Robust Iso-Surface Tracking for Interactive Character Skinning

Rodolphe Vaillant[1,2], Gaël Guennebaud[3], Loïc Barthe[1], Brian Wyvill[2], Marie-Paule Cani[4]
[1]IRIT - Université de Toulouse, [2] University of Victoria, [3]Inria - Univ. Bordeaux - IOGS - CNRS,
[4] LJK univ. Grenoble-Alpes - Inria

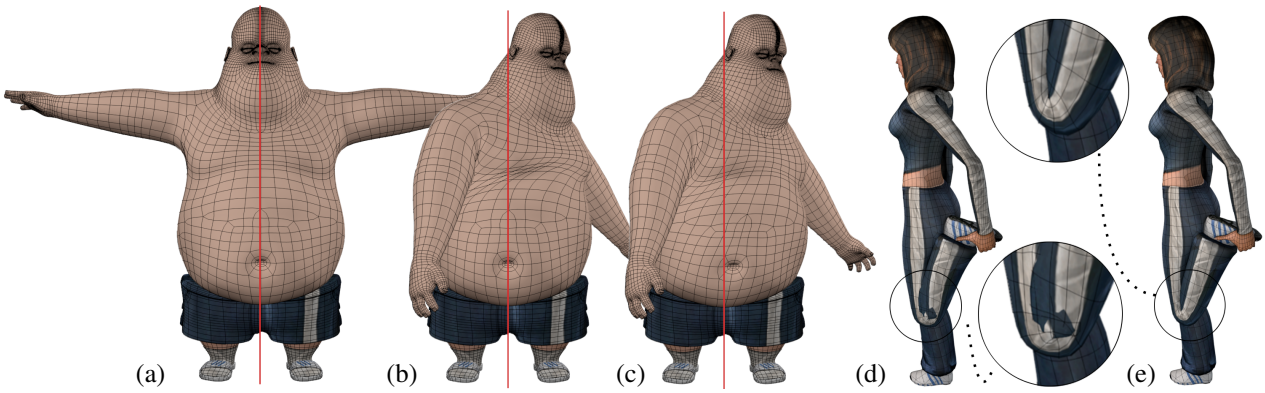


Figure 1: *(a) The Jeff model in rest pose. Its shoulders are rotated and skinned with (b) implicit skinning (by Vaillant et al. [2013]) which locally deforms the mesh according to some preset weights and (c) our new technique which automatically produces a plausible skin elasticity (notice how the belly button stretches). On the right, the Dana's knee is bent with an extreme rotation angle. (d) Implicit Skinning fails to handle deep self-intersections while (e) our technique allows large bending angles and the self-intersection at the knee is handled correctly.*

Abstract

We present a novel approach to interactive character skinning, which is robust to extreme character movements, handles skin contacts and produces the effect of skin elasticity (sliding). Our approach builds on the idea of implicit skinning in which the character is approximated by a 3D scalar field and mesh-vertices are appropriately re-projected. Instead of being bound by an initial skinning solution used to initialize the shape at each time step, we use the skin mesh to directly track iso-surfaces of the field over time. Technical problems are two-fold: firstly, all contact surfaces generated between skin parts should be captured as iso-surfaces of the implicit field; secondly, the tracking method should capture elastic skin effects when the joints bend, and as the character returns to its rest shape, so the skin must follow. Our solutions include: new composition operators enabling blending effects and local self-contact between implicit surfaces, as well as a tangential relaxation scheme derived from the as-rigid-as possible energy to solve the tracking problem.

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

Keywords: Skinning with contact, Iso-surface tracking

Links: DL PDF

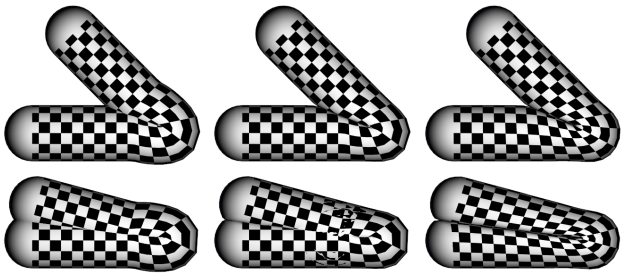
1 Introduction

Many computer graphics applications include the animation of synthetic 3D characters. One challenging aspect is the production of realistic skin deformations, especially at joints. To achieve realism, self-collisions must be avoided and contact surfaces generated instead. In addition, being able to add muscle inflation and skin stretching during body animation is highly desirable. Two main approaches have been used for skinning characters so far: geometric skinning and physically-based simulation. Geometric skinning techniques such as smooth skinning or dual quaternions enable real-time animation, but are not able to perform contact deformations nor reproduce plausible skin elasticity (i.e., its ability to stretch and slide over the character's volume). In contrast, physically-based simulation can be used to achieve these effects, however, long computation times prevent interactive feedback. In addition user control of the muscle setup (when necessary) can be very tedious.

Recently, skinning with contact deformations has been achieved in real time using the implicit skinning method [Vaillant et al. 2013]: the character's volume is approximated by a set of local, 3D scalar fields (Figure 3 (b)-top). The scalar fields approximate the limbs, and at each animation frame, they are rigidly transformed by the skeleton and combined with an appropriate composition operator, producing a single scalar field (Figure 3 (b)-bottom). To obtain the final deformation, mesh vertices are deformed using standard geometric skinning and then iteratively projected onto their original iso-surface within the scalar field.

While the results of this technique are encouraging, its use of standard geometric skinning as an initial solution requires specifying skinning weights, which are tedious to define. Moreover, as observed by the authors, the success of the correction is conditioned by the quality of the initial skinning solution. When the initial vertices are too far from their ideal final position, which always happens for large bending angles, the following problems occur:

- **Foldovers and triangle inversions** are produced when some triangles deformed by geometric skinning are nearly par-



(a) - Dual Quat. (b) - Imp. Skinning (c) -Our

Figure 2: *Illustration of different skinning solutions on a cylinder deformed with a single joint. (a) Dual quaternions produce bulge and self-intersection artifacts. (b-top) The implicit skinning corrects these artifacts, (b-bottom) but the projection stage fails for too large bending angles. (c) Our skinning technique avoids self-intersections, produces a contact surface and minimizes mesh distortions, even for very large bending angles.*

allel to the local gradient of the scalar field (Figures 1(d) and 10(c)).

- **Bad projections** occur when self-intersecting vertices cross the scalar field's medial axis. This projects vertices on the wrong side of the medial axis. (Figure 2(b)-bottom).
- **Mesh distortions** (e.g. excessive stretch) introduced by the geometric skinning step (Figure 2(a)) are not corrected during the projection on the implicit skin (2(b)).
- **The projection is unstable** due to the fact that contacts are detected as gradient discontinuities in the 3D scalar field, but not modeled explicitly as iso-surfaces of the field.

In practice, these flaws restrict the range of possible character poses, making the method less useful in a production environment.

Our goal is to design a production friendly skinning technique, which takes advantage of the desirable features of the implicit skinning framework, but avoids the pitfalls. The solution presented in this paper captures skin contact and elasticity, as well as muscle inflation and skin sliding effects. It also provides more automation and robustness than previous methods while still being applicable at interactive frame-rates.

More precisely, our contributions include:

- **A new implicit skinning pipeline**, free from the need of any initial skinning solution and therefore avoiding both tedious specification of skinning weights and dependence on the quality of a geometric solution (Section 3).
- **New composition operators** enabling us to approximate the shape of the character's skin during animation with an *implicit skin* that includes contact surfaces with a controllable depth, as depicted in Figure 3(e) (Section 4).
- **A robust tracking method** based on a linear relaxation energy (inspired from the as-rigid-as-possible energy) to update the skin mesh with a plausible tangential distribution of vertices, as shown in Figures 1(c) and 2(c) (Section 5).
- **Easy extensions to advanced effects** such as muscle inflation (Figure 10) and skin sliding, thanks to the clear decoupling between volumetric deformations handled by the implicit skin and stretching effects handled by the skin mesh (Section 6).

