A Simple Approach to Interactive Free-Form Shape Deformations

Shin Yoshizawa Alexander G. Belyaev[*] Hans-Peter Seidel

Computer Graphics Group, Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany
Phone: [+49](681)9325-414 Fax: [+49](681)9325-499
E-mails: {shin,belyaev,hpseidel}@mpi-sb.mpg.de

Abstract

In this paper, we propose a set of free-form shape deformation techniques. The basic approach can be described as follows. Given a surface represented by a mesh and a control point, for every mesh vertex let us consider the difference between the control point and the vertex. The vertex is shifted by a displacement equal to the difference times a scale factor where the scale factor is given by a function depending nonlinearly on the difference. The function is bump-shaped and depends on a number of parameters. Varying the parameters leads to a rich palette of shape deformations. The proposed approach includes also shape deformations with multiple (real, auxiliary, and virtual) control points and constrained, directional, and anisotropic deformations. It allows a user to edit a given shape interactively and intuitively. No mesh connectivity information is used and, therefore, the proposed techniques can be applied directly to a shape given as a cloud of points.

Keywords: *constrained, directional, and anisotropic free-form shape deformations.*

1 Introduction

Since a pioneering work [8], developing fast, efficient, and intuitive methods for free-form shape deformations remains to be an area of active research (see, for example, recent works [9, 3, 6] and references therein). In this paper, we introduce a set of free-form shape deformation techniques.

One of the interesting features of our approach is that it does not use any mesh connectivity information and, therefore, can be used to create free-form deformations of shapes given as point datasets. Due to recent advances in 3D scanning technologies, the point-based shape representation becomes one of the 3D graphics mainstreams and calls for developing tools for processing shapes given as clouds of points (see, for example, [5]).

Although our basic techniques (Section 2) can be considered as a modification of well-known free-form shape deformation methods [10, 1, 9, 6], our approach seems to be simpler and more intuitive. The contribution of the paper includes developing new constrained, directional, and anisotropic free-form shape deformations achieved using real, virtual, and auxiliary control points (Section 3). Although, as it was noted above, our method can be applied directly to shapes given as point datasets, for visualization purposes we work with triangle meshes.

2 Basic Shape Deformations

The ideas behind the most of techniques described in this section are rather well-known and can be found in [10, 1, 9, 6], for example. The only idea with a virtual control point seems to be completely new. However the material of the section serves as a basis for more complex shape deformations techniques developed in Section 3.

Basic idea. Given a mesh and a control point $\mathbf{C}$, let us shift a mesh vertex $\mathbf{O}$ into its new position $\mathbf{P}$ defined by

$$\mathbf{P} = \mathbf{O} + \boldsymbol{D}(\mathbf{C}, \mathbf{O}) \tag{1}$$

with the displacement vector $\boldsymbol{D}$ given by

$$\boldsymbol{D}(\mathbf{C}, \mathbf{O}) = \frac{\gamma}{\sigma} W(\mathbf{C}, \mathbf{O})(\mathbf{O} - \mathbf{C}), \tag{2}$$

where
$$W(\mathbf{C}, \mathbf{O}) = \exp(-\frac{|\mathbf{C} - \mathbf{O}|^{\alpha}}{2\epsilon^2}), \tag{3}$$

$$\sigma = W(\mathbf{C}, \mathbf{O}_{min}), \tag{4}$$

and $\mathbf{O}_{\rm min}$ is the original vertex closest to the control point $\mathbf{C}$. Here γ, α, and ϵ are parameters. Of course, other bump-shaped functions can be used instead of (3) for example the local support Gaussian-like function which was used in [4].

