

# Free-form Skeleton-driven Mesh Deformations

Shin Yoshizawa<sup>†</sup>

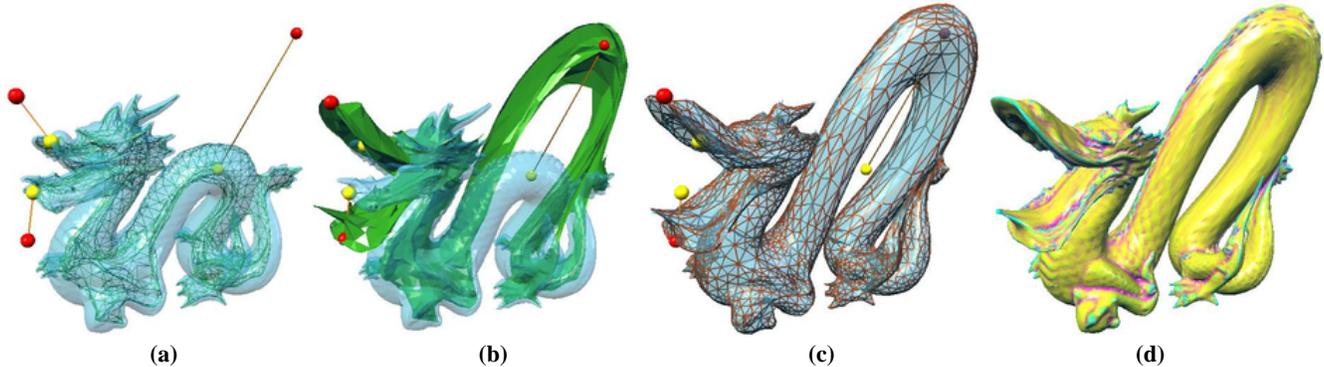
Alexander G. Belyaev<sup>†,‡</sup>

Hans-Peter Seidel<sup>†</sup>

<sup>†</sup> Computer Graphics Group, Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany

<sup>‡</sup> University of Aizu, Aizu-Wakamatsu 965-8580 Japan

{shin.yoshizawa | belyaev | hpseidel}@mpi-sb.mpg.de



**Figure 1: A skeleton-driven mesh deformation.** (a) A dragon model (100K triangles), its skeletal mesh (6K triangles), and control points used to produce a global deformation of the skeletal mesh. (b) A free-form deformation of the skeletal mesh. (c) Reconstruction of an approximate dragon model (6K triangles), from the deformed skeleton. (d): The final deformation is obtained by applying DSS [18] to the approximate dragon model; coloring by mean curvature is used for a quality evaluation of the deformed mesh.

## ABSTRACT

In this paper, we propose a new scheme for free-form skeleton-driven global mesh deformations. First a Voronoi-based skeletal mesh is extracted from a given original mesh. Next the skeletal mesh is modified by free-form deformations. Then a desired global shape deformation is obtained by reconstructing the shape corresponding to the deformed skeletal mesh. We develop a mesh fairing procedure allowing us to avoid possible global and local self-intersections of the reconstructed mesh. Finally, using a displaced subdivision surface representation [18] improves the speed and robustness of our approach.

## Categories and Subject Descriptors

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

## General Terms

Algorithms, Design, Experimentation

## Keywords

Free-form global shape deformations, Voronoi-based skeletal mesh

## 1. INTRODUCTION

Local and global free-form shape deformation techniques are important for many computer graphics and shape modeling applications.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

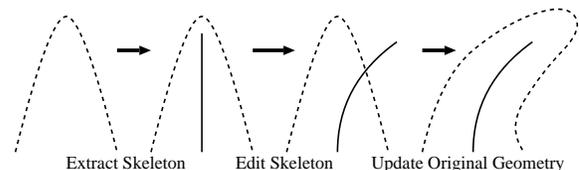
SM'03, June 16–20, 2003, Seattle, Washington, USA.

Copyright 2003 ACM 1-58113-706-0/03/0006 ...\$5.00.

Multiresolution mesh representations [26, 17, 14, 16, 18] are often used for modeling global natural-looking shape deformations. Recently skeleton-driven global free-form shape deformations drew much attention [19, 21, 7, 6] because they are well-suited for large-scale shape deformations and, therefore, can be used in numerous applications in the computer game industry.

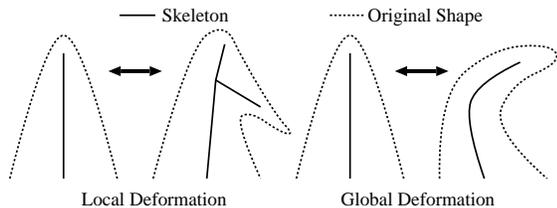
In this paper, we develop a new scheme for free-form skeleton-driven global shape deformations. Given a triangle mesh approximating 3D shape, first we build a Voronoi-based skeletal mesh. The skeletal mesh inherits the connectivity of the original mesh and there is one-to-one correspondence between the vertices of the original and skeletal meshes. We use mesh evolutions in order to improve the skeletal mesh. The original mesh is then represented as the set of displacements applied to the vertices of the improved (smoothed) skeletal mesh. The mesh deformation process is combined from deformations of the smoothed skeletal mesh and the displacement field. We also use mesh evolutions to remove local and global self-intersections of the deformed mesh. Finally, since our shape representation resembles the displaced subdivision surfaces [18], we enrich our mesh deformation approach by a multi-scale technique. Fig. 1 above gives some impression on how our approach works.