2 Related Work

2.1 Skinning methods

The most popular methods to animate the skin of a character are geometrical [Magnenat-Thalmann et al. 2004; Kavan and Žára 2005; Kavan et al. 2008; Kavan and Sorkine 2012; Kim and Han 2014]. In particular, linear blend skinning and dual quaternion skinning are very fast to compute and relatively simple to implement. At run time, they directly weight the skeleton transformations before applying them onto the corresponding mesh parts. Although associating skinning weights to mesh vertices can be done automatically [Dionne and Lasa 2014], the artist often has to manually tune them to improve the results. Moreover, these two methods tend to either increase or decrease the volume around joints, and interpenetrations are generated when different skin parts come in contact.

Shape interpolation schemes [Lewis et al. 2000] and example-based methods [Mohr and Gleicher 2003; Wang et al. 2007; Weber et al. 2007] can also be used to skin a character in real-time. Unfortunately, capturing contact deformations is not very practical since it requires the inclusion of all possible contact situations in the example database. Moreover, skin shapes cannot be easily interpolated or extrapolated in the neighborhood of a contact, the deformation being highly non-linear.

A solution to get more realistic results is to use physically-based simulation, for instance based on anatomical knowledge [Ng-Thow-Hing 2001; Teran et al. 2005; McAdams et al. 2011]. The necessary anatomical data can be semi-automatically transferred between different characters [Ali-Hamadi et al. 2013], yet, the intensive computation required usually prevents interactive feedback. In addition, user control of the muscle setup (when necessary) can be very tedious. Methods such as elasticity skinning [McAdams et al. 2011] enables the computation of skin squash and stretch directly on the mesh by solving underlying physical equations, however, several seconds per animation frame are required for visualizing the results. Alternatively, one can use a set of training poses [Teng et al. 2014] to accelerate the simulation.

Recently, skinning with contact modeling has been achieved in real-time using the implicit skinning method [Vaillant et al. 2013]. In addition to generating visually compelling shapes at joints, this method provides a variety of transitional effects, from smooth skin junctions at the elbow for low bending angles to local bulges when fingers articulate. These effects capture at low cost some of the dynamic behavior provided by physically-based approaches. Similarly to geometric skinning approaches, implicit skinning requires the specification of skinning weights. Indeed dual quaternion skinning is used at each frame as an initial solution. Another flaw is that the method does not provide any implicit representation (i.e. the 3D scalar fields) of contact surfaces between skin parts: skin vertices trying to reach their associated iso-surface are simply stopped when they cross gradient discontinuities of the scalar field, causing instabilities. Lastly, as already discussed, the correction fails for deep self-penetrations between skin parts.

In this work, we revisit implicit skinning to avoid the use of an initial geometric skinning solution. Our method relies on both a new composition operator enabling us to model contact surfaces within the implicit skin, and a new, robust implicit surface tracking method applied to mesh points. We therefore review the previous work in these two domains below.

2.2 Composition operators for contact modeling

Starting from 3D scalar fields reconstructing the different parts of the input skin mesh, implicit skinning can make use of a variety of field composition operators to create the implicit skin, i.e. the 0.5

iso-surface around which the mesh is repositioned: If a fold needs to be generated as soon as the joint bends, Ricci's [1973] $\max$ operator is applied. More advanced effects such as a bulge-in-contact, or a smooth blend relies on specific *gradient-based* composition operators [Gourmel et al. 2013]. Gradient-based operators output a scalar field value which depends on both the input fields and their local gradients. Although they enable capturing the desired outside shape, these different operators have a shortcoming: they do not model the contact surface between the two shape parts. While this does not create any visual artifact, contact surfaces being usually hidden, it prevents handling the projection of mesh vertices in contact regions in a robust manner, resulting in some vertices being pushed towards the wrong side of the shape, causing the instabilities we mentioned.

A contact operator is a composition operator able to define a contact surface (within the 0.5 iso-surface in our case) where the combined implicit primitives collide. Such operators were developed for long [Cani 1993] and were reformulated recently for more flexibility [de Groot et al. 2014]. With these operators, the the implicit primitives to be combined remain completely disjoint: they are separated by the 0.5 iso-surface modeling the contact surface in the collision region. Unfortunately, this solution is not feasible for blending two implicit primitives representing the same limb near a joint. We need to design composition operators able to merge the primitives into a single volume in the rest pose, while locally producing a contact surface whose extent should increase as the joint's bend increases (see Figure 3(e)). In this work, we have developed specific contact composition operators, enabling this advanced combination of implicit skin parts.

2.3 Implicit surface tracking

Tracking an iso-surface of a dynamic 3D scalar field with a mesh remains a challenging problem in the general case. Indeed, the lack of natural tangential parameterization of the field level-sets makes it difficult to maintain an even distribution of mesh vertices.

In the case of implicit skinning, each mesh vertex needs to track a slightly different iso-surface over the animated implicit skin, in order to maintain geometric details. Vaillant et al. [2013] solved the tracking problem by starting at each frame of a geometric skinning deformation of the mesh, and then interleaved Newton iterations in the field gradient direction with tangential mesh relaxation steps based on mean-value coordinates [Hormann and Floater 2006] to project each mesh vertex on its iso-surface of interest. While effective in most cases, this method is subject to fold-over and triangle inversions, projection failures and inappropriate mesh distortions, as already explained in Section 1.

Another approach is to perform tracking iteratively: then, instead of using some additional initial solution, the mesh restarts at each frame from its position at the previous step. Most mesh-based iterative tracking techniques perform an orthogonal projection of mesh vertices onto the iso-surface, combined with some tangential relaxation controlling the vertex distribution. The first solution was to consider mesh vertices as particles and apply a global optimization scheme in order to maintain their distribution [Witkin and Heckbert 1994]. Other relaxation schemes make use of mass-spring systems [Rodrian and Moock 1996] or of other physically inspired energies [Bouthors and Nesme 2007]. These solutions cannot easily be adapted to our problem, because the relaxation of a particle system would deform the initial skin mesh, which we would like to keep unchanged since it stores geometric details. Also, mass-spring systems are known to be subject to fold-over. Lastly, all these methods would require dynamic remeshing through edge split and collapse to be kept robust. This is to be avoided in our case, where we would like the skin mesh to come back close to its initial configuration when the character comes back to the rest pose.

Active contours [Xu and Prince 1998; Sonka et al. 2007] are also well known iso-surface tracking techniques. They focus on the accurate approximation of a given iso-surface rather than trying to deform an existing mesh while guaranteeing the time coherence of the deformation. Tracking can also be performed by assuming a translational transformation to extract a tangential component of the unknown deformation velocity field [Stam and Schmidt 2011] using differential information. This approach does not prevent the use of a relaxation scheme, and it appears less attractive for skinning, which mostly involves rotational transformations.

Making iterative tracking possible in an implicit skinning framework requires the introduction of a technique able to deal with contact surfaces and minimize the distortions in the vertex distribution. In particular, mesh distortions should be avoided if the character comes-back to rest pose after several transformations. This motivates the introduction of a new, dedicated tracking scheme.

3 Overview

Our skinning framework is decomposed in two distinct stages.