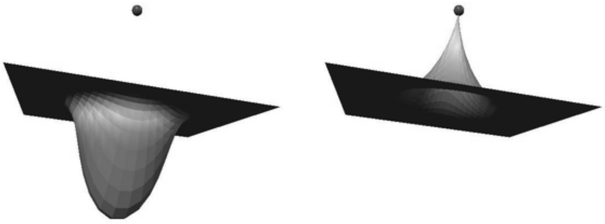


Figure 1. Deformation of a plane according to (1), (2), (3), (4) gives round-repelling when $\gamma = 1$ (left) and sharp-attracting when $\gamma = -1$ (right).

[*] On a leave from University of Aizu, Japan

The sign of the parameter γ in (2) defines whether the deformation attracting or repelling as demonstrated in Fig. 1. Note that $\gamma = -1$ and $\gamma = 1$ lead to different shape deformations: round-repelling when $\gamma = 1$ and sharp-attracting when $\gamma = -1$.

Virtual control point. In order to obtain a round-attracting and sharp-repelling deformations a concept of a virtual control point is introduced. The virtual control point $\mathbf{V}$ is obtained from $\mathbf{C}$ by the reflection with respect to $\mathbf{O}_{min}$, the mesh vertex closest to $\mathbf{C}$. Then the following displacement filed is used instead of (2)

$$\boldsymbol{D}(\mathbf{C}, \mathbf{O}) = \frac{\gamma}{\sigma} W(\mathbf{C}, \mathbf{O})(\mathbf{O} - \mathbf{V}). \tag{5}$$

Fig. 2 demonstrates round-attracting and sharp-repelling deformations created by using the virtual control point technique.

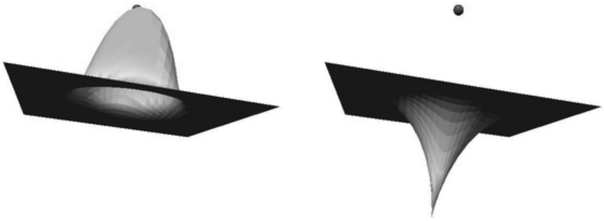


Figure 2. A virtual control point allows us to achieve a round-attracting (left) and sharp-repelling (right) effects.

Varying parameters. Varying parameters ϵ, α, and γ in (2), (3), (5) provides a user with a rich palette of interesting shape deformations: changing influence kernel size γ, region ϵ, and sharpness α, interactively.

Multiple control points A straightforward extension of the techniques described above to the case of multiple control points is simple. For example, instead of (1) the following updating rule can be used

$$\mathbf{P} = \mathbf{O} + \sum \boldsymbol{D}(\mathbf{C}_k, \mathbf{O}), \tag{6}$$

where the sum is taken over all the control points $\mathbf{C}_k$. The parameters (α_k, ϵ_k and γ_k) can be defined separately for each control point $\mathbf{C}_k$. The deformations shown in the middle and right images of Fig. 8 were obtained using (6).

However if two or more control points are close to each other the following procedure blending the effects of the control points is preferable:

$$\mathbf{P} = \mathbf{O} + \frac{\sum \boldsymbol{D}(\mathbf{C}_k, \mathbf{O}) |\boldsymbol{D}(\mathbf{C}_k, \mathbf{O})|^{\beta}}{\sum |\boldsymbol{D}(\mathbf{C}_k, \mathbf{O})|^{\beta}}, \tag{7}$$

A similar blending procedure was used in [9]. Here β is a user-defined parameter which adjusts the blending magnitude.

3 Advanced Deformation Techniques

Constrained deformations. The technique introduced in the previous section may produce undesirable deformations of a surface if the control point is close to two different parts of the surface, as demonstrated in the left image of Fig. 3. One possible remedy for the problem consists of using $W(\mathbf{O}_{\min}, \mathbf{O})$ instead of $W(\mathbf{C}, \mathbf{O})$ in (2) and set $\sigma = 1$ instead of (4). Here $\mathbf{O}_{\min}$ is the mesh vertex closest to the control point $\mathbf{C}$. This shifting the center the weight function W from $\mathbf{C}$ to $\mathbf{O}_{\min}$ eliminates unintended deformations shown in the right image of Fig. 3. However the position of $\mathbf{O}_{\min}$ may bifurcate when the position of the control point changes slightly.