The basic (and very simple) idea of our skeleton-based approach to global shape deformations is sketched in Fig. 2.



**Figure 2: Skeleton-based shape deformation.**

Notice that usually a local shape deformation corresponds to a skeleton bifurcation (branching) while a global shape deformation corresponds to skeleton bending, as seen in Fig. 3.



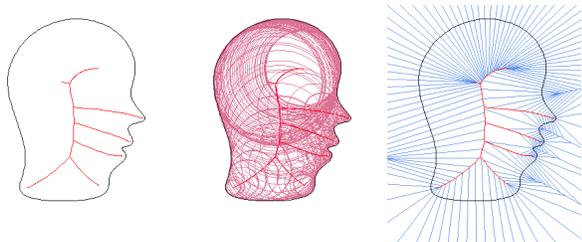
**Figure 3: Local vs. global shape deformations. Left: local shape deformations usually produce new branches of the skeleton. Right: skeleton bendings correspond to natural-looking global shape deformations.**

The rest of the paper is organized as follows. In Section 2 we explain how to construct the skeletal mesh. In Section 3 we describe our global skeleton-based shape deformation approach. Section 4 is devoted to combining our approach with the DSS technique [18]. We conclude in Section 5.

## 2. VORONOI-BASED SKELETAL MESH

A skeleton (medial axis) of a figure is the closure of those points inside the figure that have more than one closest point among the points of the boundary of the figures. The skeleton was originally introduced by Blum [8, 9] for 2D shapes in order to provide with a symmetry-based shape representation for shape perception and recognition purposes. In 3D, the skeleton has been recently studied in connection with a research on shape organization [13] and shape manipulation [22, 7, 6].

Given an oriented surface, the skeleton can be also described as the set of singularities of the signed distance function from the surface [4]. Thus we distinguish two skeletons of an oriented surface, the inner skeleton and the outer skeleton (one of them may be empty, for example, for a surface bounding a convex figure).



**Figure 4: Left: a closed 2D curve and its skeleton. Middle: the skeleton is formed by the centers of all inner bitangent circles. Right: the skeleton is approximated by the vertices of the Voronoi diagram generated by points scattered densely over the curve.**

The skeleton of a figure is very sensitive to small perturbations of the boundary of the figure: small perturbations of the boundary may result in large changes of the skeleton structure. Practical extraction of the skeleton of a 3D shape is usually based on 3D Voronoi diagram techniques [23, 2], see also references therein. Fig. 4 presents a 2D example demonstrating how the skeleton of a figure can be approximated by Voronoi vertices corresponding to points scattered densely over the boundary of the figure. Fig. 4 demonstrates also how sensitive the skeleton of a figure is with respect to small perturbations of the boundary of the figure.

Recently several improvements over the basic technique developed in [2] were proposed [3, 12, 15]. For our needs, we adapt the approach developed very recently in [15] where it was proposed to approximate the skeleton of a mesh by a skeletal mesh having the same connectivity as the original mesh. The vertices of the skeletal mesh are in one-to-one correspondence with the vertices of the original triangle mesh and the skeletal mesh inherits the connectivity of the original mesh. It allows for editing the skeleton by standard mesh processing tools.

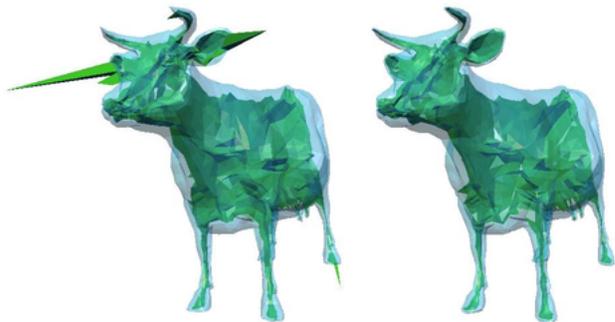
Given a mesh  $\mathcal{M}$ , our first goal is to extract an approximate skeletal mesh  $\mathcal{S}$  such that

$$\mathcal{M} = \mathcal{S} + dN, \quad (1)$$

where  $N$  is the field of unit mesh normals defined at the vertices of  $\mathcal{M}$  and  $d$  is the set of distances from the vertices of  $\mathcal{M}$  to the corresponding vertices of  $\mathcal{S}$  along  $N$ .