Presets: Starting with a mesh partitioned according to the bones of its animation skeleton (Figure 3 (a)), each mesh part i is approximated by the 0.5 iso-surface of a smooth 3D scalar field $f_i : \mathbb{R}^3 \rightarrow \mathbb{R}$ (Figure 3 (b)-top). Scalar fields f_i have compact support and vary in $[0, 1]$ with $f_i(\mathbf{p}) < 0.5$ for points $\mathbf{p}$ outside the limb and $f_i(\mathbf{p}) > 0.5$ for points $\mathbf{p}$ inside. They are generated by fitting Hermite-Radial-Basis-Functions to a subset of the mesh vertices. A single scalar field $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, whose 0.5 iso-surface (denoted as *implicit skin*) approximates the shape of the skin, is then computed by combining the fields f_i with specific composition operators (Figure 3 (b)-bottom). In order to preserve the details of the original mesh, each vertex $\mathbf{v}$ stores its own iso-value $f(\mathbf{v})$ within the resulting scalar field at rest-pose, to be projected back on this iso-value during animation.

While most of these presets follow the lines of Vaillant et al. [2013], they differ from the nature of the composition operator used for f: we define a new family of gradient-based composition operators, able to capture contact surfaces between different parts of the implicit skin and to generate a smooth field around it (Section 4). Note that our new composition is flexible, and therefore enables us to progressively increase the extent of the contact surface while a joint bends.

During animation: The scalar fields f_i are transformed according to their associated bone, resulting in a transformed version of the composed field f that embeds the implicit skin (Figures 3(c,d,e)-bottom). In these Figures, we show planar sections of the 3D scalar fields. Blue lines are sections of outer iso-surfaces, grey lines are the section of the implicit skin and red lines are sections of inner iso-surfaces. This color convention is used throughout the paper.

In contrast with original implicit skinning, our tracking technique uses the mesh position at the previous time step, to deform the skin mesh according to the composed field f: Each mesh vertex starts from its final position at the previous animation step (Figure 3(c)) and is projected on its saved iso-value $f(\mathbf{v})$ in the deformed field f while maintaining plausible elastic tangential mesh deformations and generating contact (Figures 3(d,e)). This is robustly done thanks to a new tracking method that successfully combines rigid transformations, standard Newton iterations following the gradients of the field f, and dedicated tangential relaxations (Section 5). Finally, a pass of Laplacian smoothing around con-

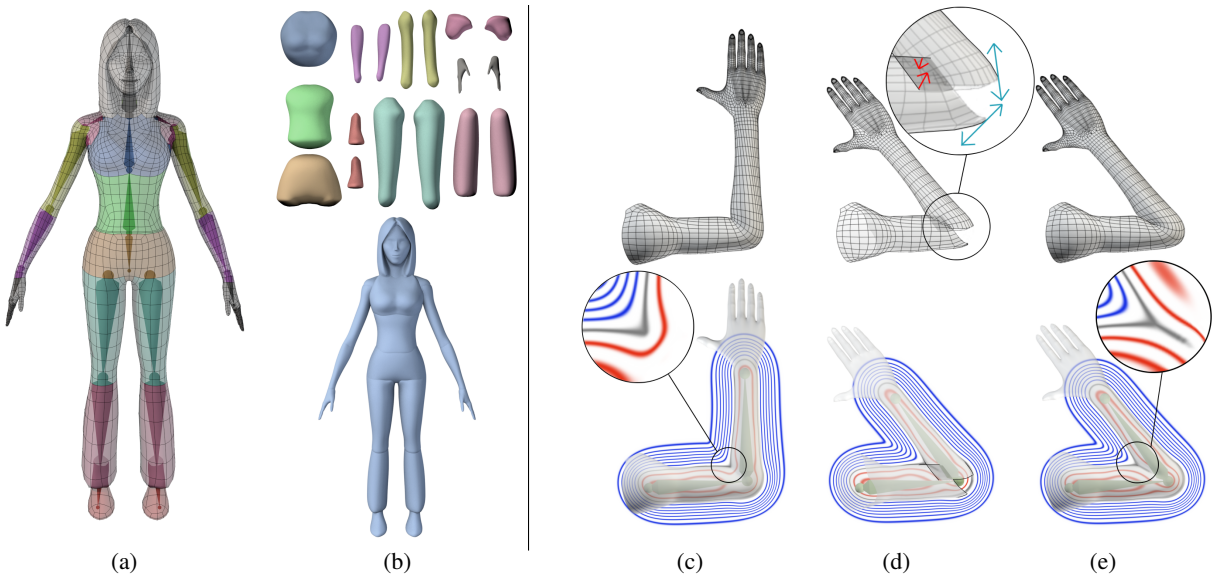


Figure 3: *Presets: (a) Initial mesh partitioned according to its skeleton. (b) (top) Each partition is approximated by a scalar field f_i and (bottom) composed with our new operators to produce the implicit skin. (c) (d) (e) We illustrate one step of our animation algorithm (top row the mesh deformation, bottom row the isolines of the implicit skin f). During animation, (c) the deformed mesh parts at an initial step n (d) are rigidly transformed at the step $n+1$. Then, vertices are projected onto their own iso-value $f(\mathbf{v})$ (red arrows) and tangentially relaxed (blue arrows). (e) The final result with the contact surface illustrated on the mesh (top) and in grey in the implicit skin (bottom).*

tact regions filters the sharp creases at the boundaries of contact surfaces.

The two next sections respectively detail our solutions to the implicit composition and to the mesh tracking problems.

4 Implicit skin with contact modeling

We need to define a single 3D scalar field f whose 0.5 iso-surface, the implicit skin, produces self-contact regions upon collision while merging implicit primitives around the joints in a smooth manner. The generation of a smooth C^1 continuous scalar field is required for robust iso-surface tracking (Section 5) as the gradient directions around the implicit skin need to point towards the closest (or nearly closest) point on the implicit skin, to limit mesh distortion at the tracking step. We present new operators for composing f in Section 4.1, and we explain how they are efficiently computed while retaining smoothness in Section 4.2.

4.1 Composition operators with contact

We solve the problem of enabling skin volumes to blend while sharing local contact surfaces using two new binary operators, organized in a composition tree [Wyvill et al. 1999]. The leaves of the tree are the field functions f_i each transformed in the same way as its corresponding bone and the composition operators are the interior nodes. A binary composition operator is a function $g : \mathbb{R}^2 \to \mathbb{R}$ combining two scalar fields f_1 and f_2 in a new 3D scalar field $f_k(\mathbf{p}) = g(f_1(\mathbf{p}), f_2(\mathbf{p}))$. Thus, each node of the tree defines a 3D scalar field and the root is the scalar field f including the implicit skin. Also, in our application we use a special type of composition operators called gradient-based operators.

As mentioned by Vaillant et al. [2013], gradient-based operators $g_d : \mathbb{R}^2 \to \mathbb{R}$ are parameterized by a function $d : \mathbb{R} \to \mathbb{R}$ whose argument is the angle α between ∇f_1 and ∇f_2. The parameter d controls the interpolation between two types of compositions, allows the discrimination of regions that blend according to the gradient angle and the application of different types of compositions. In the case of skinning, it is used to localize blending or bulge-in-contact effects in the fold only.

