Implicit Skinning: Real-Time Skin Deformation with Contact Modeling

Rodolphe Vaillant[1,2], Loïc Barthe[1], Gaël Guennebaud[3], Marie-Paule Cani[4],
Damien Rohmer[5], Brian Wyvill[6], Olivier Gourmel[1], Mathias Paulin[1]

[1]IRIT - Université de Toulouse, [2] University of Victoria, [3]Inria Bordeaux,
[4] LJK - Grenoble Universités - Inria, [5] CPE Lyon - Inria, [6] University of Bath

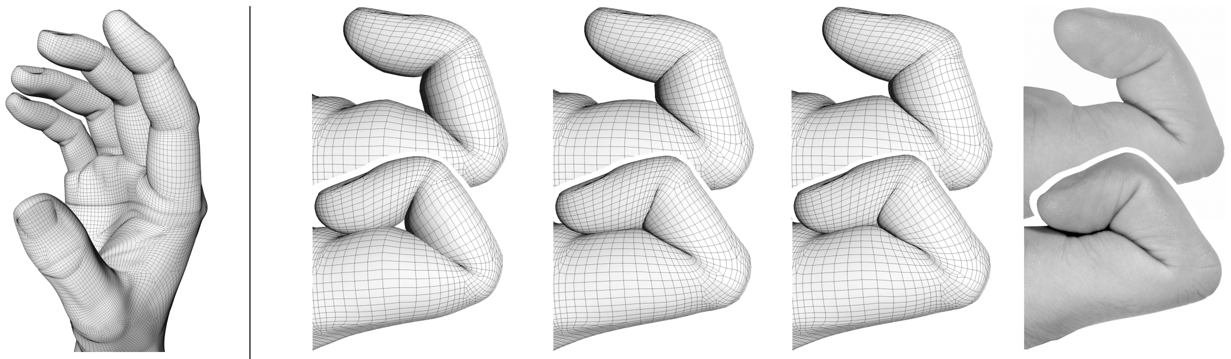

(a) (b) Linear blending (c) Our with contact (d) Our with bulge (e) Real finger

Figure 1: *(a) Real-time animation of the index finger of a hand model composed of 31750 vertices. Two poses are shown in each column: (b) standard smooth skinning with linear blending at* 95 *frames per second (fps), (c) our method which compensates the loss of volume on top of joints and models contact in the fold at* 37 *fps, (d) our method with an additional bulge mimicking tissues inflation at* 35 *fps, and (e) a picture of a real bent finger.*

Abstract

Geometric skinning techniques, such as smooth blending or dual-quaternions, are very popular in the industry for their high performances, but fail to mimic realistic deformations. Other methods make use of physical simulation or control volume to better capture the skin behavior, yet they cannot deliver real-time feedback. In this paper, we present the first purely geometric method handling skin contact effects and muscular bulges in real-time. The insight is to exploit the advanced composition mechanism of volumetric, implicit representations for correcting the results of geometric skinning techniques. The mesh is first approximated by a set of implicit surfaces. At each animation step, these surfaces are combined in real-time and used to adjust the position of mesh vertices, starting from their smooth skinning position. This deformation step is done without any loss of detail and seamlessly handles contacts between skin parts. As it acts as a post-process, our method fits well into the standard animation pipeline. Moreover, it requires no intensive computation step such as collision detection, and therefore provides real-time performances.

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

Keywords: Skinning, Mesh deformation with contact

Links: DL PDF

1 Introduction

One of the main challenges when animating a virtual character is the production of plausible skin deformations at joints (Figure 1). This animation step is called *skinning*. Popular methods include linear blending skinning (LBS, also called smooth skinning) and dual quaternions [Magnenat-Thalmann et al. 1988; Kavan et al. 2008]. Such *geometric skinning* techniques give the artist real-time interaction with the 3D model while only requiring limited user input, especially when automatic rigging techniques are used [Baran and Popović 2007; Jacobson et al. 2011; Bharaj et al. 2011]. However, the resulting deformation does not maintain a constant volume, and may cause local self-penetrations, thus failing to produce convincing organic-like deformations near joints (Figure 1(b)).

Example-based methods [Mohr and Gleicher 2003] and shape interpolation schemes [Lewis et al. 2000] were proposed to achieve more realism while still enabling real-time animation. However, they increase the amount of necessary input and often require tedious work from the artists. Furthermore, they are limited to a given range of deformations, which usually does not include deformations due to contact with other skin parts.

Lastly, physical simulation can also produce appealing deformations [Ng-Thow-Hing 2001; Teran et al. 2005]. However, the amount of computation makes it only suitable for off-line rendering. In addition, muscles and rigid bones of realistic shapes need to be predefined from medical data or by the user. Intermediate methods such as elasticity skinning [McAdams et al. 2011] enables the computation of skin squash and stretch by solving the underlying physical equations. While robust and visually plausible, it still requires several seconds per animation frame.

Our goal is to provide the artist with a real-time skinning technique that automatically produces self-penetration free deformations when skin folds, while preserving the aspect of a rigid bone near joints (Figure 1(c)). We also want to be able to generate subtle effects such as bulges (Figure 1(d)).

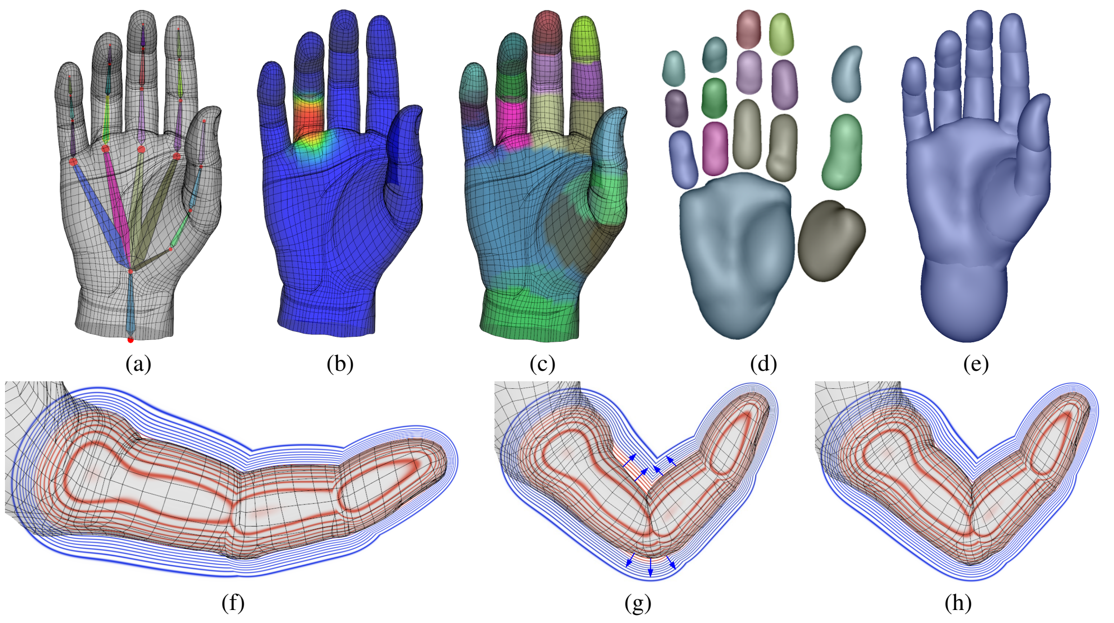


Figure 2: ***Overview.*** *(a) An input mesh with its animation skeleton, (b) deformation weights at a joint, and (c) mesh segmentation. (d) Implicit surfaces computed as* 0.5*-isosurfaces of HRBFs approximating each part of the mesh. (e) Composition with a union operator, and resulting shape. (f) Each vertex is assigned to a field value in the rest pose. (g) During animation, both the mesh and the field functions are transformed and mesh vertices march along the field gradient until they reach their individual iso-value, unless they reach a gradient discontinuity modeling some contact region. (h) This produces the final skinned mesh. In (f), (g) and (h), the isosurfaces of the field function are plotted in a vertical plane centered on the skeleton. In blue, the outside values (*$f < 0.5$*) and in red the inside values (*$f > 0.5$*).*

To this end, we start from an initial mesh deformation produced by standard geometric skinning. The key idea is to approximate the individual mesh parts attached to each skeleton bone using implicit surfaces, and to exploit their ability to be combined using operators such as union, gradient-based blending or gradient-based bulge-in-contact [Gourmel et al. 2013]. Combining these operators enables us to generate plausible skin deformations at joints during motion, which is done by projecting each mesh vertex onto the adequate iso-surface, from its smooth skinning position. In addition to a novel skinning approach, our main technical contributions are:

- A specific reconstruction method for mesh parts, based on Hermite Radial Basis Functions (HRBF) [Wendland 2005].
- A dedicated blending operator for modeling skinning effects.
- A fast marching algorithm for mesh vertices in a scalar field, used in conjunction with a specific mesh relaxation method avoiding the introduction of large mesh distortions.