Relation (1) is the core of our approach. It allows us to edit the original mesh  $\mathcal{M}$  via modifying its skeletal mesh  $\mathcal{S}$ . Below we explain how to achieve a robust extraction of the skeletal mesh and build relation (1).

*Mesh smoothing.* Since the skeleton of a shape approximated by a mesh is very sensitive to the mesh quality, we first apply the bilaplacian tangent flow [24] to the mesh in order to improve the mesh quality. If the step-size of the bilaplacian tangent flow is sufficiently small, the flow keeps almost does not affect the mesh geometry while improving the aspect ratios of the mesh triangles.



**Figure 5: Left: a cow mesh and its inner skeleton, notice that the skeleton intersects the cow mesh. Right: the cow mesh improved by several iterations of the bilaplacian tangent flow and its skeleton improved original cow mesh via 100 bilaplacian tangent flow provides a good skeleton mesh.**

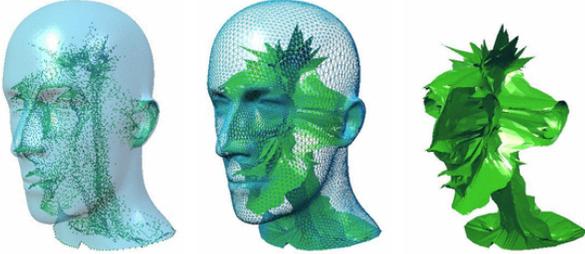
This preprocessing step improves dramatically the quality of the skeleton, as demonstrated in Fig. 5. The left image shows an original cow mesh and its inner skeletal mesh. The skeletal mesh intersects the cow mesh while the true skeleton is located inside the cow model. The cow mesh in the right image is improved by several iterations of the bilaplacian tangent flow. The inner skeletal mesh of the improved cow mesh provides with a much better approximation of the true skeleton.

*Extraction of skeletal mesh.* Given a mesh, we use the Quickhull algorithm [5] to extract the Voronoi diagram of the mesh vertices. We estimate the tangent plane at each mesh vertex and use the mesh orientation in order to distinguish the inner and outer subsets of the Voronoi vertices. We remove the outer Voronoi vertices and, in the case of an open mesh, those inner vertices which are located outside some bounding box of the mesh.

In contrast to [2], for each mesh vertex we neglect its Voronoi pole, the farthest Voronoi vertex of the Voronoi region containing the mesh vertex. In theory [1] the poles provide a good approximation of the true skeleton if the original mesh is dense enough. However, according to our experiments, the poles have poor stability properties with respect to small perturbations (defects) of the mesh. Although we do not achieve an accurate reconstruction of the skeleton, stability is much more important for our purposes than accuracy.

A point corresponding to a mesh vertex is computed as the arithmetic mean of the remaining Voronoi vertices of the Voronoi region containing the mesh vertex. The displacement  $d \in \mathbf{d}$  is computed as distance between the mesh vertex and its corresponding point. Then the skeleton vertex corresponding to the mesh vertex is found as a point situated on the normal at the mesh vertex and displaced on the distance  $d$  from the mesh vertex. The normal at a mesh vertex is computed as the normalized area-weighting mean of the normals of the mesh triangles incident to the vertex.

Finally the obtained set of the skeleton vertices is equipped with the connectivity of the original mesh, as seen in Fig. 6. Roughly speaking, we do "vacuum-packing" of the original mesh onto the skeleton generated by averaged Voronoi poles.



**Figure 6: Voronoi-based skeletal mesh. Left: the mannequin head model and its inner Voronoi vertices. Middle: the mannequin head mesh connectivity is used to create the skeletal mesh. Right: the skeletal mesh (skeleton) of the mannequin head model.**

### 3. BASIC MESH DEFORMATION PROCESS

Consider a free-form deformation of the skeletal mesh  $\mathcal{S}$ . In our implementation we use a set of techniques developed in [25] although other free-form mesh deformations techniques can be used as well. A direct reconstruction of a deformed mesh from the deformed skeletal mesh according to (1) may produce severe self-intersections of the deformed mesh. So we use a homotopy method [20] to decompose the deformation into a sequence of  $L$  deformations connecting the original skeletal mesh  $\mathcal{S}_0 = \mathcal{S}$  and deformed skeletal mesh  $\mathcal{S}_L$ :

$$\mathcal{S}_j = \mathcal{S}_0 + j \frac{\mathcal{S}_L - \mathcal{S}_0}{L}. \quad (2)$$

Now the corresponding deformations of the original mesh are computed as

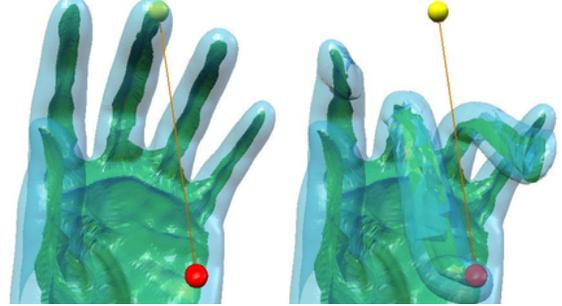
$$\mathcal{M}_j = \mathcal{S}_j + \mathbf{dN}_{j-1}, \quad j = 1, 2, \dots, L, \quad (3)$$

where  $\mathbf{N}_0$  is the field of unit mesh normals for  $\mathcal{M} = \mathcal{M}_0$  and  $\mathbf{N}_j$  is the field of unit mesh normals for  $\mathcal{M}_j$ . The scalar field of displacements  $\mathbf{d}$  is not changed during the deformation steps. According to our numerical experiments, the decomposition into  $L = 3$  steps delivers a satisfactory combination of quality and speed.