In the case of skinning, the effect of these operators is simpler to understand by noticing that the angle α between gradients in the folding regions is often close to the joint rotation angle. In the rest pose, no rotation is applied and thus, no specific skinning effect is to be performed. In that case, we need the 0.5 iso-surface of the scalar fields f_i to be combined with a union while the rest of the iso-surfaces are smoothly blended around the implicit skin. This type of operator is called a clean-union operator [Pasko et al. 1995; Bernhardt et al. 2010]. When a joint such as the elbow bends between $\alpha = 0$ and $\alpha = \pi/2$, the skin does not crease and no contact iso-surface need be generated. The use of a clean-union operator produces a smooth field around the implicit skin. For larger bending angles, $\alpha > \pi/2$ at an elbow, the contact iso-surface has to be created using our contact operator. In this case the larger the bending angle, the deeper the contact surface goes towards the joint. Finally, when the gradients point in opposite directions ($\alpha = \pi$), it means two disjoint skin parts have come into contact, for instance an arm against the torso, and a full contact operator should be applied.

Our first operator g_{d_c} is parameterized by the controller d_c for contact handling. It smoothly interpolates between a clean-union (no contact) when $d_c(\alpha) = 0$ and a full contact (contact of maximal length) when $d_c(\alpha) = 1$. Therefore g_{d_c} can control the depth of the contact. The behavior presented above leads us to the plot of $d_c(\alpha)$ depicted in Figure 4(a).

Figures 5(a,b) illustrate the relation between (a) the plot of a contact composition operator $g_{d_c}(f_1, f_2)$ and (b) its effect on the composition of two scalar fields f_1 and f_2 for an intermediate value of d_c. In Figure 5(a), the abscissa (resp. ordinates) are values of f_1 (resp. f_2) and vertical (resp. horizontal) lines represents its iso-surfaces. The operator defines the way the iso-surfaces of f_1 and f_2 are combined to produce the resulting scalar field f_k. Values of g_{d_c} are those of f_k when $f_k(\mathbf{p}) = g_{d_c}(f_1(\mathbf{p}), f_2(\mathbf{p}))$. Here, $g_{d_c} = 0.5$ on the 0.5 iso-surface of f_1 (in yellow) up to its intersection with

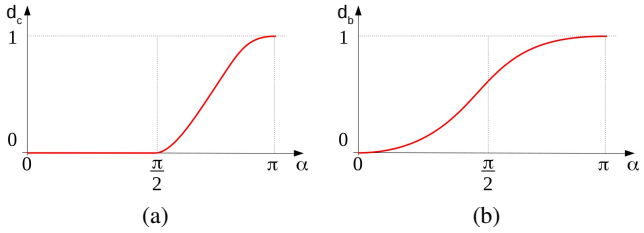


Figure 4: *(a) Plot of $d_c(\alpha)$ to get contact at joints, and (b) $d_b(\alpha)$ to mimic the inflation of tissues in contact situation.*

the 0.5 iso-surface of f_2 (in purple), and on the 0.5 iso-surface of f_2 up to its intersection with the 0.5 iso-surface of f_2. The result is the union of these two surfaces. In addition, $g_{d_c} = 0.5$ on the green line which starts at the intersection of the combined 0.5 iso-surfaces and goes inside the fields f_1 and f_2 following $f_1 = f_2$. This part of $g_{d_c} = 0.5$ is the contact surface that is shown with the same color on f_k in Figure 5(b). This line always starts at the intersection of the combined primitives and its length is controlled by d_c. Other iso-lines of g_{d_c} smoothly link values of f_1 to values of f_2 by following the shape of $g_d = 0.5$, resulting in a smooth distance field around $g_{d_c} = 0.5$ in Figure 5(a) and thus around $f_k = 0.5$ in Figure 5(b).

By designing the same operator, but without contact (i.e., when $d_c = 0$), we obtain the clean-union operator presented in Figure 5(c) whose effect on scalar fields is depicted in Figure 3(c). On the other hand, with a contact line of maximal length (i.e., when $d_c = 1$), we get the full contact operator shown in Figure 5(d).

Our second operator g_{d_b} is built similarly. It interpolates both the depth of the contact and the inflation of the bulge-in-contact between a clean union operator when $d_b(\alpha) = 0$ (no contact and no bulge, Figure 6(a)) and a full contact operator with maximal bulge when $d_b(\alpha) = 1$ (Figure 6(c)). Figure 6(b) illustrates this operator with intermediate contact length and bulge-in-contact ($d_b(\alpha) = \frac{1}{2}$). The bulge-in-contact is performed by the inflation of the 0.5 iso-lines of f_1 and f_2 in Figures 6(b,c). During the animation of a finger, the parameter $d_b(\alpha)$ is set as illustrated in Figure 4(b). As we can see, in that case, the bulge starts at low bending angles. Results produced by this operator are illustrated in Figure 13.

In practice, the composition operator g_{d_c} is applied by default at each skeleton joint, but the user can select g_{d_b} if desired. The binary composition tree (where implicit primitives are the leaves and composition operators are the nodes) is then built as follows: We start by creating nodes that link pairs of primitives with bulge-in-contact composition g_{d_b}. This forms the bottom of the tree. Then all the other nodes are added on top using gradient-based contact g_{d_c}, until all primitives are combined within a single tree. This specific composition enables us to achieve both the required bulge surfaces at fingers and the contacts we are looking for.

4.2 Bi-harmonic evaluation

The advantage of using scalar fields with compact support is that they are bounded in space and thus, can be stored in 3D grids enabling fast evaluation. It is the same for composition operators: we pre-compute and store them in 3D grids for the three entries (f_1, f_2, d).

Most advanced composition operators, such as those based on gradient blending, have very complex closed form equations, so some of them must be evaluated numerically, for instance, using binary search. Building new composition operators this way is a long

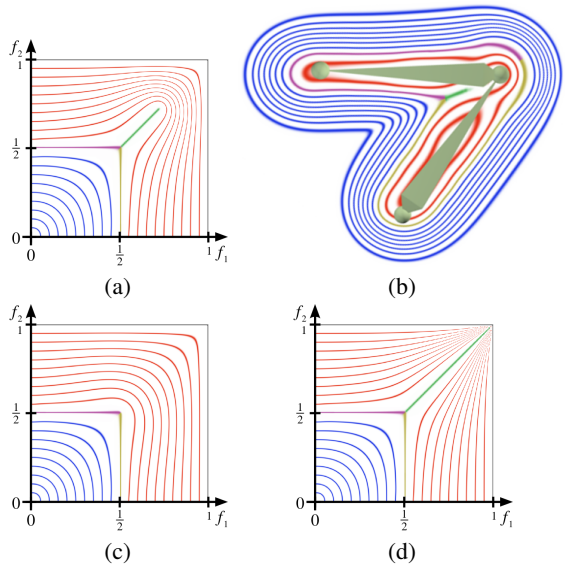


Figure 5: *(a) Our contact operator $g_{d_c}(f_1, f_2)$ for an intermediate value of d_c and (b) its application to the composition of two field functions articulated by a joint. In all figures, the contact surface is in green. (c) Clean-union operator ($d_c = 0$, no contact) and (d) full contact operator ($d_c = 1$).*

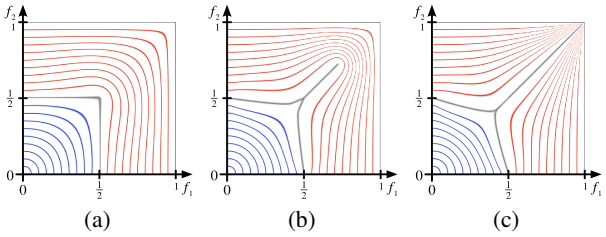


Figure 6: *Our gradient-based bulge-in-contact operator g_{d_b}. It interpolates between (a) a clean union ($d_b(\alpha) = 0$ no contact and no bulge) and (c) a full contact with maximal bulge ($d_b(\alpha) = 1$). In (b), this operator for an intermediate value of d_b.*

and tedious task: one must first derive intricate equations for each new operator and find the appropriate numerical evaluation method. Fortunately, in the case of skinning, only C^1 continuity is required for composition operators. This enables us to propose a simpler and more general way of constructing C^1 operators g directly into a discrete grid.