The resulting technique generates visually plausible skin deformations in real-time. Our method automatically generates contact surfaces between skin parts, without requiring any collision detection step. Finally, it lies on top of standard geometric skinning and can therefore be easily implemented within a standard animation pipeline.

2 Related work

Skeleton driven skin deformation was first introduced by the study of a hand model [Magnenat-Thalmann et al. 1988]. This linear blend skinning (LBS) technique combines joint transformations based on skinning weights associated to each bone. Its limitations have been extensively studied, and include the candy wrapper effect (mesh shrinkage when bones twist) and the elbow collapse effect (loss of volume at a bone join).

Many example based techniques were developed as an alternative [Lewis et al. 2000; Sloan et al. 2001; Kry et al. 2002; Wang and Phillips 2002; Mohr and Gleicher 2003; Wang et al. 2007; Weber et al. 2007]. They achieve realistic results but require the production of several static meshes with the necessary key poses as input in order to provide the desired deformation.

More advanced geometrical skinning methods were introduced to limit the artifacts while keeping their simplicity of use. They relied on matrices interpolation [Magnenat-Thalmann et al. 2004; Kavan and Žára 2005], dual quaternion deformation [Kavan et al. 2008], or fast optimization processes [Kavan et al. 2009; Shi et al. 2007], while possibly allowing bone scaling [Jacobson and Sorkine 2011]. These methods remain however prone to the elbow collapse effect and only produce smooth deformations such as those illustrated in Figures 1(b) and 10. Displacement textures encoded along spline curves were used to generate organic-like deformation [Forstmann et al. 2007]. However, being pre-stored as a texture, the deformation does not adapt to the skinning deformation and does not avoid self-intersections. Kavan and Sorkine [2012] proposed to pre-compute optimized skinning weights for linear skinning and dual quaternions at joints to approximate the skin deformations produced by nonlinear variational deformation methods [McAdams et al. 2011]. Even though this expensive pre-computation produces more convincing real-time deformations, the solution remains too smooth and does not faithfully reproduce bone joints behaviors where the skin stretches around the joint and where it is crushed due to contact.

Volume preserving skinning methods allow users to correct for volume changes through the generation of extra bulges and/or wrinkles [Funck et al. 2008; Rohmer et al. 2009]. However, this approach is computationally expensive, and contacts between different parts of the skin are not handled. Self-collision can be avoided using a volume preserving algorithm based on vector field integration [Angelidis and Singh 2007]. However this method does not fit into the standard animation pipeline, since it requires some temporal integration. Moreover, the parameters controlling the appearance of the final shape may not be intuitive to set-up.

The idea of using implicit representations in conjunction with meshes for character animation is not new. Metaballs [Shen and Thalmann 1995], polygon-based implicit primitives [Singh and Parent 1995], convolution surfaces [Van Overveld and Broek 1999] and ellipsoidal implicit primitives [Leclercq et al. 2001] were used to animate skinned characters, with the ability in the last case to add organic details (muscles) to a coarse mesh representing the characters body. Closer to our work, implicit modeling has been used to improve skinning deformation [Bloomenthal 2002]. The shape is encoded through its medial axis, itself animated through skinning, which limits the candy wrapper and collapsing elbow artifacts. However organic effects such as bulges and contact surfaces are not handled by this approach.

Contrary to previous methods, we do not use implicit surfaces to model the shape of the character, but rather to preserve the original mesh properties, produce controllable deformations through compositing operators, and avoid self-collision during deformation. Meanwhile, surface details always remain encoded by the mesh.

3 Overview

The different steps of our approach are illustrated in Figure 2. As for standard geometric skinning [Magnenat-Thalmann et al. 1988; Kavan et al. 2008], we start from a mesh equipped with an animation skeleton defined as a hierarchy of bones, and the associated skinning weights (Figure 2(a,b)). In addition, the mesh is already partitioned with respect to skeleton bones (Figure 2(c)). This input can either be provided by artists or automatically generated. In this paper we use weights computed with the heat diffusion technique of Baran and Popovic [2007].

From these initial settings, we use Hermite Radial Basis Functions (HRBF) [Wendland 2005; Macêdo et al. 2011] to approximate each part of the mesh with the 0.5-isosurface of a smooth scalar field $f_i : \mathbb{R}^3 \to \mathbb{R}$ (Figure 2(d), Section 4.1). After this step, each vertex $\mathbf{v}$ of the mesh stores its current field value $f_i(\mathbf{v})$ that encodes the details of the mesh. Then, a single field function f is defined from the combination of the f_i using either the union [Ricci 1973], gradient-controlled blending or gradient-controlled bulge operators [Gourmel et al. 2013], depending in the desired result (Figures 2(e,f), Section 4.2). The key idea is to animate simultaneously the mesh with geometric skinning and the field function f by applying rigid transformations to the underlying field functions f_i. The mesh vertices then march following the gradient of f (Figure 2(g)) until they reach their original field value or cross a gradient discontinuity representing a contact surface (Section 5), thus producing the final deformation (Figure 2(h)).

4 Field function construction

In this section, we explain how the field function that closely approximates the mesh and mimics the skin deformation is computed. First, Section 4.1 presents the individual reconstruction of parts of the mesh associated with a given skeleton using an adaptation of HRBF methods. The construction of the final field function by composition of these HRBFs is then detailed in Section 4.2.

4.1 HRBF Reconstruction

Problem setting. Given a bone i and its associated sub-mesh $\mathcal{M}_i$, our goal is to reconstruct a smooth scalar field f_i that tightly approximates $\mathcal{M}_i$. In contrast to the standard surface reconstruction problem, our f_i needs to satisfy several specific constraints.

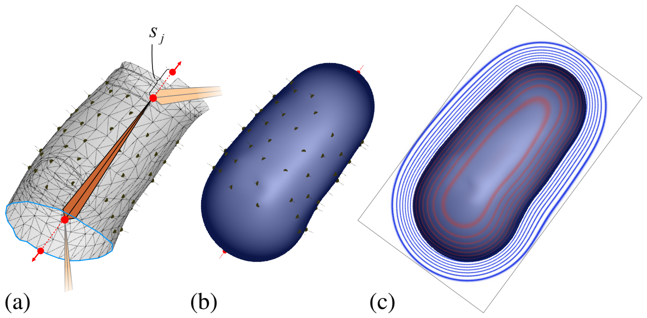


Figure 3: *Reconstruction of a mesh part (a phalanx of the hand shown in Figure 2). (a) HRBF nodes are uniformly spread on the mesh surface, and two additional nodes in red are aligned with the bone to close the holes. (b) The resulting implicit surface, and (c) its associated compactly supported scalar field.*