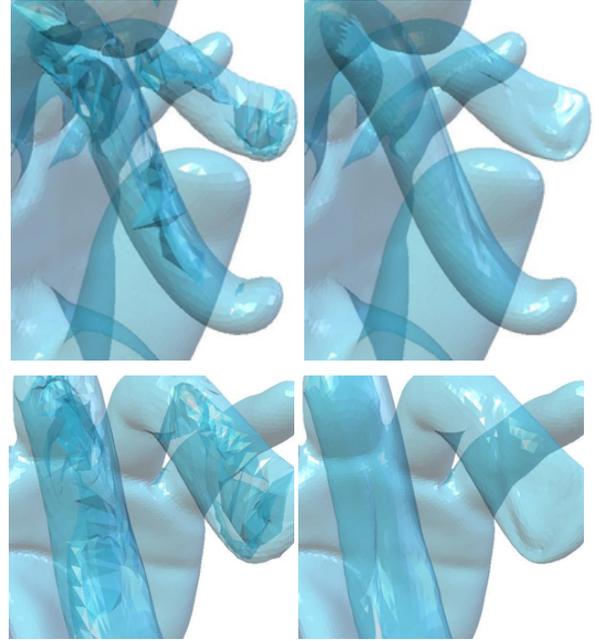
#### 3.1 Removing Folds and Protrusions

If a large skeleton deformation is applied, see, for instance, Fig. 7, the resulting deformed mesh  $\mathcal{M}_L$  may have some defects, as

demonstrated in the left images of Fig. 8. In this section, we explain how to remove such defects of the deformed mesh as folds and protrusions, as seen in Fig. 9. One possible way to avoid such mesh defects consists of reconstructing the deformed mesh  $\mathcal{M}_L$  from the deformed skeleton  $\mathcal{S}_L$  as the envelope of medial balls [3] centered at the vertices of  $\mathcal{S}_L$ . However it works well only for dense meshes. So we have chosen a different approach based on mesh evolutions.



**Figure 7: A large skeleton-based deformation of a hand model.**



**Figure 8: Left: zoomed parts of the deformed hand model from the right image of Fig. 7. Right: fixing mesh defects by (4).**

We consider the following mesh evolution

$$\frac{\partial \mathcal{M}}{\partial t} = -\alpha \Delta^2 \mathcal{M} - \mathbf{F} - \mathbf{V}, \quad \mathcal{M}(0) = \mathcal{M}_L, \quad (4)$$

where the negative bilaplacian  $-\Delta^2$  and force  $-\mathbf{V}$  are used for mesh relaxation and regularization purposes and force  $-\mathbf{F}$  pushes the evolving mesh towards the envelope of the medial balls.

We approximate the bilaplacian operator via the bi-umbrella operator [17]. The parameter  $\alpha > 0$  is not constant. Let us consider a mesh vertex  $M$  and its neighbors, compute the umbrella operators (vectors) for them, and count the number of those neighbors whose umbrella vectors form an obtuse angle with the umbrella vector at  $M$ . We assign  $\alpha = 0.25$  to  $M$  if the fraction that obtuse angles is

less than 0.3. Otherwise we set  $\alpha = 0$  at  $M$ .

We want to define the force  $\mathbf{F}$  such that  $-\mathbf{F}$  fits the evolving mesh to the envelope of the medial balls. For each triangle  $T$  of the deformed skeletal mesh let us consider the convex hull of the medial balls centered at triangle vertices. A general approach to compute the convex hull of a set of spheres can be found in [10]. However, in our simple case, the convex hull is computed analytically: we use the fact that the convex hull can be computed as the envelop of the balls centered inside the triangle and obtained by the trilinear interpolation of the balls centered at the vertices. We describe the envelop as an implicit function. Let us define a function  $w = E_T(P)$  at point  $P$  as the value of the implicit function at  $P$ . Now consider a mesh vertex  $M$ , the set of mesh triangles incident with  $M$  and their centroids  $C_j$ ,  $j = 1, \dots, n$ . The force  $\mathbf{F}$  at  $M$  is defined by

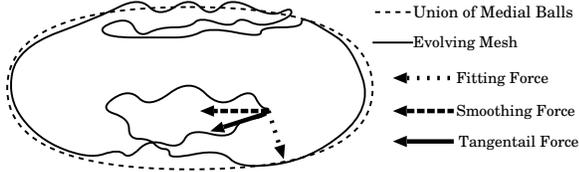
$$\mathbf{F}(M) = \frac{1}{n} \sum_{j=1}^n E_{T_j}(C_j) \nabla E_{T_j}(C_j),$$

where  $T_j$  is a deformed skeletal mesh triangle corresponding to the mesh triangle with centroid  $C_j$ . Notice that the force  $E \nabla E$  attracts the vertices to the zero level set of  $E$ .