Our solution is to use bi-harmonic interpolation of some constraints: that is, the function g must satisfy the fourth-order bi-Laplacian equation $\Delta^2 g = 0$, where Δ is the Laplacian differential operator, while matching constraints at specific input values, as those listed in the previous section. This choice ensures a minimal continuity of order C^1. We directly solve this partial differential equation for each 2D slice (f_1, f_2) of the grid using a standard finite difference discretization. We use Dirichlet constraints on the system boundaries. The grid boundaries constraints are set according to the specific boundary properties of composition operators expressed by Canezin et al. [2013] (Equation 1) and inner grid constraints are placed along the profile curve defining the 0.5 iso-curve of $g(f_1, f_2)$ (Equation 2). Formally, we set:

$$g(f_1, 0) = f_1\,,\ g(0, f_2) = f_2\,,\ g(f_1, 1) = g(1, f_2) = 1, \qquad (1)$$

and

$$g(f_1, f_2) = 0.5 \quad \forall (f_1, f_2) \in \text{profile curve}. \tag{2}$$

In addition to these Dirichlet boundary conditions, the normal derivatives have to be constrained to zero on the $g(f_1, 0)$ and $g(0, f_2)$ axes to ensure a C^1 continuity at the boundaries of g.

Even though bi-harmonic interpolation does not guarantee the monotonicity of the function variation between constraints, we found the practical results satisfactory in our case without having to resort to more involved quadratic programming solvers [Jacobson et al. 2011; Jacobson et al. 2012].

Our different operators are thus generated by constraining $g = 0.5$ along a profile curve. For the contact operator g_{d_c}, the profile curve is composed of three line segments (in purple, yellow and green in Figure 5). The purple and yellow lines defined as the linear interpolation of the points $(0, 0.5)$, $(0.5, 0.5)$ and $(0.5, 0.5)$, $(0.5, 0)$ are fixed in all slices (f_1, f_2) $\forall d \in [0, 1]$. The green line is the linear interpolation of the points $(0.5, 0.5)$ and $(0.5 + \frac{d}{\sqrt{2}}, 0.5 + \frac{d}{\sqrt{2}})$.

Aside from the operator g_{d_c}, the profile curve is in general not aligned with grid vertices (e.g., bulge operator). We constrain the grid vertices of the intersected grid cells with the shortest signed distance to the profile curve.

The bulge-in-contact operator g_{d_b} is constructed following exactly the same procedure (see Figure 6). The only difference are the purple and yellow profile curves that are defined with cubic B-splines whose control points are interpolated with respect to d_b, between an aligned position when $d_b = 0$ and the maximal bulge when $d_b = 1$.

Using this evaluation procedure, our composition operators are precomputed once for all in 3D grids and at run time, a simple texture fetch is required for each evaluation. More generally, this approach enables the design of free-form composition operators: one only needs to specify a profile curve to generate a new operator.

5 Iso-surface tracking

One of the main challenge of our general implicit-skinning approach is to ensure that the initial mesh properly follows the iso-surface defined by the time-varying global scalar field f. Even though f is usually obtained through the composition of implicit primitives f_i which are rigidly transformed, the underlying implicit skin exhibits severe non-linear deformations around the joints. User defined non-linear deformations might also be applied to the f_i to mimic some complex effects (see section 6). Moreover, in our context the mesh embeds many spatially varying data such as relief details and texture layers. Therefore, the topology and density of the mesh cannot be arbitrarily changed, and each vertex must carefully track its position on the implicit surface to avoid unnatural distortions.

To this end, we propose to track the iso-surface incrementally from one animation step to the next using a three stage procedure:

1. Each vertex is moved by applying the transformation difference of its nearest bone, as defined by the initial mesh partitioning (Figure 3 (d)).
2. Vertices are projected onto their respective iso-value (derived from the rest pose) along the gradient direction using Newton iterations.
3. This yields a good initial guess on which we can apply a tangential relaxation scheme to account for the stretch of the implicit skin while reducing distortions.

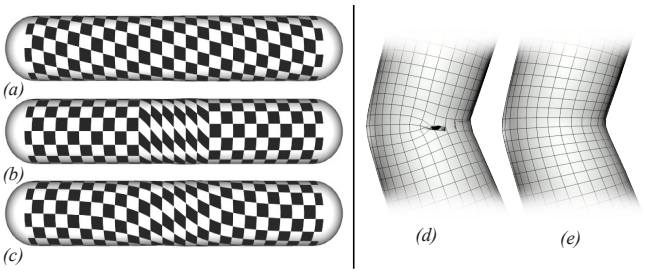


Figure 7: *Standard ARAP energy (a), (b), (d) is compared to our dual-quaternion based rotation targets (c) and (e). In (a) only the extremities of the tube are fixed while in (b) a larger part is fixed. Both cases lead to C^0 deformations on the constraints with an unplausible distribution of vertices inside the domain. In (d) triangle inversions occurred. In (c) and (e) these problems are avoided and our technique produces an adequate result.*

Since stages one and two are already described by Vaillant et al. [2013], the rest of this section focuses on the last one.

5.1 Tangential relaxation energy

The relaxation scheme must reduce the distortions while maintaining the mesh on the underlying surface. On the one hand, incrementally applying the relaxation based on mean value coordinate from the implicit skinning method does not reduce distortions sufficiently and contact is not captured adequately as illustrated in Figure 12(b). On the other hand, doing an accurate physical simulation of the skin would be prohibitively expensive for our purpose. We thus derived a novel relaxation scheme inspired by Sorkine et al. [2007] as-rigid-as-possible (ARAP) energy:

$$E(\{\hat{\mathbf{p}}_i\}) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)\|^2 , \tag{3}$$

where $\mathbf{p}_i$ are the vertex positions in rest pose, $\hat{\mathbf{p}}_i$ are the unknown final positions, $\mathbf{R}_i$ is a 3×3 rotation matrix, and $\mathcal{N}(i)$ denotes the one-ring neighborhood of vertex i. The weights w_{ij} are classically defined from the cotangent formula [Meyer et al. 2002], i.e., $w_{ij} = \frac{1}{2}(\cot(\alpha_{ij}) + \cot(\beta_{ij}))$, where α_{ij}, β_{ij} are the angles opposite to the edge (i, j).

As motivated and detailed below, in our relaxation scheme the rotations $\mathbf{R}_i$ are prescribed a priori taking advantage of the specificity of skinning deformations. Indeed, the energy E is by construction invariant to translation while the rotation $\mathbf{R}_i$ locally cancels rigid transformations. In most applications of this kind of energy, including iso-surface tracking [Bouthors and Nesme 2007], $\mathbf{R}_i$ is computed from the polar decomposition of the Jacobian of the unknown deformation. In addition to being an expensive process, this strategy exhibits several drawbacks. Firstly, this inserts a non-linear component in the energy, E which thus requires additional iterations to be minimized while introducing local minima. In addition, as illustrated in Figure 7(a,b), there is no smoothness constraints on the target rotations $\mathbf{R}_i$ and the result is only C^0 at the constraint boundaries. Moreover, we observed during animation that vertices appear to slide over the implicit skin instead of behaving as if the skin had elasticity, resulting in an unrealistic animation of the skin. Finally, as depicted in Figure 7(d), this strategy is well known to be subject to triangle inversions which are very tedious to avoid, even on a planar domain [Schüller et al. 2013].

