-H., LIN, I.-C., TSAI, M.-H., AND LU, P.-H. 2011. Lattice-based skinning and deformation for real-time skeleton-driven animation. In *Proc. CAD/GRAPHICS*, 306–312.

FORSTMANN, S., OHYA, J., KROHN-GRIMBERGHE, A., AND MCDOUGALL, R. 2007. Deformation styles for spline-based skeletal animation. In *Proc. Symp. Comput. Animation*, 141–150.

FREY, I. Z., AND HERZEG, I. 2011. Spherical skinning with dual quaternions and QTangents. In *ACM SIGGRAPH Talks*, 11:1–11:1.

GREGORY, A., AND WESTON, D. 2008. Offset curve deformation from skeletal animation. In *ACM SIGGRAPH Talks*, 57:1–57:1.

HYUN, D.-E., YOON, S.-H., CHANG, J.-W., SEONG, J.-K., KIM, M.-S., AND JÜTTLER, B. 2005. Sweep-based human deformation. *The Visual Computer 21*, 8-10, 542–550.

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

JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4, 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, 77:1–77:10.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. Graph. 24*, 3, 399–407.

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

JU, T., ZHOU, Q.-Y., VAN DE PANNE, M., COHEN-OR, D., AND NEUMANN, U. 2008. Reusable skinning templates using cage-based deformations. *ACM Trans. Graph. 27*, 5, 122:1–122:10.

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

KAVAN, L., COLLINS, S., AND O'SULLIVAN, C. 2009. Automatic linearization of nonlinear skinning. In *Proc. Symp. on Interactive 3D Graphics and Games*, 49–56.

KHO, Y., AND GARLAND, M. 2005. Sketching mesh deformations. In *Proc. Symp. on Interactive 3D Graphics and Games*, 147–154.

KIM, T., AND JAMES, D. L. 2011. Physics-based character skinning using multi-domain subspace deformations. In *Proc. Symp. Comput. Animation*, 63–72.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. EigenSkin: real time large deformation character skinning in hardware. In *Proc. Symp. Comput. 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, 99:1–99:17.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. ACM SIGGRAPH*, 165–172.

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

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*, 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, 1400–1423.

MIN, P., 2012. Binvox. `http://www.cs.princeton.edu/~min/binvox/`.

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

MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. *ACM Trans. Graph. 24*, 3, 471–478.

NOORUDDIN, F., AND TURK, G. 2003. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph. 9*, 2, 191–205.

RIVERS, A., AND JAMES, D. 2007. FastLSM: fast lattice shape matching for robust real-time deformation. *ACM Trans. Graph. 26*, 3.

ROHMER, D., HAHMANN, S., AND CANI, M.-P. 2009. Exact volume preserving skinning with shape control. In *Proc. Symp. Comput. Animation*, 83–92.

SCHENK, O., AND GARTNER, K. 2006. On fast factorization pivoting methods for symmetric indefinite systems. *Elec. Trans. Numer. Anal. 23*, 58–179.

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2008. Example-based dynamic skinning in real time. *ACM Trans. Graph. 27*, 3, 29:1–29:8.

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

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

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proc. Symp. Comput. Animation*, 129–138.

WANG, R. Y., PULLI, K., AND POPOVIĆ, J. 2007. Real-time enveloping with rotational regression. *ACM Trans. Graph. 26*, 3, 73:1–73:9.

WAREHAM, R., AND LASENBY, J. 2008. Bone Glow: An improved method for the assignment of weights for mesh deformation. *Articulated Motion and Deformable Objects*, 63–71.

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, 281–292.