The force  $\mathbf{V}$  is defined as the projection of the bilaplacian vector on the plane orthogonal to  $\mathbf{F}$

$$\mathbf{V} = \Delta^2 \mathcal{M} - (\Delta^2 \mathcal{M} \cdot \frac{\mathbf{F}}{|\mathbf{F}|}) \frac{\mathbf{F}}{|\mathbf{F}|}.$$

Fig. 9 explains why flow (4) eliminates mesh folds and protrusions.

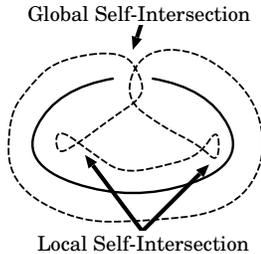


**Figure 9: Effect of (4).** Force  $\mathbf{F}$  pushes the mesh vertices towards the envelope of the medial balls. Two other forces in the right hand-side of (4), tangential force  $\mathbf{V}$  and smoothing force  $-\alpha \Delta^2 \mathcal{M}$ , are used to eliminate mesh folds and protrusions.

The right images of Fig. 8 demonstrate fixing defects of the deformed hand mesh by (4).

### 3.2 Eliminating Global and Local Self-Intersections

self-intersections, as sketched in Fig. 10. The deformed mesh still may have global and local self-intersections, as sketched in Fig. 10.



**Figure 10: Global and local self-intersections.**

Again we use a mesh evolution approach in order to eliminate global and local self-intersections of the deformed mesh. Consider a vertex  $M = (x, y, z)$  of the deformed mesh and its corresponding vertex  $S = (s_x, s_y, s_z)$  of the deformed skeleton. Let us define the function

$$g(x, y, z, S) = (x - s_x)^2 + (y - s_y)^2 + (z - s_z)^2 - d^2,$$

where  $d$  is the radius of the medial ball centered at  $S$ . Now we introduce a function  $f(x, y, z)$  whose zero level set approximates the envelope of the medial balls. Let us divide the bounding box (unit box) uniformly into  $20 \times 20 \times 20$  voxels  $G_{l,m,n}$ . We set

$$h_{l,m,n}(x, y, z) = \min \{g(x, y, z, S) : S \in G_{l,m,n}\},$$

where the minimum is taken over all skeleton vertices  $S$  that belong to the cell  $G_{l,m,n}$ . Then  $f(x, y, z)$  is defined for  $(x, y, z) \in G_{l,m,n}$  by

$$f(x, y, z) = \begin{cases} h_{l,m,n}(x, y, z) & \text{if } h_{l,m,n}(x, y, z) < 0; \\ \min \{h_{l,m,n}, h_{l\pm 1, m\pm 1, n\pm 1}\} & \text{otherwise.} \end{cases}$$

The mesh evolution we use to eliminate global and local self-intersections evolves each mesh by

$$\frac{\partial \mathcal{M}}{\partial t} = -f(\mathcal{M}) \nabla f(\mathcal{M}) - W \Delta^2(\mathcal{M}). \quad (5)$$

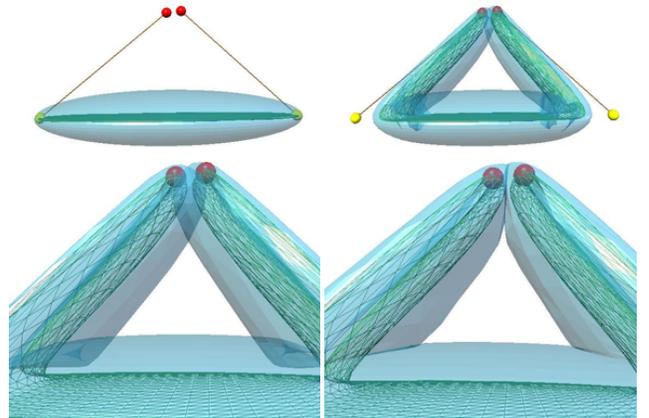
Here  $-f(\mathcal{M}) \nabla f(\mathcal{M})$ , the antigradient of  $\frac{1}{2} f^2$ , pushes the mesh vertices towards the zero level set of  $f(x, y, z)$ . The weight  $W(M)$  in (5) for a mesh vertex  $M \in \mathcal{M}$  is given by