We address all these drawbacks by fixing *a priori* the target rotations $\mathbf{R}_i$. In order to leverage a coherent behavior over time and to

recover the rest-pose, these rotations must not depend on the positions of the mesh vertices of the current animation step but only on the rest pose and affine transformations of the bones.

Following the first stage of our tracking procedure, a first choice consists of using the rotation associated with the nearest bone of vertex i for $\mathbf{R}_i$. As illustrated in the first row of Figure 8, this strategy yields promising results despite the discontinuous nature of the underlying rotation field. An effective solution thus consists of performing a blend of the bone transformations as in standard geometric skinning methods. In this research we use the dual-quaternion interpolation scheme [Kavan et al. 2008] with default harmonic weights [Baran and Popović 2007]. As discussed in more details in Section 6 and illustrated in Figure 8, we emphasize that both the choice of the blending technique and weights have only subtle effects on the final distribution of the vertices, and they do not affect the final shape or robustness of the tracking. Figures 7(c,e) show the superiority of our approach that better localize continuous tangential deformations and prevents triangle inversions.

5.2 On-surface constraints

The energy E has to be minimized while maintaining the vertices to their own field value. This is a non-linear constraint which implies an iterative scheme in which $\hat{\mathbf{p}}_i$ is obtained through a sequence $\hat{\mathbf{p}}_i^0, \hat{\mathbf{p}}_i^1, \ldots$, where $\hat{\mathbf{p}}_i^0$ corresponds to the initial projection onto its respective iso-value, as obtained through the aforementioned second stage of the tracking system. At each iteration of the minimization, we first locally linearize this constraint by enforcing each vertex to lie on its respective current tangent plane. This can be accomplished by writing the unknown position $\hat{\mathbf{p}}_i^{k+1}$ as:

$$\hat{\mathbf{p}}_i^{k+1} = \hat{\mathbf{p}}_i^k + \begin{bmatrix} \mathbf{u}_i^k & \mathbf{v}_i^k \end{bmatrix} \begin{pmatrix} u_i^{k+1} \\ v_i^{k+1} \end{pmatrix}, \tag{4}$$

where $\mathbf{u}_i^k$, $\mathbf{v}_i^k$ are two arbitrary tangent vectors of the current iso-surface at $\hat{\mathbf{p}}_i^k$, and u_i^{k+1}, v_i^{k+1} are the 2D coordinates of $\hat{\mathbf{p}}_i^{k+1}$ in this local frame. Minimizing E under this constraint is still a linear and very sparse problem. Since there is no need to completely solve this problem before projecting the solution on the surface and iterating, we interleave one Jacobi iteration minimizing E with one Newton iteration projecting $\hat{\mathbf{p}}_i^{k+1}$ onto its respective field value. At each iteration, an approximation of the local tangent frame is obtained for free from the gradient computed during the Newton iteration.

The choice for Jacobi iterations is motivated by its simplicity to be parallelized on the GPU. Higher performance may be achieved using the conjugate-gradient method which can also be efficiently implemented on GPUs [Weber et al. 2013]. In order to guarantee the availability of a very good initial guess $\hat{\mathbf{p}}_i^0$ and to avoid projection issues, the skeleton animation between two successive frames is decomposed into several virtual frames such that the maximal rotation angle of a bone is smaller than a third of the current bending angle. The intermediate skeleton configurations are linearly interpolated. Such a decomposition has little impact on the performance because a smaller animation step leads to a faster convergence.

6 Implementation and results

6.1 Implementation

In our prototype implementation, the deformation part of our skinning system is entirely executed on the GPU using CUDA. Parallelization is performed on a per-vertex basis with one kernel responsible for the computation of the initial guesses (rigid transformations and Newton projections) and a second kernel performing one iteration of the minimization (one Jacobi iteration, plus one Newton projection). This second kernel is applied multiple times per animation step until convergence. To accelerate the evaluation of the scalar field f, the scalar fields f_i are indexed in a 32^3 grid, and both our composition operators g_k and scalar fields f_i are stored in 3D textures and evaluated using hardware trilinear interpolation.

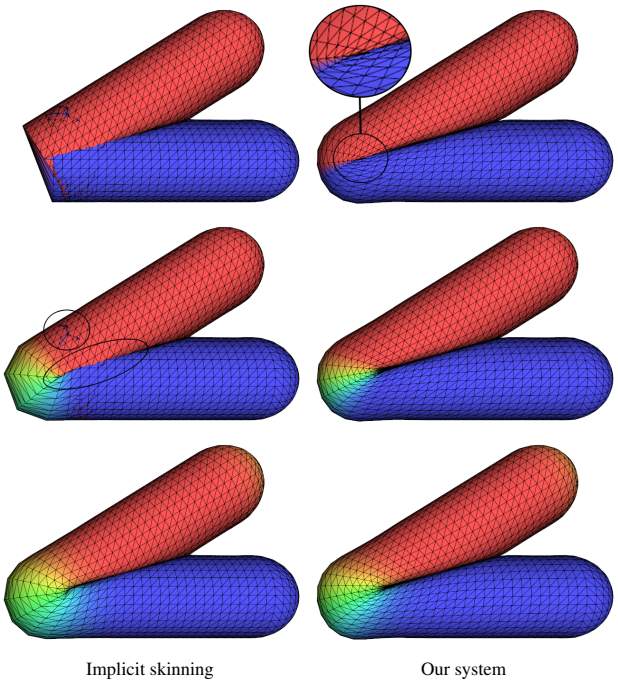


Figure 8: *Comparison of the sensitivity of the vertex distribution regarding variations of skinning weights (shown in color code) for the Implicit skinning (left) and our method (right). From top to bottom, skinning weights are rigid, smooth and smoother respectively.*

Results were produced on an Intel Core i5-3570K at 3.4GHz, with 8GB of memory and a Geforce 670 GTX. The memory consumption is exactly the same as for the implicit skinning method: for the models presented in this paper, from 10 to 30MB of extra texture memory are required to store the operators and field functions. During an interactive session, joints are usually adjusted one at a time. In this case, deformations occur mostly locally: only a small fraction of vertices have to be projected and relaxed, hence the density of the mesh has a small impact on performance. For a single joint, we thus observed a rate between 30 to 300 frames per second. More detailed statistics and performance results are given in Table 1 for complete animations and different time steps. As expected, the number of minimization iterations and thus the computation time highly depends on the magnitude of the skeleton transformation between two animation steps.

6.2 Results

Provided a mesh with an animation skeleton, our system automatically produces a default solution: partitioning and field-functions f_i are automatically computed as described by Vaillant et al. [Vaillant et al. 2013], and skin elasticity and self-contacts are automatically and globally resolved during the animation.