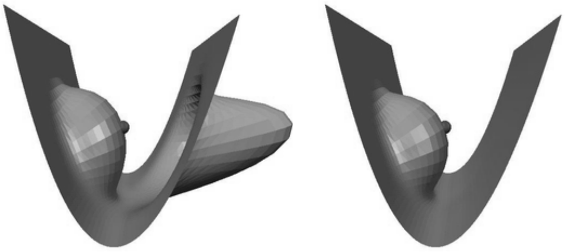
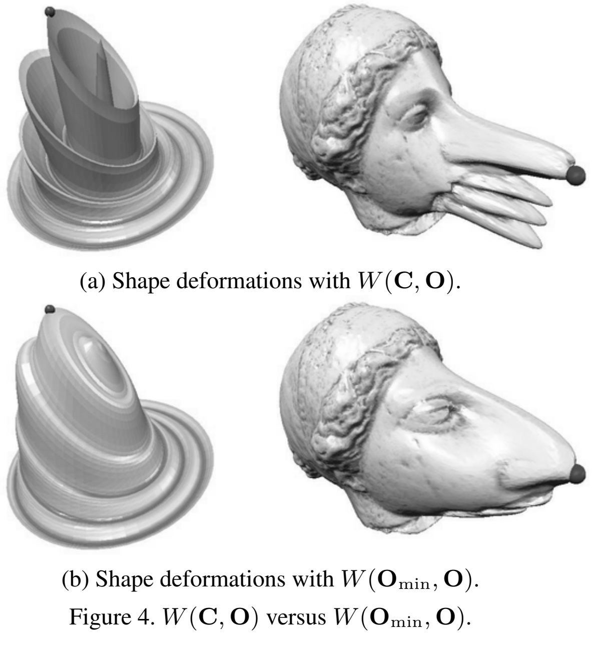


Figure 3. Left: unintended deformations of a surface may occur if control point $\mathbf{C}$ is close to two different parts of the surface. Right: round-repelling with $W(\mathbf{O}_{\min}, \mathbf{O})$ instead of $W(\mathbf{C}, \mathbf{O})$ in (2).

In spite of the problem with bifurcations, deformations with $W(\mathbf{O}_{\min}, \mathbf{O})$ instead of $W(\mathbf{C}, \mathbf{O})$ in (2) enrich the palette of free-form shape deformations, see Fig. 4. In general, the deformations with $W(\mathbf{C}, \mathbf{O})$ are more sensitive to shape features than those with $W(\mathbf{O}_{\min}, \mathbf{O})$.



(a) Shape deformations with $W(\mathbf{C}, \mathbf{O})$.

(b) Shape deformations with $W(\mathbf{O}_{\min}, \mathbf{O})$.

Figure 4. $W(\mathbf{C}, \mathbf{O})$ versus $W(\mathbf{O}_{\min}, \mathbf{O})$.

Directional deformations. We can easily avoid the problem with bifurcations if instead of the mesh vertex $\mathbf{O}_{\min}$ closest to the control point $\mathbf{C}$ a reference point $\mathbf{R}$ on the mesh is chosen. The virtual control point $\mathbf{V}$ is defined as the point symmetric to $\mathbf{C}$ with respect to $\mathbf{R}$:

$$\mathbf{V} = 2\mathbf{R} - \mathbf{C}.$$

We use $W(\mathbf{R}, \mathbf{O})$ instead of $W(\mathbf{C}, \mathbf{O})$ in (2) for the directional deformations. Fig. 5 demonstrates directional deformations of a plane.