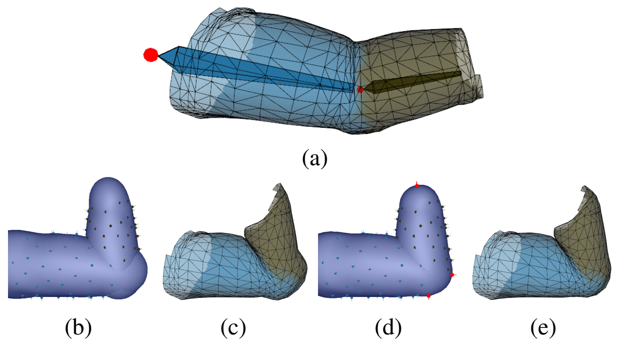


Figure 4: *Two implicit surfaces reconstructed from two phalanxes of the index finger (Figure 1(a)) with a joint in its center shown in (a). (b) When reconstructed without our additional closure constraints (extra points along the bone axis), the junctions overlap, resulting in unexpected bumps. (c) Undesired visual result obtained after projecting the mesh vertices. (d) The additional closure constraints used to solve the problem are shown in red. (e) Resulting mesh after vertex projection: bumps are prevented and the mesh adequately models the joint.*

Firstly, high field smoothness is essential to avoid unexpected behaviors of the gradient-controlled operators (Section 4.2) and to stabilize gradient-descent projection (Section 5) during animation. Smooth fields also require fewer parameters, which reduces their memory footprint and speeds up the fitting and evaluation steps. In order to reach this smoothness while preventing loss of details of the input shape, our insight is to embed the mesh in the resulting scalar field, rather than having it exactly coincide with the 0.5-isosurface: each vertex $\mathbf{v}_j$ stores its local field value $f(\mathbf{v}_j)$ at rest, enabling it to be projected onto its own isovalue during deformation.

Secondly, the field function f_i should appropriately close the large holes left by the mesh partitioning near joints as depicted by the blue curve in Figure 3(a).

Lastly, to allow for local compositions of the f_i, as well as to speed-up the evaluation of f, the field functions should be compactly supported. We also require f_i ranges in $[0, 1]$, with 0.5 being the reference isovalue, and use the convention: $f_i(\mathbf{x}) > 0.5$ if $\mathbf{x}$ is inside, $f_i(\mathbf{x}) < 0.5$ if $\mathbf{x}$ is outside.

Hermite RBF. Given all these constraints, we propose the use of Hermite RBFs [Wendland 2005; Macêdo et al. 2011] to reconstruct a global signed distance field d_i (the mesh being approximated by the zero isosurface). This distance field is then re-parameterized to yield the compactly supported field function f_i.

The HRBF is a variant of RBF that enables to seamlessly interpolating Hermite data (set of positions and normals) without having to populate the whole space with RBFs, nor relying on offset constraint points which can produce reconstruction artifacts [Shen et al. 2004]. In addition, it is scale independent and it robustly reconstructs concavities. More precisely, we seek a distance field d_i of the form:

$$d_i(\mathbf{x}) = \sum_{k=1}^{m} \Big(\lambda_k \phi(\|\mathbf{x} - \mathbf{v}_k\|) + \boldsymbol{\beta}_k^T \nabla \phi(\|\mathbf{x} - \mathbf{v}_k\|) \Big) \ , \tag{1}$$

where the vertices $\mathbf{v}_k$ are the HRBF centers, the scalars λ_k and vectors $\boldsymbol{\beta}_k$ are the unknown coefficients, and ϕ is a smooth function for which we recommend the polyharmonic spline $\phi(\mathbf{x}) = \mathbf{x}^3$. Given a set of m points $\mathbf{v}_k$ with prescribed normals $\mathbf{n}_k$, the $4m$ unknown coefficients are easily found by solving for the system of $4m$ equations: $d_i(\mathbf{v}_k) = 0$ and $\nabla d_i(\mathbf{v}_k) = \mathbf{n}_k$.

Still remains the delicate choice of the HRBF centers. Since we only want to approximate the input sub-mesh $\mathcal{M}_i$ while leaving out the details, a natural choice is to sample the mesh surface with a few samples. To robustly handle arbitrary meshes, we employ a Poisson disk sampling strategy [White et al. 2007] (Figure 3(a,b)). In all our tests, targeting for 50 samples has always been sufficient.

Even though degree three polyharmonic RBFs naturally fill the large holes left at the bone extremities (Figures 4(a,b)), they have to be closed in such a way that the common extremities of two adjacent fields slide over each other without introducing gaps, or creating outgrowths as it is the case in Figure 4(b). Indeed, this would result in a poor skin deformation at joint, as illustrated in Figure 4(c). Ideally, the extremities of two fields at a given joint should be filled with spherical components of the same radius and centered at the joint location, which is impossible in general. Following this observation, we propose to adjust the closure of the distance field as in Figure 4(d) by adding one HRBF center at each extremity along the line supporting the bone i and at a distance s_j from the respective joint j (Figure 3). Their normal constraints are aligned with the bone and point outward. The distance s_j is set as the distance from the joint to the nearest vertex. This allows us to automatically generate a plausible bone joint deformation as shown in Figure 4(e).

Finally, in order to let the HRBFs produce a very smooth surface, samples which are too close to a joint are automatically removed. To this end we employ culling planes orthogonal to the bone and placed at a distance h of the extremities. Formally, let $\mathbf{b}_i^0$ and $\mathbf{b}_i^1$ the two extremities of the bone i, all vertices $\mathbf{v}_k$ that do not satisfy the following criterion are removed:

$$h < \frac{(\mathbf{v}_k - \mathbf{b}_i^0)^T(\mathbf{b}_i^1 - \mathbf{b}_i^0)}{\|\mathbf{b}_i^1 - \mathbf{b}_i^0\|^2} < 1 - h \ . \tag{2}$$

We found $h = 0.05$ to be an effective value in all our results.

Re-parameterization. Finally, the compactly supported field functions f_i we seek are computed using the following remapping: $f_i(\mathbf{x}) = t_r(d_i(\mathbf{x}))$, t_r being defined as:

$$t_r(x) = \begin{cases} 1 & \text{if} \quad x < -r \\ 0 & \text{if} \quad x > r \\ \frac{-3}{16}(\frac{x}{r})^5 + \frac{5}{8}(\frac{x}{r})^3 - \frac{15}{16}(\frac{x}{r}) + \frac{1}{2} & \text{otherwise} \ , \end{cases}$$

where r is a value which sets the size of f_is compact support. The purpose is to convert the infinite support of the HRBF d_i into a finite support field function f_i, while ensuring smoothness at the border of its support. More precisely, t_r is defined as to smoothly map the HRBF values of d_i which are in $[-r, r]$ onto the interval $[0, 1]$, with $t_r(-r) = 1$ and $t'_r(-r) = 0$ (inside the shape), $t_r(0) = 0.5$ (on the surface) and $t_r(r) = t'_r(r) = 0$ (outside). In order to avoid getting constant values $f_i = 1$ and thus null gradients inside the shape, we set r to the distance between the bone and the farthest sampling point used for reconstruction. Then, $f_i = 1$ is only reached on the skeleton and the support size nicely scales with the size of the reconstructed shape.

4.2 Composition into a global field function

Once the individual compactly supported field function f_i are computed, we combine them to define a global field function f fitting the whole mesh to be skinned. To this end, we propose the following two level composition strategy. First, each pair of functions f_i and f_j associated to neighboring skeleton bones are combined using a binary composition operator $g_k : \mathbb{R}^2 \to \mathbb{R}$. The result of this composition is a new field function $\tilde{f}_k = g_k(f_i, f_j)$. Then, the final field f is defined as the union of all combined pair of functions (f_i, f_j) using Ricci's [1973] $\max$ operator:

$$f = \max_k(\tilde{f}_k) \tag{3}$$

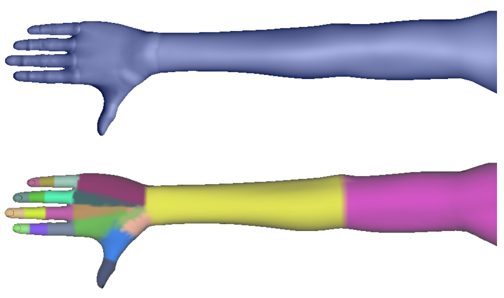
This composition strategy allows us to individually control the deformation behavior for different types of joints, as well as to compensate for the lack of advanced n-ary composition operators.

The choice of operators g_k depends on the desired deformation. For instance, the union operator $\tilde{f}_k = \max(f_i, f_j)$ can be used to generate a fold when the joint articulates while avoiding the elbow collapse effect. This produces the results shown in Figures 1(c), 2(h) and 8(b).