$$W(M) = \frac{|f(M) - g(M, S)|}{\max_{M \in \mathcal{M}} |f(M) - g(M, S)|},$$

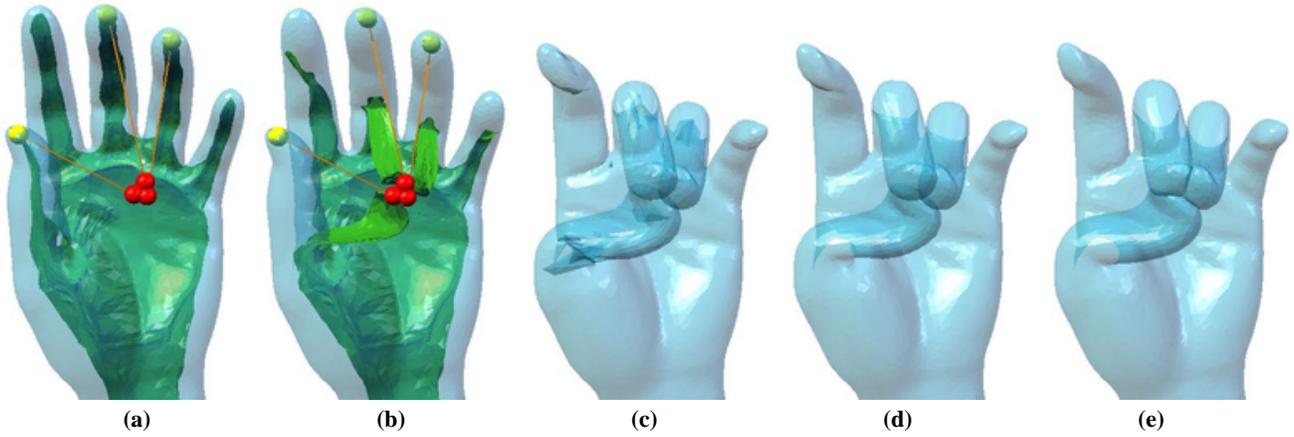
where  $S$  is the skeleton vertex corresponding to mesh vertex  $M$ .

Similar to (4) the bilaplacian term in (5) makes the flow more stable while another term in the left hand-side of (5) pushes the mesh vertices towards the envelope of the medial balls centered at the vertices of the deformed skeleton.

Mesh fairing with (4) and (5) is demonstrated in Fig. 11 for a large-scale deformation of an ellipsoid model.



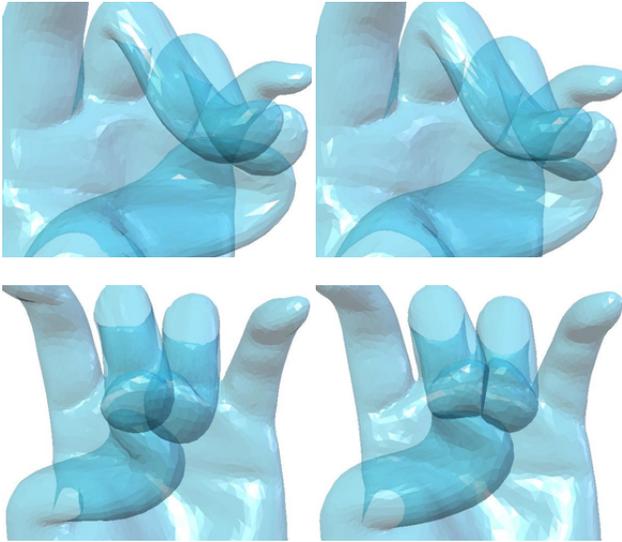
**Figure 11: Fairing global and local self-intersections.** Top-left: an ellipsoid, its skeletal mesh, and control points used to deform the skeletal mesh. Top-right: the ellipsoid is bended via a skeleton-based deformation. Bottom-left: folds and protrusions are removed by (4), however local and global self-intersections remain. Bottom-right: removing the self-intersections by (5).



**Figure 12: Basic mesh deformation process.** (a) The original hand mesh, its skeletal mesh, and control points to be used to deform the skeletal mesh. (b) A deformed skeletal mesh. (c) Folds and protrusions are observed in the deformed mesh. (d) The folds and protrusions are removed by (4); however global and local self-intersections are still presented. (e) The global and local self-intersections are eliminated by (5).

### 3.3 Gathering All Together

Our basic mesh deformation process is demonstrated in Fig. 12. Given a mesh, first its Voronoi-based skeletal mesh is extracted. Next a free-form deformation is applied to the skeletal mesh. Then a deformed mesh is reconstructed from the deformed skeletal mesh according to (1). We employ (2), (3) with  $L = 3$  to produce the deformed mesh. Finally mesh fairing is applied. Mesh evolution (4) eliminates folds and protrusions and mesh evolution (5) removes the self-intersections.



**Figure 13: Removing self-intersections of the deformed hand mesh from Fig. 12(d).** Left: various views at a zoomed part of Fig. 12(d). Right: corresponding views at the same parts of Fig. 12(e). The self-intersections are gone.