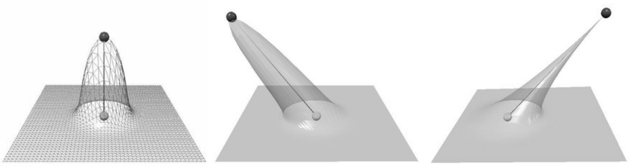


Figure 5. Directional deformations.

Anisotropic deformations. Given a positive definite matrix M, an anisotropic squared norm of a vector $\boldsymbol{x}$ is given by $\boldsymbol{x}^t \cdot M \cdot \boldsymbol{x}$. Using in (3) the anisotropic distance squared $(\mathbf{C} - \mathbf{O})^t \cdot M \cdot (\mathbf{C} - \mathbf{O})$ between the control point $\mathbf{C}$ and a mesh vertex $\mathbf{O}$ instead of the standard Euclidean squared distance $(\mathbf{C} - \mathbf{O})^t \cdot (\mathbf{C} - \mathbf{O})$ allows us to create a new class of free-form deformations.

In order to implement anisotropic deformations with a user-controlled matrix M an additional control point $\mathbf{Q}$ is added such that it always belongs to the plane orthogonal to $\mathbf{CO}_{\min}$. Here $\mathbf{O}_{\min}$ is the mesh vertex closest to $\mathbf{C}$.

Three points $\mathbf{C}$, $\mathbf{O}_{\min}$, and $\mathbf{Q}$ are used to define a basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ as follows:

$$\boldsymbol{e}_1 = \frac{\mathbf{Q} - \mathbf{C}}{|\mathbf{Q} - \mathbf{C}|}, \quad \boldsymbol{e}_3 = \frac{\mathbf{C} - \mathbf{O}_{\min}}{|\mathbf{C} - \mathbf{O}_{\min}|}, \quad \boldsymbol{e}_2 = \boldsymbol{e}_1 \times \boldsymbol{e}_3.$$

See the left image of Fig. 6.

Let $B = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$ be the coordinate transform matrix associated with the basis $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ and

$$A = \begin{bmatrix} g(|\mathbf{Q} - \mathbf{C}|) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $g(x)$ is a smooth function defined for $x > 0$. The matrix M is defined by $M = B^t A\, B$. In our current implementation, we use $g(x) = \frac{1}{10x}$.

The anisotropic deformations can be easily combined with the directional shape deformations if a user-specified mesh vertex $\mathbf{R}$ is used instead of $\mathbf{O}_{\min}$. The closest point $\mathbf{O}_{\min}$ remains to be used for computing the basis vector $\boldsymbol{e}_3$.

Fig. 7 demonstrates how a directional anisotropic deformation modifies the head of a lion-dog statue[1].

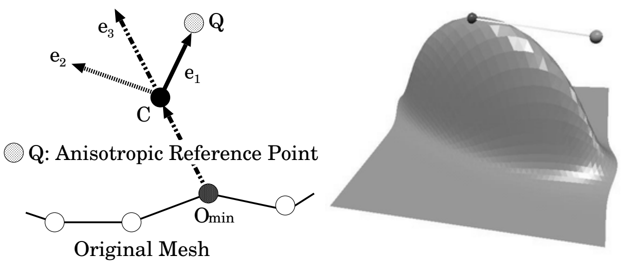


Figure 6. Anisotropic deformations are defined via an additional control point $\mathbf{Q}$.

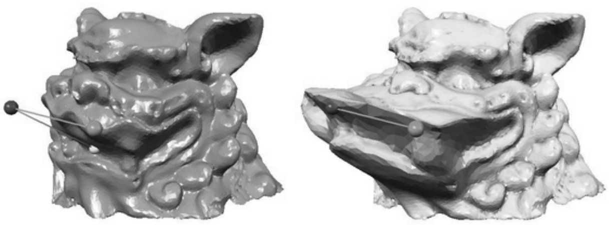


Figure 7. A directional anisotropic deformation is used to open the “lion-dog” mouth.