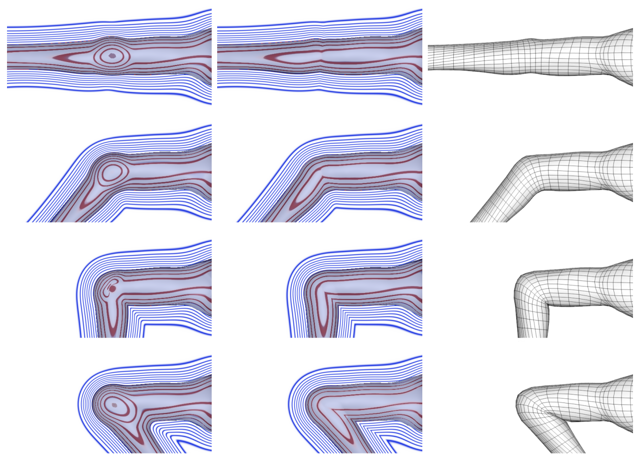
Blending. Smooth blending operators [Blinn 1982; Barthe et al. 2004; Bernhardt et al. 2010] such as the sum and bulge-in-contact operators [Cani 1993] have been popular in implicit modeling. They can be useful to model, respectively, smoother surfaces near joints such as at the elbow of a character, and organic bulging effects with larger contact surfaces as when a finger bends. However, the standard version of these operators would create unwanted bulges at joints, and lead to inadequate surface behaviors when a bone rotates (see Figure 5(b) for smooth blending). The recently introduced gradient-based composition operators [Gourmel et al. 2013] overcome these limitations. They interpolate between union and blending depending on a parameter θ, function of the local angle α between gradients of the composed field functions f_i and f_j at any point $\mathbf{p}$ of the ambient space. θ is thus defined as $\theta(\alpha)$ and a union is performed wherever $\theta = \theta_u$, a smooth blend where $\theta = \theta_b$, and an intermediate local blend when θ varies in $]\theta_b, \theta_u[$.

In our case the definition of functions $\theta(\alpha)$ is made easier by observing that the angle α between field gradients in the folding region is often close to the value of the rotation angle between the bones associated to f_i and f_j (see Figure 6(a)).

Let us take the example of joints with quite smooth skin shapes for small angles, as the ones at the elbow or at the knee of a character (as Juna's elbow in Figure 5(a,d)). Setting $\theta(0) = \theta_u$ (union), we avoid blending when f_i and f_j are aligned, i.e. when their gradients are collinear or when their bones are aligned (Figure 5(c) 1^{st} row), which avoids the unwanted bulge. Then, setting $\theta(\pi/4) = \theta_b$ (blending), we generate a maximum blend between f_i and f_j at a position that would correspond to, for instance, a slight flexion of



(a)

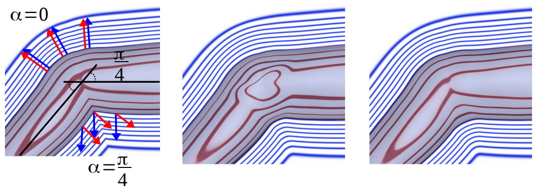


(b) (c) (d)

Figure 5: *(a) Arm of the Juna model (Figure 12(a)) and its partitioning. (b,c) Plot of a plane section of the field function $\tilde{f}$ produced by two different compositions of the two field functions reconstructing the arm around the elbow at different rotation angle: (b) With a standard blending and (c) with our new gradient-based blending. (d) The resulting mesh after vertex projections when the gradient-based composition shown in (c) is used.*

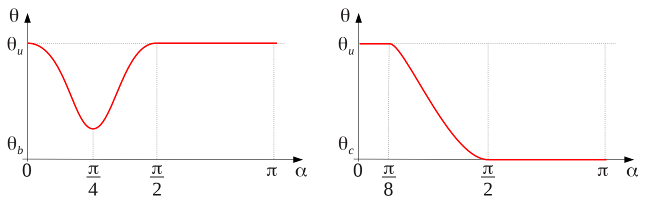
an arm at the elbow, or of a leg at the knee (Figure 5(c) 2^{nd} row). Observing that at elbow or knee, the composition should be back to a union (with a sharp crease of the skin at the joint) when $\alpha = \pi/2$ (Figure 5(c) 3^{rd} row) and then remain a union up to the maximal flexion, we also set $\theta(\alpha) = \theta_u$ for $\alpha \in [\pi/2, \pi]$ (Figure 5(c) 4^{th} row). This specific setting, designed for knees and elbows, leads us to the plot of $\theta(\alpha)$ given in Figure 7(a). The corresponding operator equation is computed from the piecewise definition given in [Gourmel et al. 2013].

Unfortunately, the original gradient-based blending operator [Gourmel et al. 2013] cannot directly be used for our purpose. Indeed, while the 0.5-isosurface of the composed field f roughly represents the skin, its gradient will be used at each animation step for projecting mesh vertices back onto their original iso-values (Section 5). Since geometric skinning shrinks the mesh during deformation, getting adequate gradient vectors inside the shape is especially important. Unfortunately, the existing gradient-based blending operator exhibits an inner depression around the joints, as depicted in Figure 6(b). We thus modified the blending operator to produce an inside-field well suited for the march of the vertices during projection. The equation of our new gradient-based blending operator is given in Appendix A and the improvement in the field is illustrated in Figure 6(c).



(a) (b) (c)

Figure 6: *(a) The blue arrows are gradients of the upper right limb field function and the red arrows are gradients of the lower left limb field function reconstructing the arm of Figure 5(a) after an elbow flexion of $\pi/4$ (fields are composed with a union). On the bottom, $\alpha = \pi/4$, and on the top, gradients are collinear ($\alpha = 0$). (b) Illustration of the field distortions introduced by the gradient-based blending operator inside the shape, around the joint. (c) Our gradient-based blending operator avoiding this undesired depression.*



(a) (b)

Figure 7: *Plot of our functions for $\theta(\alpha)$, (a) to get skinning with some blending (around $\pi/4$) at an elbow or a knee, and (b) to mimic the inflation of tissues in contact situation.*

Bulge-in-contact. Lastly, other types of joints, such as finger joints, almost immediately produce contact surfaces between neighboring skin parts, surrounded by muscular bulges (Figure 1(d,e)). This is best modeled using a gradient-based bulge-in-contact operator as proposed in [Gourmel et al. 2013]. This operator allows the interpolation between a union and a contact surface surrounded by a bulge, giving an organic look to the resulting shape by localizing the bulge around contacts only. As for the gradient-based blending operator, choosing specific values for θ allows us to tune the bulging behavior when a joint bends: to avoid the unwanted bulge at the joint in the rest pose, we set $\theta(0) = \theta_u$ (union); we then slowly inflate the bulge up to its maximum when the bones are orthogonal ($\alpha = \pi/2$), and keep it maximal for larger angles, by setting $\theta(\alpha) = \theta_c$ (bulge in contact) , $\forall\alpha \in [\pi/2, \pi]$, as plotted in Figure 7(b).

5 Surface tracking

During animation, the implicit primitives f_i associated with each bone are rigidly transformed and combined as explained in the previous section, resulting in a time-dependent global field f. Directly deforming the mesh by tracking the 0.5-isosurface using the field function derivatives [Stam and Schmidt 2011] would result in a smooth distorted mesh after several transformations. Our approach is rather to use the whole region where f is non-zero as an embedding volume for the mesh, used to control its deformations from a standard geometric skinning pose. This enables us to have the mesh track the implicit deformations at low cost, while not losing any surface detail.