As demonstrated in Figure 9, our approach produces satisfactory behavior without requiring any tedious tuning of skinning weights, while previous work still exhibit distortions even after an artist edits

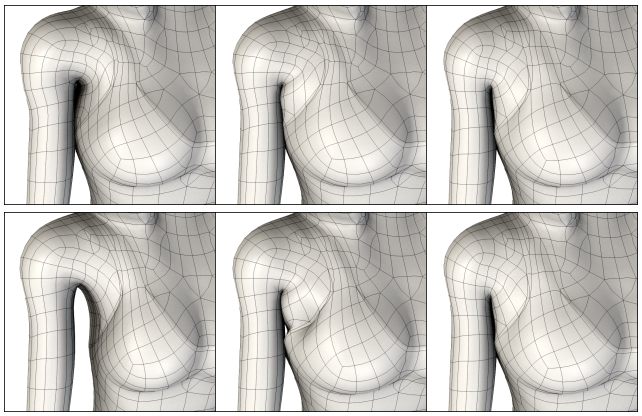


Figure 9: *Comparison of different skinning method using manually edited skinning weights (top), and automatic weights (bottom).*

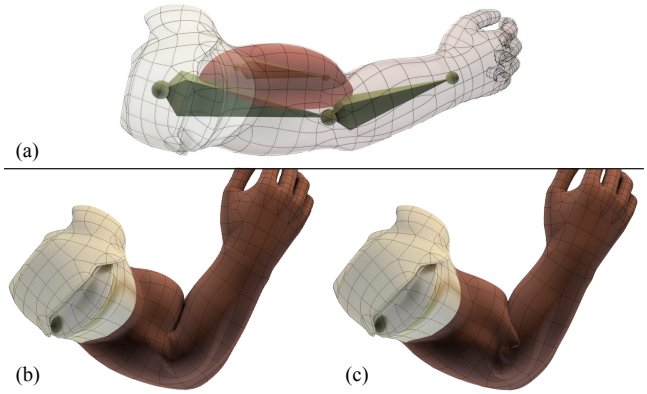


Figure 10: *Reproduction of the inflation of a bicep by the addition of an extra primitive (a) and its animation using our system (b). Owing to the prominent shape of the muscles, implicit skinning yields projection artifacts (c).*

the weights manually. The sensitivity of our method with respect to skinning weights is further evaluated and compared on a simpler configuration in Figure 8. Recall that in our system such weights are only used to blend the target rotation in the minimization of our tangential energy. Therefore, even large changes in the weights leads only to small differences in the distribution of the vertices, and the shape of the implicit skin remains unchanged. In contrast, the projection method of the implicit skinning technique fails if the weights are not smooth enough, hence, further editing of the skinning weights is often needed.

	#vertices	#bones	time step (rad)	#Jacobi iterations	full animation	Imp skinning
Hand	31.7k	21	0.5	600	650ms	83ms
Hand	31.7k	21	0.05	330	220ms	83ms
Armadillo	16.4k	43	0.05	170	70ms	23ms
Jeff	13.3k	45	0.5	300	100ms	30ms
Jeff	13.3k	45	0.05	100	50ms	30ms
Jeff	13.3k	45	0.005	10	24ms	20ms

Table 1: *Statistics and performance of our system and implicit skinning for various models and time steps when all joints are animated simultaneously.*

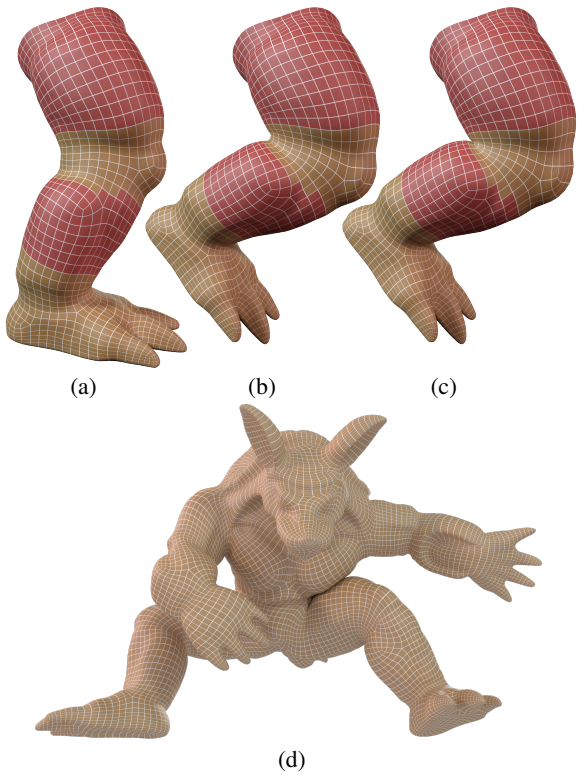
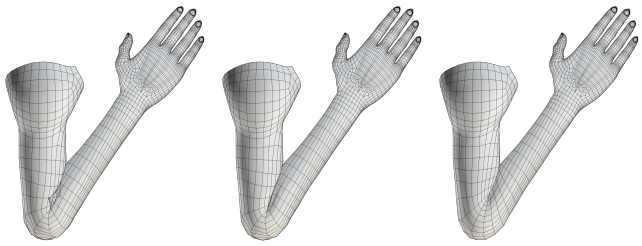


Figure 11: *The Armadillo's knee (a) is bent using (b) our default solution, and (c) by disabling the tangential relaxation for red areas. (d) Global self-contact between the belly and the leg.*

The global nature of the skin sliding effect produced by our tangential relaxation is especially visible on the character's belly of Figure 1-left. In contrast to geometric or implicit skinning, with our method the belly button is stretched as the torso twists. As shown in Figure 11-top, this skin elasticity can be easily controlled by painting regions where the tangential relaxation is not wanted. In such regions, vertices are only rigidly transformed according to their nearest bone.

Global contacts can be observed in Figures 11(d) and 14. The shape of the deformation can also be adjusted by choosing between different types of composition operator to enable, for instance, subtle bulge in contact where this is desired as on the finger of Figure 13. More advanced deformations can also be performed by designing and controlling additional field functions. For instance, in Figure 10 the user added an extra implicit primitive with its associated bone to mimic the contraction of a bicep by applying a scaling proportional to the arm bending angle. Such a manipulation would not be possible without a proper tracking system taking care of the skin elasticity and contacts.

In addition to all these novel possibilities, our system permits reaching extreme bending angle without introducing artifacts. This can be seen in Figure 1(e) for a bending angle of around 150 degrees. This high robustness is due to the combination of both a proper modeling of surface contacts and the incremental nature of our novel surface tracking mechanism. Figure 12(a) shows that with a standard union operator, implicit skinning exhibits artifacts if it is implemented with skinning weights that are not smooth enough. In Figure 12(b) our novel contact operator, used alone in implicit skinning (i.e., without our novel tracking system), improves the projection artifacts but does not reduce mesh distortions. Finally, our novel tracking algorithm (Figure 12(c)) enables even larger bending

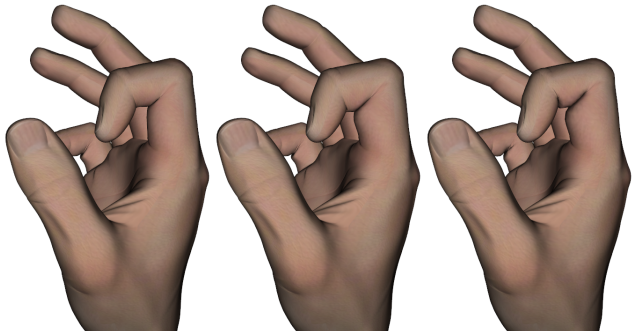


(a) Implicit Skinning using a standard union

(b) Implicit Skinning using our union with contact

(c) Our complete system

Figure 12: *(a) Without proper modeling of the surface contact, vertices tend to be projected towards the exterior of the contact region thus leading to incorrect projections. (b) Our novel operators address this precise issue, unfortunately, tracking with implicit skinning arbitrarily extends the contact region in an uncontrollable manner. (c) Our system adequately tracks the skin fold.*



Our system with our default contact operator

Our system with bulge and contact

Implicit skinning with bulge

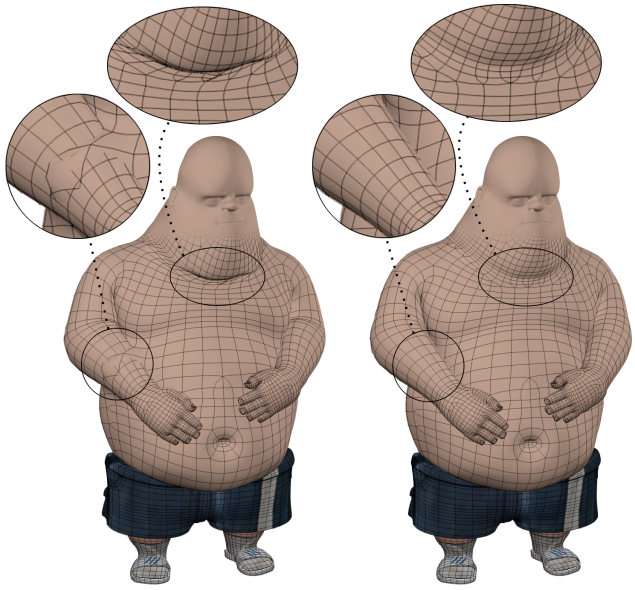
Figure 13: *Comparison of our bulge with contact operator.*

angles, a proper tracking of the fold depth and a reduction of mesh distortions.