The anisotropic deformations considered in this section are not capable to create shape deformations with sharp edges. Sharp edges can be produced if several control points are used, as seen in Fig. 8.

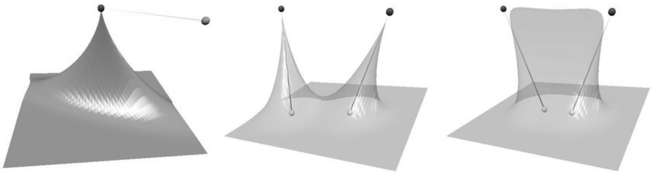


Figure 8. Left: anisotropic sharp-attracting deformation. Middle and right: anisotropic deformations can be created via superposition of several isotropic deformations.

4 Conclusion

We hope that the set of tools presented in the paper will provide a modeler with a rich palette (fast, efficient, and intuitive) of free-from shape deformations, because the linear computational cost is achieved for each control point to describe various interesting deformation effects. The techniques described in this paper are implemented in a Java 3D toolkit developed by the first author [11]. Fig. 10 shows the main GUI window of the toolkit. Our experiments with the toolkit assure us that the shape deformations described in the paper can be used for shape morphing, see Fig. 9, changing facial expressions, see Fig. 11, and many other shape editing tasks.

Given a multiresolution mesh [7], the shape deformations we proposed can be applied to various levels of detail. The deformations can be also applied to control lattices [8] and combined with the wires technique [9]. Implementing a dynamic mesh connectivity [2] and shape deforma-

[1] The mythical Buddhist ”lion-dog” combines the power of the lion and the loyalty of the dog.

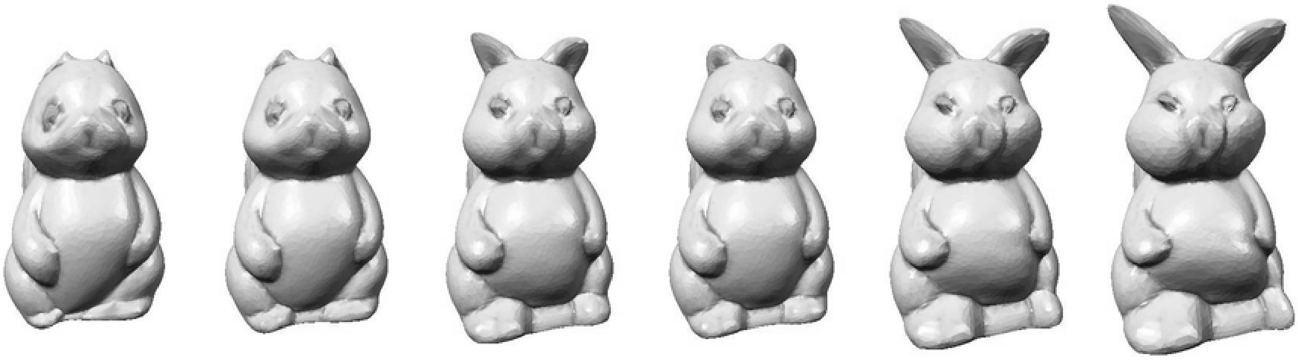


Figure 9. Morphing a squirrel (left) into a piglet (right).

tions acting via modifying the field of shape normals and/or shape curvatures constitute other promising areas for future research.

Acknowledgments

The mannequin head and Venus head models are courtesy of University of Washington and Cyberware Inc., respectively. We are grateful to Yutaka Ohtake for the lion-dog, squirrel, and Moai statue models.

References

[1] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics*, 13(2):137–155, 1994.

[2] L. Kobbelt, T. Bareuther, and H. P. Seidel. Multi-resolution shape deformations for meshes with dynamic vertex connectivity. *Computer Graphics Forum (Eurographics 2000 issue)*, 19(3):C249–C260, 2000.