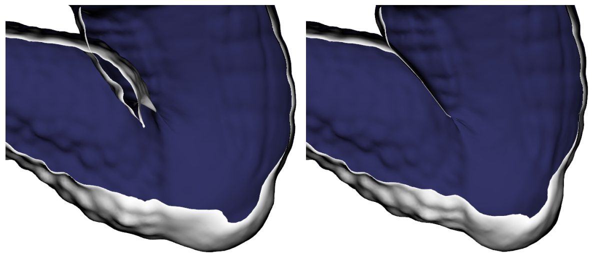

(a) (b)

Figure 8: *Section of the Armadillo's knee of Figure 9 showing the inside part of the mesh. The blue area is the interior side of the mesh and the clear gray its outside. The white lines are the mesh boundaries produced by the cut. As we can see in (a), LBS generates self-intersections in the fold that are converted in (b) into a contact region using our projection.*

Dual quaternions [Kavan et al. 2008] are used to compute an initial position of the mesh vertices at each animation step. The challenge then consists in designing a fast projection operator to correct vertex positions while accounting for both the detailed mesh shape at rest, and the current implicit deformation modeled by f. Our solution makes use of projection in gradient field directions, tangential relaxation, and local Laplacian smoothing, as detailed below.

Vertex projection. During deformation, we project each vertex $\mathbf{v}_i$ back onto its original iso-value iso_i (with $iso_i = f(\mathbf{v}_i)$ in the rest pose), enabling us to account for both the detailed mesh shape and for the current deformation. This is done using Newton iterations, which are both faster and more robust than the use of a constant step size:

$$\mathbf{v_i} \leftarrow \mathbf{v_i} + \sigma\left(f(\mathbf{v_i}) - iso_i\right)\frac{\nabla f(\mathbf{v_i})}{\|\nabla f(\mathbf{v_i})\|^2} \quad . \tag{4}$$

Using $\sigma = 0.35$ is an effective compromise between speed of convergence and accuracy.

In order to prevent any self-intersection of the final surface (Figure 8(a)), as well as to preserve the regularity of the mesh, the projection has to be stopped at the contact surface between different skin parts (Figure 8(b)). Contact surfaces correspond to gradient discontinuities of the scalar field. We detect the later using the following conservative heuristic: during projection, we keep track of the angle between the gradient at two consecutive iterations, and stop when it is greater than $55°$, thus considering we are on the contact surface (see Figures 2(h) and 8(b)).

Tangential relaxation. Since the initial vertex positions provided by geometric skinning might be far away from their final position, simply using the previous method could introduce high distortions of mesh faces, and at the extreme cause self-intersections. In order to both minimize them and improve the march of the vertices in the gradient field, we interleave projection steps with tangential relaxation steps, which move each vertex towards the weighted centroid of its neighbours. More precisely, given a vertex $\mathbf{v}_i$, let $\mathbf{q}_{i,j}$ be its one-ring neighbors projected onto its tangent plane. As a preprocess, we compute the barycentric coordinates $\Phi_{i,j}$ such that $\mathbf{v}_i = \sum_j \Phi_{i,j}\,\mathbf{q}_{i,j}$, using the mean value coordinate technique [Hormann and Floater 2006]. Each relaxation step moves the vertices tangentially to the local surface using:

$$\mathbf{v}_i \leftarrow (1-\mu)\mathbf{v}_i + \mu\sum_j \Phi_{i,j}\,\mathbf{q}_{i,j} \ , \tag{5}$$

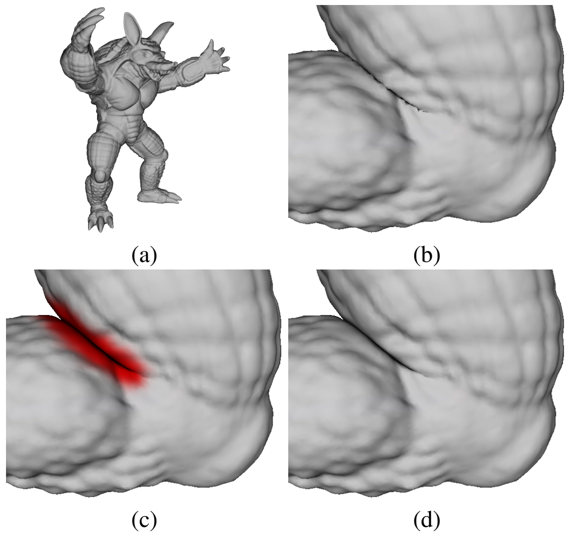

(a) (b) (c) (d)

Figure 9: *Animation of the Armadillo's knee (a). (b) The result of the projection without smoothing. (c) In red, the smoothed area where $\beta_i > 0$ and (d) the result after a few steps of Laplacian smoothing.*

where $\mu \in [0, 1]$ controls the amount of relaxation. While by construction, this has no effect on the rest pose, it improves the distribution of the vertices after deformation. A relaxation step is applied after each pair of vertex projection steps with an adjusted value of μ, such that the displacement of the vertices decreases with the distance to their target isosurface. This is done by setting:

$$\mu = \max\left(0, 1 - (|f(\mathbf{v}_i) - iso_i| - 1)^4\right) \ . \tag{6}$$

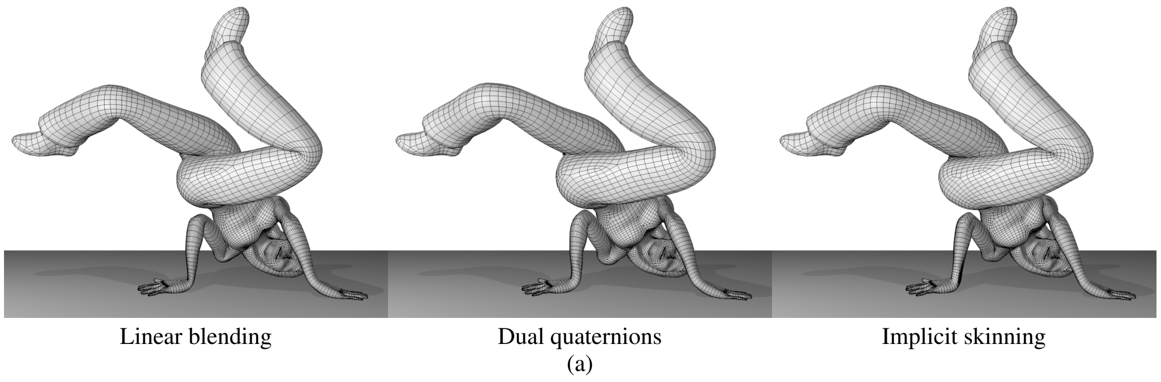
Laplacian smoothing. Even though we assume smooth inputs, a deformation performed by the union operator introduces sharp features at the boundaries of contact regions (Figure 9(b)). The latter should be smoothed out in order to mimic realistic skin deformations. We remove these high frequencies by locally applying Laplacian smoothing to the final mesh, at and around contact regions:

$$\mathbf{v}_i \leftarrow (1-\beta_i)\mathbf{v}_i + \beta_i\tilde{\mathbf{v}}_i \ , \tag{7}$$

where $\tilde{\mathbf{v}}_i$ is the centroid of the one-ring neighborhood of $\mathbf{v}_i$, and β_i controls the amount of smoothing. This is only done for mesh vertices marked as in contact regions, so that surface details are preserved elsewhere. To this end, we set β_i to 1 for vertices stopped at a gradient discontinuity, and to 0 for the others, and smooth the β_i values over the mesh by diffusion, prior to applying Laplacian smoothing (Figure 9(c)). As illustrated in Figure 9(d), this effectively smoothes out the mesh oscillations at sharp features, resulting in organic-looking shapes. Note that this smoothing step has to be performed after the projection step. Therefore it cannot be combined with the relaxation that is interleaved with projection during the march of the mesh vertices.