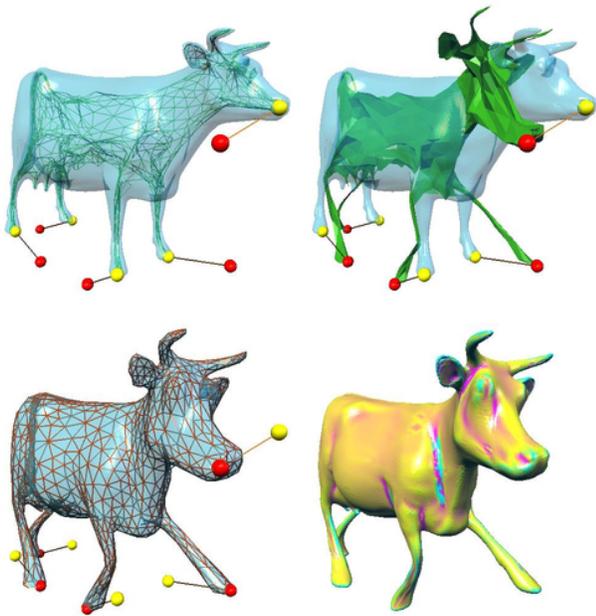
## 4. COMBINING WITH DISPLACED SUBDIVISION SURFACE REPRESENTATION

The most time consuming steps of our basic method presented in previous sections are those described in subsections 3.1 and 3.2. For example, for the hand model consisting of 16K triangles only, its deformation processes shown in Fig. 13 takes approximately one and a half minutes for eliminating self-intersections, about six seconds for removing folds and protrusions, and less than one second for all the other operations (a Java3D implementation on a 1.7GHz Pentium 4 computer was used). In order to perform the deformation process in a matter of few seconds we combine it with a displaced subdivision surface representation [18].

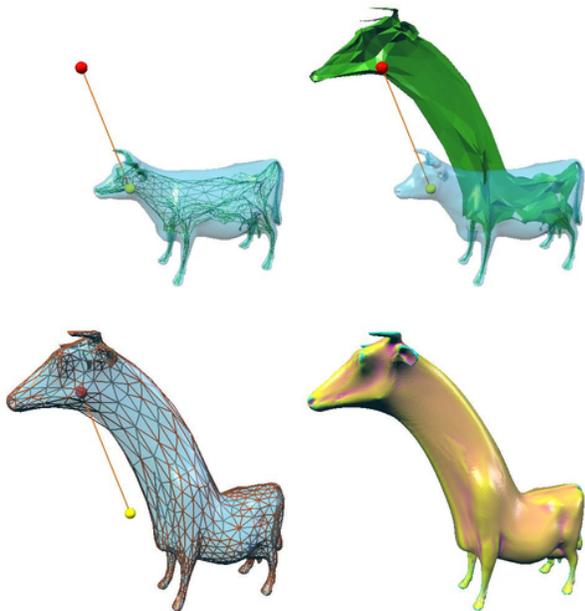
The displaced subdivision surface representation, DSS-rep, is a compact surface representation capturing small-scale details of an original surface as a scalar displacement field over a decimated and then subdivided surface.

Given a dense mesh, first we obtain a DSS representation of the mesh: a decimated mesh and a scalar displacement field. Then we build our skeleton-based representation of the decimated mesh. The decimated mesh has much fewer vertices than the original dense mesh and do not contain small-scale details. This leads to fast and robust extraction of the Voronoi-based skeleton for the decimated mesh. Moreover DSS-rep protects fine geometry features of the original mesh from being damaged by mesh evolutions (4) and (5). The mesh deformation process is now organized as follows: a free-form deformation is applied to the skeleton of the decimated mesh and implies a deformation of the decimated mesh. The deformed mesh is then subdivided and, finally, a deformation of the original dense mesh is obtained from the subdivided deformed mesh by adding the scalar displacement field.

To demonstrate how the above combination of DSS-rep and the skeleton-driven mesh deformation approach described in previous sections works we used the dragon, mannequin head, cow, and hand models. The models are remeshed (topological noise removal, decimation, subdivision) in order to improve their quality. See Figures 1, 14, 15, and 16 for the results. Coloring by the mean curvature is used for a quality evaluation of the deformed models.



**Figure 14: Skeleton-based deformations enriched by DSS: the cow model consists of 45K triangles while its skeletal mesh has 3K triangles only.**



**Figure 15: Another global deformation of the cow model.**

In these examples, the whole mesh deformation process takes only a few seconds without taking into account computing the DSS representation. In our current implementation, we compute the DSS representation without its most computationally expensive optimization step [18]. Besides DSS-rep has to be computed only once.