[3] L. Markosian, J. M. Cohen, T. Crulli, and J. Hughes. Skin: A constructive approach to modeling free-form shapes. In *Computer Graphics (Proceedings of SIGGRAPH 1999)*, pages 393–400, 1999.

[4] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation and Depth, Lecture Notes in Computer Science, Vol. 662*. Springer, 1993.

[5] M. Pauly and M. Gross. Spectral processing of point-sampled geometry. In *Computer Graphics (Proceedings of SIGGRAPH 2001)*, pages 379–386, 2001.

[6] R. Raffin, M. Neveu, and F. Jaar. Curvilinear displacement of free-form-based deformation. *The Visual Computer*, 16(1):38–46, 2000.

[7] P. Schröder and D. Zorin. Subdivision for modeling and animation. Siggraph 2000 course notes 23, 2000.

[8] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Computer Graphics (Proceedings of SIGGRAPH 1986)*, pages 151–160, 1986.

[9] K. Singh and E. Fiume. Wires: A geometric deformation technique. In *Computer Graphics (Proceedings of SIGGRAPH 1998)*, pages 405–414, 1998.

[10] G. Wyvill and D. McRobie. Local and Global Control of Cao En Surfaces. In *Proceedings of Computer Graphics International 1993*, pages 216–227, 1993.

[11] S. Yoshizawa. A Java3D Toolkit for Interactive Free-Form Shape Deformations. Java3D application, http://www.mpi-sb.mpg.de/∼shin/Research/DeformMesh/DeformMesh.html, 2001.

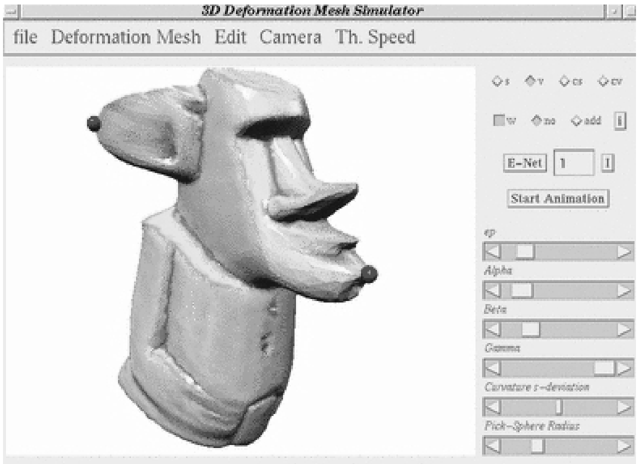


Figure 10. GUI of Yoshizawa's toolkit for free-form shape deformations. A deformed Moai statue model is shown.

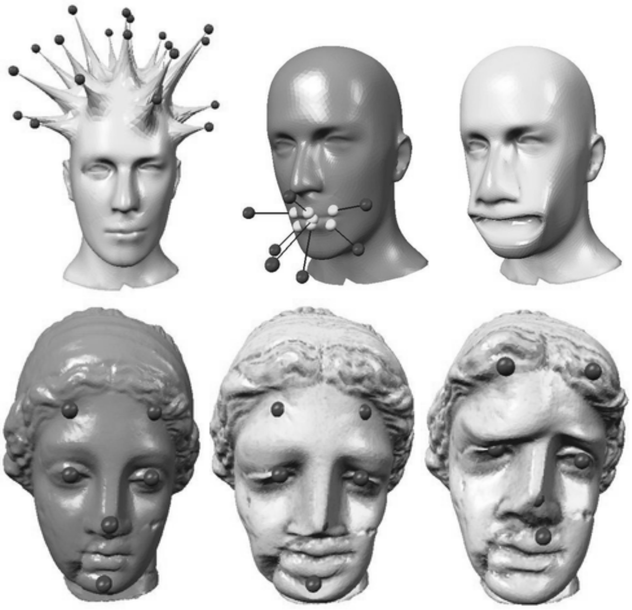


Figure 11. Deformations of complex objects.