6 Implementation and results

Implementation. All our results were generated on a Intel Core i7 3770K at 3.5GHz, with 16GB of memory and a Geforce 680. Our CUDA implementation makes extensive use of GPU parallelism. In particular dual quaternions, projection, tangential relaxation and localized Laplacian smoothing are all parallelized over



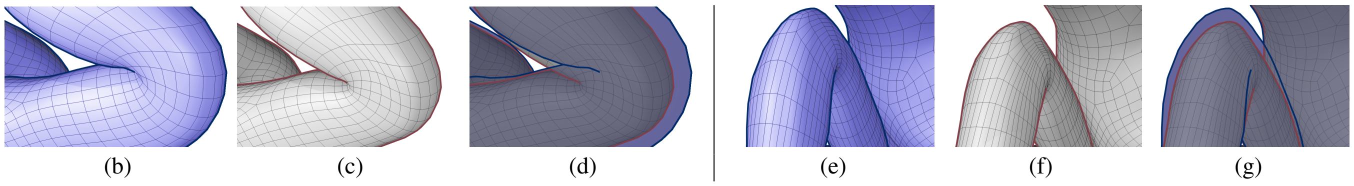


Figure 10: *Dana model in a break-dance pose. (a) From left to right, the model is deformed with LBS, dual quaternions and our technique. Note the very smooth deformation with visible loss of volume produced by the LBS (left). (b) Close-up on Dana's knee deformed with dual quaternions and (c) after correction with our method using the union operator. (d) The difference is shown by superposing the two surfaces in (b) and (c). The dual quaternion deformation is in flat-blue with a dark blue silhouette, and our improved solution is transparent gray with wire-frames and a brown silhouette. The overgrowth produced by dual quaternions is corrected and contact surfaces are generated at knee and crotch. (e, f, g) Same illustrations on Dana's elbow.*

	vertices	bones	memory	1 joint	animation	
					DQ	Our
Hand	31,750	21	10.5	35	95	15
Armadillo	172,974	23	11.5	36	87	3
Juna	32,056	55	22.5	86	340	15
Dana	4,164	67	32.5	270	> 800	95
Carl	6,866	67	32.5	300	> 800	68

Table 1: *For each model, from left to right: the number of vertices, the number of skeleton bones, the memory consumption for the storage of HRBFs in MB, the frame rates in fps when animating a single joint using our technique, and the frame rates in fps when moving simultaneously most bones in a real animation (poses are shown in Figure 11) with dual quaternions and our technique.*

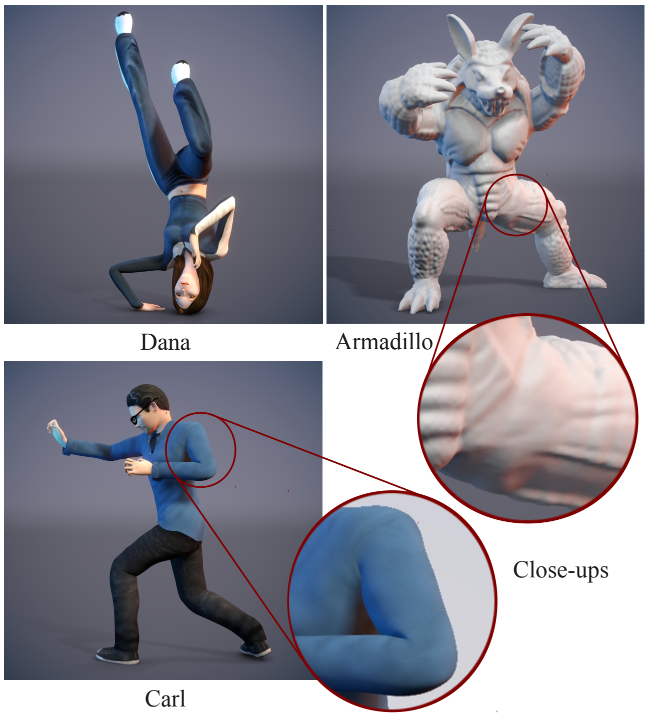


Figure 11: *Different models skinned with our technique. Number of bones and vertices, memory and frame rates are given in Table 1.*

the vertices. The individual field functions (Section 4.1) and composition operators (Section 4.2) being both compactly supported, are respectively sampled into 3D textures of resolution 32^3 (computed from the HRBF equation in 15 ms) and 128^3 with trilinear interpolation, leading to very fast evaluations of f and g_k on the GPU. Our models are composed of 20 to 70 bones, requiring 10Mb to 35Mb of memory for storing the field functions (Table 1).

The performances of our method for animating the different models used in the paper are summarized in Table 1. The frame rates only stand for the deformation time. As we can see, the performance of our technique mainly depends on the number of deformed mesh vertices that are to be projected on the implicit surface. However, the parallel implementation performed on recent GPUs allows us to animate simultaneously most bones of models composed of thousands of vertices at more than 60 fps and models composed of tens of thousands of vertices at more than 12 fps (examples of animation poses are shown in Figures 10 and 11). Even highly detailed models composed of more than $170,000$ vertices are still animated at several fps.

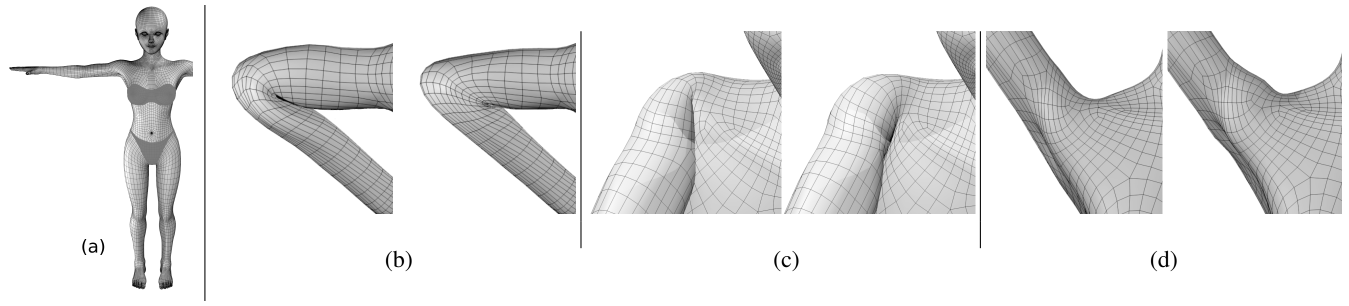


Figure 12: *Skinning of the Juna model shown in (a). Dual quaternion skinning (left) vs our skinning with gradient-based blending (right) on (b) the bended elbow, and (c,d) two poses of the shoulder. (b,c) While geometric skinning produces self intersections, our method generates the contact and keep it farther from the joint. (d) Shoulders details are smoothed by geometric skinning and well preserved by our method.*

Deformations. We illustrate the use of our skinning technique with different composition operators, and on several models and poses (Figure 11). The union operator is used for the fingers joints in Figure 1(c) and the Dana model in Figure 10. It allows us to correct the "elbow collapse" effects of the LBS as well as the smooth outgrowth produced by dual quaternions. It also nicely captures the aspect of a solid bone at joints with, inside the bent region, a contact between skin parts as illustrated in Figures 10. At bone joints, this makes our approach closer to real skin deformation than other real-time skinning techniques. The use of our new gradient-based blending operator, illustrated on Juna's elbow and shoulder in Figures 5(d), 12(b,c,d), and on the Armadillo's knee in Figure 9(d), allows, in addition, the creation of a smooth skin deformation inside the bend region for small rotation angles, before generating the contact for angles greater than $\pi/2$. This is especially useful where the generation of a hollow or a contact for small angles is unrealistic. More subtle and also more realistic, for angles larger than $\pi/2$, the contact area is kept farther from the joint (Figure 12(b)). Finally, the bulge-in-contact operator can be used to mimic the tissues/fat bulging when fingers bend. This is illustrated in Figure 1(d) on phalanx deformations during the hand animation. In all these animations, smooth deformations and contact are adequately generated while the mesh details are preserved, as illustrated on the Armadillo in Figure 11, its thigh and calf in Figure 9 and on Juna's shoulder in Figure 12(d).

Our technique also addresses the candy wrapper effect when bones twist. This is illustrated on a twisted cylinder in Figure 13 where we can see that the deformation is adequately corrected with our operators, even when the bones bend. We illustrate our result in Figure 13(c) using gradient-based blending as it is the most challenging operator to handle with our projection method.