6.3 Limitations

Owing to the incremental nature of our tracking technique, our approach gets closer to a physical simulation of the skin than the original implicit skinning method. Even though our approach is considerably simpler and faster than physical simulations, it still has some of their drawbacks. Firstly, the result for a given animation step is not entirely determined by the skeleton pose. Because of the non-linearity introduced by the projections on the implicit skin the result might depend on the complete history of the animation. For a similar reason, the result might also depend on the chosen time step. As with any simulation, choosing the right time step is a trade-off between the computation cost and the numerical stability: a very large time step will lead to very poor initial guesses with similar problems as implicit skinning, while a very small time step will be prohibitively expensive.

In practice, as long as no incorrect projection occurs during animation, our solution is almost insensitive to a particular time step. The automatic solution proposed in Section 5.2 to choose an appropriate time step strives to make sure that the initial guess obtained through rigid transformation of the previous animation-step does not lead to a projection in the wrong direction. The proposed solution is rather conservative as, for instance, we observed that bending an arm from a fully open to a $\pi/2$ angle in a single time-step is correctly handled by our tracking system. Higher performance could thus be achieved



Implicit Skinning

Our complete system

Figure 14: *Comparison of our system to implicit skinning on the model shown in rest pose in Figure 1a. The arm collides too deeply with the belly to be properly recovered by implicit skinning while the chin is unrealistically creased. These issues are properly addressed by our system.*

by relaxing the proposed heuristic. An implementation could also use large time steps, and dynamically subdivide the animation when incorrect projections or triangle inversions are detected.

7 Conclusion

In this work, we have presented a new skinning technique that takes advantage of the best features of implicit skinning, and makes the method more robust and suitable for production pipelines. We showed how to properly model and control contact surfaces using advanced implicit composition operators. To this end, we designed a process for the creation of free-form composition operators based on biharmonic interpolation over a discrete grid. The scope of this technical contribution goes beyond the context of implicit skinning, as this process enables interactive design of custom composition operators which is a central topic in implicit modeling. We also showed how to track a deformable implicit surface by deriving a linearized relaxation energy from Sorkine et al.'s as-rigid-as-possible energy with some *a priori* knowledge on the expected deformation that we can extract from the animation skeleton. As a result, our skinning system is robust, handles contact even when surfaces are deeply interpenetrating, and maintains a plausible solution where implicit skinning fails. The method thus reduces the amount of necessary user interaction, while maintaining interactive performance, and enables new features such as skin elasticity.

Based on these well controlled skin deformations, several new avenues could be explored. For instance, very fine control of the tangential deformations could be studied by setting a few vertices in key-frames. Also, performances could be improved by investigating more efficient GPU solvers and better automatic adjustment of the time steps. Automatically computing the scalar fields at complex joints is still a challenging problem which deserves some future specific attention. Finally, the more general use of our technique on deformable dynamic 3D objects is still to be explored.

8 Acknowledgments

This work was partially funded by the IM&M project (ANR-11-JS02-007) and by the advanced grant EXPRESSIVE from the European Research council (ERC-2011-ADG-20110209). Partial funding also came 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. We wish to thank Laura Paiardini from Inria Grenoble for providing us with the very nice Armadillo's skeleton animation and Garrett Pond from Brigham Young University for the modeling of the Jeff model (the big guy).

References

ALI-HAMADI, D., LIU, T., GILLES, B., KAVAN, L., FAURE, F., PALOMBI, O., AND CANI, M.-P. 2013. Anatomy transfer. *ACM Trans. Graph. 32*, 6, 188:1–188:8.

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

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

BOUTHORS, A., AND NESME, M. 2007. Twinned meshes for dynamic triangulation of implicit surfaces. In *Graphics Interface*.

CANEZIN, F., GUENNEBAUD, G., AND BARTHE, L. 2013. Adequate Inner Bound for Geometric Modeling with Compact Field Function. *Computer & Graphics (proceedings of SMI 2013) 37*, 6 (July), 565–573.

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.

DE GROOT, E., WYVILL, B., BARTHE, L., NASRI, A. H., AND LALONDE, P. 2014. Implicit decals: Interactive editing of repetitive patterns on surfaces. *Comput. Graph. Forum 33*, 1, 141–151.

DIONNE, O., AND LASA, M. D. 2014. Geodesic binding for degenerate character geometry using sparse voxelization. *IEEE Transactions on Visualization and Computer Graphics 20*, 10, 1–1.

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 25*, 4.

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.

JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *Computer Graphics Forum (proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing) 31*, 5, 1577–1586.

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, I3D '05, 9–16.

KAVAN, L., COLLINS, S., ŽÁRA, J., AND O'SULLIVAN, C. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics 27* (November), 105:1–105:23.

KIM, Y., AND HAN, J. 2014. Bulging-free dual quaternion skinning. *Comp. Anim. Virtual Worlds 25*, 3-4, 323–331.

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, SIGGRAPH '00, 165–172.

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, SIGGRAPH '11, 37:1–37:12.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. VisMath*, 35–57.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. In *ACM SIGGRAPH 2003 Papers*, ACM, 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.

PASKO, A., ADZHIEV, V., SOURIN, A., AND SAVCHENKO, V. 1995. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer 11*, 8, 429–446.

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

RODRIAN, H.-C., AND MOOCK, H. 1996. Dynamic triangulation of animated skeleton-based implicit surfaces. In *Implicit Surfaces '96*.

SCHÜLLER, C., KAVAN, L., PANOZZO, D., AND SORKINE-HORNUNG, O. 2013. Locally injective mappings. *Computer Graphics Forum (proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing) 32*, 5, 125–135.

SONKA, M., HLAVAC, V., AND BOYLE, R. 2007. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, 109–116.

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

TENG, Y., OTADUY, M. A., AND KIM, T. 2014. Simulating articulated subspace self-contact. *ACM Transactions on Graphics 33*, 4.

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.

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

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.

WEBER, D., BENDER, J., SCHNOES, M., STORK, A., AND FELLNER, D. 2013. Efficient gpu data structures and methods to solve sparse linear systems in dynamics applications. *Computer Graphics Forum 32*, 1, 16–26.

WITKIN, A. P., AND HECKBERT, P. S. 1994. Using particles to sample and control implicit surfaces. In *ACM Computer Graphics (SIGGRAPH '94)*, 269–277.

WYVILL, B., GUY, A., AND GALIN, E. 1999. Extending the csg tree - warping, blending and boolean operations in an implicit surface modeling system. *Comput. Graph. Forum 18*, 2, 149–158.

XU, C., AND PRINCE, J. L. 1998. Snakes, shapes, and gradient vector flow. *IEEE TRANSACTIONS ON IMAGE PROCESSING 7*, 3, 359–369.