Swirling-Sweepers: Constant Volume Modeling

Alexis Angelidis, Marie-Paule Cani, Geoff Wyvill, Scott King

Swirling-Sweepers: Constant-Volume Modeling

Alexis Angelidis*
University of Otago

Marie-Paule Cani†
Laboratoire GRAVIR‡

Geoff Wyvill§
University of Otago

Scott King¶
University of Otago

Abstract

Swirling-sweepers is a new method for modeling shapes while preserving volume. The artist describes a deformation by dragging a point along a path. The method is independent of the geometric representation of the shape. It preserves volume and avoids self-intersections, both local and global. It is capable of unlimited stretching and the deformation can be constrained to affect only a part of the model.

We argue that all of these properties are necessary for interactive modeling if the user is to have the impression that he or she is shaping a real material. Our method is the first to implement all five.

1. Introduction

In a virtual modeling context, there is no material: no wax, clay, wood or marble. A challenge for computer graphics is to provide a virtual tool that convinces the artist that there is material. To perfect this illusion, the shape must behave in accordance with a suitable modeling metaphor.

Volume is one of the most important factors influencing the manner in which an artist models with real materials. A virtual tool preserving volume is needed to help the artist believe he is interacting with material. Also, modeling by preserving the available amount of material will produce a shape with style, that other virtual modeling methods can only achieve with more effort.

* email: alexis@cs.otago.ac.nz
† email: Marie-Paule.Cani@imag.fr
‡ GRAVIR is a joint lab of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.
§ email: geoff@cs.otago.ac.nz
¶ email: scott@cs.otago.ac.nz

1.1. Previous volume-control models

Volume preservation has been recognized for a long time in animation, as a desirable property for the animation of believable animal and human characters [13]. [9] use constrained optimization methods for objects discretized into lattices. [5] use controllers for maintaining the implicit surface that coats a set of particles to a constant volume during deformation. [7] achieve incompressibility in water simulation by maintaining a divergence free velocity field, thanks to the Poisson equation. [12] rely on finite volume methods to simulate quasi-incompressible materials such as muscular tissue.

Volume preservation has also been considered as a very useful constraint for the intuitive modeling of shapes. [10] propose an optimization method to adjust the control points of the popular free form deformations (FFD) [11], but it works only for tensor-solids. [8] also adjust FFD control points, but their method does not allow local editing. [3] propose a volume preserving space deformation based on a model called DOGME. The deformation does not have a local support, and requires the computation of the shape's volume. [4] preserve only a volume between the surface and a base surface. [6] introduce mass-preserving local and global deformations for shapes represented by a mass-density field sampled in a grid.

The limitation of existing methods is either that they only apply to a specific type of geometric representation, or they only apply to shapes whose volume can be computed.

1.2. Overview

This paper presents swirling-sweepers, a new method dedicated to modeling shapes while preserving the shape's volume. Our technique belongs to space deformations, and is therefore applicable to a wide range

of geometric representations, including all of the popular parametric surfaces.

It is the first method that preserves volume, has a local support, prevents local and global self-intersection of the surface and does not require any volume computation. Most importantly, using the method is simple: the artist only has to provide the trajectory of a point, for instance with a mouse.

In Section 2 we summarize the principle of the space deformations called sweepers. Then we present in Section 3 our new method for modeling by constant volume deformation.

2. Principle of Sweepers

We briefly review the elements required for understanding the space deformations called *sweepers* [2].

Space deformation provides a formalism to specify any modeling operation by successively deforming the space in which an initial shape, $S(t_0)$, is embedded. A deformed shape is given by the *modeling equation*[1]:

$$S(t_n) = \left\{ \bigOmega_{i=0}^{n-1} f_{t_i \mapsto t_{i+1}}(p) \mid p \in S(t_0) \right\} \tag{1}$$

where $f_{t_i \mapsto t_{i+1}} : \mathbb{R}^3 \to \mathbb{R}^3$ are n space deformations, deforming a point, p, of shape $S(t_i)$ into a point of shape $S(t_{i+1})$.

Informally, a *sweeper* is a geometric tool together with a motion path. This tool defines an influence function. The basic idea is that the tool is placed somewhere in the region of a shape to be deformed and moved along the path. The motion drags a part of space defined by the influence function, in a manner that prevents the shape from self-intersecting.