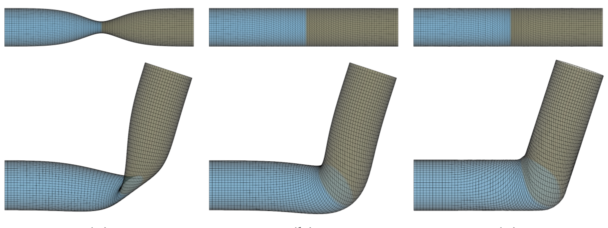


Figure 13: *Illustration of our method for modeling smooth skin surfaces at joints (new gradient-based blending operator), when the bones respectively twist (top row), or twist and bend (bottom row). (a) LBS, (b) dual quaternions, (c) our technique with gradient-based blending.*

Parameters and shape control. The parameter settings given in this paper allow the automatic production of results. The parameters used by our method are: the number of sub-mesh vertices used as HRBF centers for fitting implicit surfaces; the location of closure points (Section 4.1); the choices of the composition operators; the shape of the associated function $\theta(\alpha)$ when gradient-based composition operators are used (Section 4.2); and finally σ and μ that respectively set the step size of the vertices displacements in the gradient direction and the strength of the tangential relaxation during projection (Section 5).

If not considering the composition operator that is left as a free parameter able to perform a union, a blend or a bulge-in-contact, the deformation is controlled by preset parameters. In our technique, the resulting shape is not very sensitive to a slight variation of these parameter values and modifications are predictable. Thanks to the interactive feedback provided by our method, we could easily set these parameter values experimentally in interactive sessions, as illustrated in the accompanying video with the adjustment of the bulge size for a finger. Also, when reconstructing the mesh with implicit surfaces, we are seeking for a smooth approximation of mesh parts, independently from the part size and geometry. Fifty samples were sufficient for all our models, which are of different resolution and include different levels of detail. Automatically modulating this number according to some given approximation error may however be a useful extension.

In our tests, complex joints such as shoulders may require the user to add or remove a few sample points used as HRBF centers at the vicinity of the joint. Here, the number of points to be added or removed mainly depends on the appropriateness of the input partitioning solution (Figure 2(c)). In our framework, this action is again performed with real-time feedback.

By default, the composition operator is set to the union for all the model bone joints, as was done for the Dana model (Figure 10). The result is already a more plausible deformation than the one provided by LBS, dual-quaternions or the more recent elasticity-inspired deformer [Kavan and Sorkine 2012] thanks to the rigid bone aspect generated outside the fold and to the contact produced inside. The deformation can then benefit from the gradient-based blend and bulge-in-contact implicit composition operators to be improved at, for instance, elbows, shoulders and knees for the former and fingers for the later. These two pre-set gradient-based operators are provided to the user: he only has to select between a sharp (union), a smooth (blending) or a bulging (bulges-in-contact) deformation at each joint.

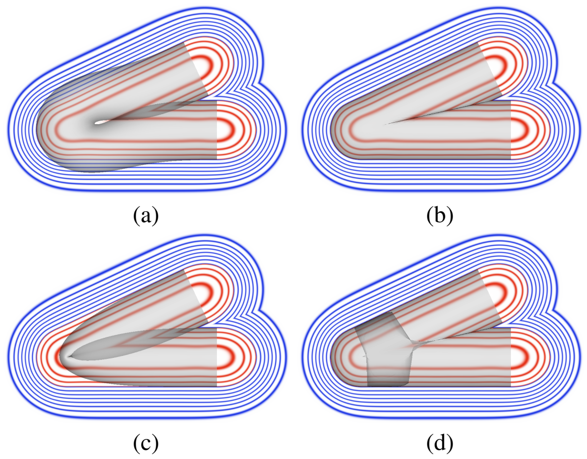


Figure 14: *The field of a bent cylinder, reconstructed with a union operator. In red, the part inside the* 0.5*-isosurface, and in blue the outside. The mesh is shown in transparency. (a) The mesh deformed using dual quaternions is (b) adequately corrected by our method. However, (c) the mesh deformed with LBS, has sets of vertices crossing each other, and located beyond the bones. (d) It leads to many projections in the wrong direction where our method fails.*

7 Discussion and limitations

Influence of the initial geometric skinning solution. At each animation step, the mesh is first deformed using a standard geometric skinning, before being corrected through the projection of its vertices to the adequate isosurfaces. Therefore, the quality of our results is affected by the choice of the initial skinning solution. In particular, our correction gives better results if the diffusion of the bone influences is large enough to produce a smooth initial deformation. The geometric skinning used should also avoid the apparition of deep self-intersections which would send some mesh vertices beyond the bones. In that case, these vertices would be projected on the opposite surface, as illustrated with the union operator in Figure 14. In the first row of the figure, the smoother dual quaternions deformation enables an adequate correction, while the input LBS deformations in the second row generate deep self-intersections, causing a failure of our method. Also, in extreme poses, even though contact is handled, the deformation may be inaccurately corrected: some mesh vertices may cross a contact and be stopped, while they were supposed to be projected on the visible skin part, outside the fold.

Smooth skin parts. The neck of a giraffe, the elephant's trunk or the arms of an octopus smoothly deform when they are animated. This effect is not handled by our method which focuses on more rigid bone-like joints. In this case, the smoothing property of LBS near a joint, usually seen as an artifact, may produce a result closer to the expected shape.

Time independence. At each animation step, the initial mesh vertex positions are computed from the rest pose after a geometric skinning deformation. Therefore, the final vertex positions at a given animation step do not depend on their positions at the previous step. The consequences are two-fold. On the one hand, the solution might be subject to flickering. In practice, as soon as the mesh resolution is dense enough to capture the local shape at joints, the continuity in time of the initial solution computed with geometric skinning coupled with the smoothness of the field functions prevent this effect. On the other hand, this eases the integration of our approach in standard animation pipelines where frames are computed in parallel.

Mesh resolution. As mentioned in the previous paragraph, the mesh resolution should be dense enough to capture features such as contact, blend or bulge. When the mesh has too coarse a resolution, very few vertices lie on the different features introduced by our skinning technique at joints. Thus, they are not appropriately represented and the relaxation followed by the local smoothing (at the end of the projection) are likely to degrade the vertex distribution, especially if the mesh is very irregular. This can make the mesh deformation unrealistic and discontinuous in time.

8 Conclusion and future work

We presented a novel approach for skinning virtual characters, which operates in real-time. Based on the embedding of the skin mesh into a deformable implicit volume, made of parts rigidly attached to each bone, it does not suffer from the loss of volume inherent to standard geometric skinning methods. Choosing the way the implicit parts are combined enables us to tune transitions between smooth and sharper skin shapes when a joint bends, to generate contact surfaces between neighboring skin parts, and to drive organic bulging effects. As a result, our approach achieves visually plausible deformations of the different joints in a character's body, even for extreme bending angles. Since no optimization or collision processing steps are required, our approach is robust and efficient. Lastly, computations being independent from one frame to the next, the method perfectly fits into standard animation pipelines.

With our technique, the union operator is used to generate contacts between non-neighboring reconstructed skin parts, as done in the teaser image. In the future, we would like to investigate more advanced field combination schemes to model a wider set of deformations due to collisions between these non-neighboring parts. This may require extending the binary gradient-based composition operators to n-ary compositions. The field values along contact surfaces could also be used, either to output contact forces to be applied to skeleton bones, or simply to detect that an angular limit is reached at the joint.

We would also like to develop more general combination operators, enabling us to capture both smooth blends for small angle values, and contact surfaces surrounded by bulges for larger bends. The fact that gradient-based operators can be edited in real-time by playing with the function θ should ease this tuning. We could also imagine an expert mode where users could edit and control skin deformation at joints through direct interaction, without requiring any knowledge on the theory of our method: after selecting the deformation (sharp, smooth or bulging) for a joint, they could adjust its parameters by directly articulating the input mesh into a specific bending angle, tuning shape composition there, and looking at the resulting animation in real-time.

Finally, inspired by the work of Rohmer et al. [2009] and Barthe et al. [2001], implicit skinning could be directly tuned from profile curves sketched by the user, and depicting the desired skin shape. These profiles curves, possibly including wrinkles such as those that appear when a wrist articulates, would drive the generation of specific gradient-based compositions.