**Figure 16: Skeleton-based deformations enriched by DSS: the hand model has 38K triangles while its skeletal mesh consists of 2K triangles only.**

## 5. CONCLUSION

In this paper, we have developed a new approach to free-from skeleton-based mesh deformations. Using the skeleton as the base of our technique allows us to generate natural-looking large-scale mesh deformations. The main features of our approach are using Voronoi-based skeletal mesh, applying mesh evolutions for skeleton fairing, and combining skeleton-based mesh deformations with the DSS mesh representation approach. All this makes it possible to produce global mesh deformations of satisfactory quality.

To evolve a mesh according to (4) and (5) we employ a simple explicit finite-difference approximation. Better results can be achieved if a semi-implicit approximation similar to that proposed in [11] is used.

It turns out that a stability in extraction of the medial axis (skeleton) is more important for our method than an approximation accuracy. This is why we do not build our skeleton from the Voronoi poles introduced and effectively used for mesh reconstruction purposes in [2, 1]. Instead, following ideas of [12], we approximate the skeleton by linear combinations of Voronoi vertices.

As demonstrated, our skeleton-based approach works well for deforming objects composed of elongated parts. The approach has limitations in processing objects of spherical-like shapes because their skeletons are usually very complex. Another limitation of our method consists in deforming models with sharp edges and corners since the mesh evolutions (4) and (5) may destroy the mesh sharp features. Finally

## Acknowledgments

We would like to thank the anonymous reviewers of this paper for their valuable and constructive comments. Models are courtesy of the Stanford University (dragon), University of Washington (mannequin head), Kitware, Inc. (cow), and FarField Technology Ltd (hand).

## 6. REFERENCES

- [1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.
- [2] N. Amenta, M. Bern, and M. Kamvyselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH 1998*, pages 415–421, 1998.
- [3] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001.
- [4] E. V. Anoshkina, A. G. Belyaev, O. G. Okunev, and T. L. Kunii. Ridges and ravines: a singularity approach. *International Journal for Shape Modeling*, 1(1):1–11, 1994.
- [5] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22:469–483, 1996.
- [6] J. Bloomenthal. Skinning: Medial-based vertex deformation. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, pages 147–151, 2002.
- [7] J. Bloomenthal and C. Lim. Skeletal methods of shape manipulation. In *Proceedings of Shape Modeling International '99*, pages 44–47, 1999.
- [8] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Symposium on Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.
- [9] H. Blum. Biological shape and visual science. *J. Theor. Biology*, 38:205–287, 1973.
- [10] J.-D. Boissonnat, A. Cérézo, O. Devillers, J. Duquesne, and M. Yvinec. An algorithm for constructing the convex hull of a set of spheres in dimension  $d$ . *Comp. Geom. Theory Appl.*, 6:123–130, 1996.
- [11] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proceedings of ACM SIGGRAPH 99*, pages 317–324, 1999.
- [12] T. K. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. In *Proceedings of 7th ACM Symposium on Solid Modeling and Applications*, pages 356–366, Saarbrücken, Germany, June 2002.
- [13] P. Giblin and B. B. Kimia. A formal classification of 3D medial axis points and their local geometry. In *Computer Vision and Pattern Recognition (CVPR'00), Vol. I*, pages 566–575, 2000.
- [14] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proceedings of ACM SIGGRAPH 1999*, pages 325–334, 1999.
- [15] M. Hisada, A. G. Belyaev, and T. L. Kunii. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum*, 21(4):1–12, 2002.
- [16] L. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. In *Proceedings of EUROGRAPHICS 2000*, pages 249–260, 2000.
- [17] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH 1998*, pages 105–114, 1998.
- [18] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proceedings of ACM SIGGRAPH 2000*, pages 85–94, 2000.
- [19] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of ACM SIGGRAPH 2000*, pages 165–172, 2000.
- [20] K. Ohmori and T. L. Kunii. Shape modeling using homotopy. In *Proceedings of International Conference on Shape Modeling and Applications 2001*, pages 126–133, Genoa, Italy, May 2001.
- [21] K. Singh and E. Kokkevis. Skinning characters using surface-oriented free-form deformations. In *Proceedings of Graphics Interface 2000*, pages 35–42, 2000.
- [22] D. W. Storti, G. M. Turkiyyah, M. A. Ganter, C.T. Lim, and D. M. Stat. Skeleton-based modeling operations on solids. In *Proceedings of the 4th ACM Symposium on Solid Modeling and Applications*, pages 141–154, Atlanta, May 1997.
- [23] G. M. Turkiyyah, D. W. Storti, M. Ganter, H. Chen, and M. Vimawala. An accelerated triangulation method for computing the skeletons of free-form solid models. *Computer-Aided Design*, 29(1):5–19, 1997.
- [24] Z. J. Wood, M. Desbrun, P. Schneider, and D. Green. Semi-regular mesh extraction from volumes. In *Proceedings of IEEE Visualization 2000*, pages 275–282, 2000.
- [25] S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel. A simple approach to interactive free-form shape deformations. In *Proceedings of Pacific Graphics 2002*, pages 471–474, 2002.
- [26] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Proceedings of ACM SIGGRAPH 1997*, pages 259–268, 1997.