9 Acknowledgments

This work has been partially funded by the IM&M project (ANR-11-JS02-007) and the advanced grant EXPRESSIVE from the European Research council. Partial funding also comes from the Natural Sciences and Engineering Research Council of Canada, the GRAND NCE, Canada and Intel Corps. Finally, this work received partial support from the Royal Society Wolfson Research Merit Award.

References

ANGELIDIS, A., AND SINGH, K. 2007. Kinodynamic skinning using volume-preserving deformations. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 129–140.

BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3D characters. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH '07.

BARTHE, L., GAILDRAT, V., AND CAUBET, R. 2001. Extrusion of 1D implicit profiles: Theory and first application. *International Journal of Shape Modeling 7*, 179–199.

BARTHE, L., WYVILL, B., AND DE GROOT, E. 2004. Controllable binary CSG operators for "soft objects". *International Journal of Shape Modeling 10*, 2, 135–154.

BERNHARDT, A., BARTHE, L., CANI, M.-P., AND WYVILL, B. 2010. Implicit blending revisited. *Comput. Graph. Forum 29*, 2 (mai), 367–375.

BHARAJ, G., THORMÄHLEN, T., SEIDEL, H.-P., AND THEOBALT, C. 2011. Automatically rigging multi-component characters. *Comp. Graph. Forum (Proc. Eurographics 2012) 30*, 2.

BLINN, J. F. 1982. A generalization of algebraic surface drawing. *ACM Trans. Graph. 1*, 3, 235–256.

BLOOMENTHAL, J. 2002. Medial-based vertex deformation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, SCA'02, 147–151.

CANI, M.-P. 1993. An implicit formulation for precise contact modeling between flexible solids. In *20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1993*, 313–320. Published as Marie-Paule Gascuel.

FORSTMANN, S., OHYA, J., KROHN-GRIMBERGHE, A., AND MCDOUGALL, R. 2007. Deformation styles for spline-based skeletal animation. *Symposium on Computer Animation (SCA)*, 141–150.

FUNCK, W. V., THEISEL, H., AND SEIDEL, H. P. 2008. Volume-preserving mesh skinning. *Workshop on Vision, Modeling and Visualization (VMV)*.

GOURMEL, O., BARTHE, L., CANI, M.-P., WYVILL, B., BERNHARDT, A., PAULIN, M., AND GRASBERGER, H. 2013. A gradient-based implicit blend. *ACM Transactions on Graphics 32*, 2.

HORMANN, K., AND FLOATER, M. S. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Transaction on Graphics (TOG) 25*, 4.

JACOBSON, A., AND SORKINE, O. 2011. Stretchable and twistable bones for skeletal shape deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA) 30*, 6, 165:1–165:8.

JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 30*, 4, 78:1–78:8.

KAVAN, L., AND SORKINE, O. 2012. Elasticity-inspired deformers for character articulation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA) 31*, 6, to appear.

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, New York, NY, USA, 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* (November), 105:1–105:23.

KAVAN, L., COLLINS, S., AND O'SULLIVAN, C. 2009. Automatic linearization of nonlinear skinning. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, New York, NY, USA, I3D '09, 49–56.

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*, ACM, New York, NY, USA, SCA '02, 153–159.

LECLERCQ, A., AKKOUCHE, S., AND GALIN, E. 2001. Mixing triangle meshes and implicit surfaces in character animation. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 37–47.

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*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 165–172.

MACÊDO, I., GOIS, J. P., AND VELHO, L. 2011. Hermite radial basis functions implicits. *Computer Graphics Forum 30*, 1, 27–42.

MAGNENAT-THALMANN, N., LAPERRIÈRE, R., AND THALMANN, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*, Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 26–33.

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

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. In *ACM SIGGRAPH 2011 papers*, ACM, New York, NY, USA, SIGGRAPH '11, 37:1–37:12.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH '03, 562–568.

NG-THOW-HING, V. 2001. *Anatomically-based models for physical and geometric reconstruction of humans and other animals*. PhD thesis, Toronto, Ont., Canada, Canada. AAINQ58941.

RICCI, A. 1973. Constructive Geometry for Computer Graphics. *computer journal 16*, 2 (May), 157–160.

ROHMER, D., HAHMANN, S., AND CANI, M.-P. 2009. Exact volume preserving skinning with shape control. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '09, 83–92.

SHEN, J., AND THALMANN, D. 1995. Interactive shape design using metaballs and splines. In *Proceedings of Implicit Surfaces*, 187–196.

SHEN, C., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2004. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph. 23*, 3, 896–904.

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2007. Mesh puppetry: Cascading optimization of mesh deformation with inverse kinematics. *ACM Transaction on Graphics (TOG), Proceedings of ACM SIGGRAPH 26*, 3.

SINGH, K., AND PARENT, R. 1995. Implicit surface based deformations of polyhedral objects. In *first EG workshop on Implicit Surfaces*. Invited Paper.

SLOAN, P.-P. J., ROSE, C. F., AND COHEN, M. F. 2001. Shape by example. *ACM Symposium on Interactive 3D Graphics (i3D)*, 135–143.

STAM, J., AND SCHMIDT, R. 2011. On the velocity of an implicit surface. *ACM Trans. Graph. 30*, 3 (May), 21:1–21:7.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *ACM/Eurographics Symposium on Computer Animation (SCA)*, K. Anjyo and P. Faloutsos, Eds., 181–190.

VAN OVERVELD, C. W. A. M., AND BROEK, B. C. V. D. 1999. Using the implicit surface paradigm for smooth animation of triangle meshes. In *Proceedings of the International Conference on Computer Graphics*, IEEE Computer Society, Washington, DC, USA, CGI '99, 214–221.

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*, ACM, New York, NY, USA, SCA '02, 129–138.

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

WEBER, O., SORKINE, O., LIPMAN, Y., AND GOTSMAN, C. 2007. Context-aware skeletal shape deformation. *Computer Graphics Forum (Proceedings of Eurographics) 26*, 3.

WENDLAND, H. 2005. *Scattered Data Approximation*. Cambridge University Press, ch. 16.2 - Hermite–Birkhoff interpolation.

WHITE, K. B., CLINE, D., AND EGBERT, P. K. 2007. Poisson disk point sets by hierarchical dart throwing. In *Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing*, IEEE Computer Society, Washington, DC, USA, RT '07, 129–132.

A Gradient-based blending operator

In this Appendix, we give the equation of our gradient-based blending operator. It has been specifically designed for improving the march of the mesh vertices in the field during projection. Starting from the original gradient-based blending operator presented in [Gourmel et al. 2013], we modify the inside part such that the blend decreases between fields values 0.5 and 0.7. Deeper inside, when $f_i > 0.7$ or $f_j > 0.7$, the operator is a max (union) (Figure 6(c)). This leads to the following definition for our new blending operator g_b:

$$g_b(f_i, f_j) = \begin{cases} \max(f_i, f_j) & \text{if } f_i > 0.7 \text{ or } f_j > 0.7 \\ \max(f_i, f_j) & \text{if } f_i < k_\theta(f_j) \text{ or } f_j < k_\theta(f_i) \\ \\ \{C : \bar{g}_C(f_i, f_j) = 1\} & \text{otherwise,} \end{cases}$$

in which $\bar{g}_C$ is defined as :

$$\bar{g}_C = \frac{\sqrt{(f_i - k_\theta(C))^2 + (f_j - k_\theta(C))^2}}{(C - k_\theta(C))}\ , \qquad (8)$$

and the function $k_\theta(C)$ is defined as:

$$k_\theta(C) = \begin{cases} \tan(\theta)\, C & \text{if } f_i \leq 0.5 \text{ and } f_j \leq 0.5 \\ \frac{1}{2}\,(7 - 5\tan(\theta))\, C + \frac{7}{4}\,(\tan(\theta) - 1) & \text{otherwise.} \end{cases}$$

In practice, equation 8 does not need to be explicitly solved. We rather pre-compute the values of g_b in a 3D texture using the method in [Gourmel et al. 2013]. Note that this procedure requires the definition of a function $\bar{s} : \mathbb{R} \to \mathbb{R}$. In our case, we simply use $\bar{s} = 1$.