More formally, the deformation is defined by a scalar function, $\phi_t : \mathbb{R}^3 \mapsto [0,1]$, that varies over time, t. This field is defined by composing the distance to the tool, d_t, with an influence function, μ

$$\phi_t = \mu \circ d_t \tag{2}$$

Any smooth decreasing function of finite support can be used for μ. We use a C^2 continuous piecewise polynomial, in which λ defines the *radius* of the influence

$$\mu_\lambda(d_t) = \begin{cases} 0 & \text{if } \lambda \le d_t \\ 1 + (\frac{d_t}{\lambda})^3 (\frac{d_t}{\lambda}(15 - 6\frac{d_t}{\lambda}) - 10) & \text{if } \lambda > d_t \end{cases} \tag{3}$$

1 $\bigOmega_{i=0}^{n-1} f_{t_i \mapsto t_{i+1}}(p)$ expresses the finite repeated composition of functions $f_{t_{n-1} \mapsto t_n} \circ \cdots \circ f_{t_0 \mapsto t_1}(p)$

A deformation is defined by transforming the tool's position, size and orientation, given by the matrix M_{t_i} into the next configuration, given by the matrix $M_{t_{i+1}}$. Let us denote $M_i = M_{t_{i+1}} M_{t_i}^{-1}$ the transformation matrix from the previous to the new configuration. A naive deformation of a point p with a single tool would be

$$f_{t_i \mapsto t_{i+1}}(p) = (\phi_{t_i}(p) \odot M_i)\, p \tag{4}$$

where the matrix operator $\odot$ is defined as $\alpha \odot M = \exp(\alpha \log M)$. In Section 3.3, we give the closed-form for computing exp and log. We refer the reader to [1] for a more detailed overview. Loosely speaking, the operation $\odot$ is the equivalent of multiplying a matrix by a scalar. It raises a transformation to a non-rational exponent. However, Equation 4 does not prevent the surface from self-intersecting. By decomposing the deformation into s sub-functions [2], self-intersections are avoided

$$f_{t_i \mapsto t_{i+1}} = \bigOmega_{j=0}^{s-1} f_{\tau_j \mapsto \tau_{j+1}}(p) \tag{5}$$

$$\text{where } \; f_{\tau_j \mapsto \tau_{j+1}}(p) = \left(\frac{\phi_{\tau_j}(p)}{s} \odot M_i \right) p$$

For a tool in a bounding volume V_b, the number of steps is

$$s = \max(1, \; \lceil -\min(\delta\mu_\lambda / \delta d) \max_{p \in V_b} ||\log M_i p|| \rceil) \tag{6}$$

3. Constant Volume deformation

We introduce here swirling-sweepers as a method for deforming shapes while preserving their volume. Swirling-sweepers are a particular case of sweepers that use only point tools.

3.1. A basic deformation

We define a particular case of sweeper, a *swirl*, by using a point tool, c, together with a rotation of angle θ around an axis $\vec{v}$ (see Figure 1). A scalar function, ϕ, and a deformation are defined as before (see Equations 2 and 4). Informally, a swirl twists space locally around axis $\vec{v}$ without compression or dilation. We prove in Appendix A that a swirl preserves volume.

3.2. Combining for complexity

Many deformations of the above kind can be naively combined to create a more complex deformation

$$f(p) = \left(\bigoplus_{i=0}^{n-1} (\phi_i(p) \odot M_i) \right) p \tag{7}$$

where $\oplus$ is a commutative addition of transformations defined as $M \oplus N = \exp(\log M + \log N)$ [1]. We provide a convenient way for the artist to input n rotations, by specification of a single translation $\vec{t}$. Let us consider n points, c_i, on the circle of center h, and radius r lying in a plane perpendicular to $\vec{t}$. To these points correspond n consistently-oriented unit tangent vectors $\vec{v}_i$ (see Figure 2). Each pair, $(c_i, \vec{v}_i)$, together with an angle, θ_i, define a rotation. Along with radii of influence $\lambda_i = 2r$, we can define n swirls. The radius of the circle, r, is left to the user to choose. The following value for θ_i will transform h exactly into $h + \vec{t}$ (see Appendix B).

$$\theta_i = \frac{2||\vec{t}\,||}{nr} \tag{8}$$

With this information, the deformation of Equation 7 is now a tool capable of transforming a point into a desired target. We show in Figure 2 the effect of the tool for different values of n; in practice, we use 8 swirls.

Preserving coherency and volume If the magnitude of the input vector $\vec{t}$ is too large, the deformation of Equation 7 will produce a self-intersecting surface, and will not preserve volume. The reason for self-intersection is explained with details in [2]. The volume is not preserved because the blending operator, $\oplus$, blends the transformation matrices, and not the deformations. To correct this, it is necessary to subdivide $\vec{t}$ into smaller vectors. The number of steps must be proportional to the speed and inversely proportional to the size of the tool. We use

$$s = \max(1, \lceil 4||\vec{t}\,||/r \rceil) \tag{9}$$

As the circle sweeps space, it defines a cylinder. Thus the swirling-sweeper is made of $n \cdot s$ basic deformations. Figure 3 illustrates this decomposition applied to a shape.

3.3. Swirling-sweepers algorithm

We summarize here the swirling-sweepers algorithm:

Input point, h, translation, $\vec{t}$, and radius, r
Compute the number of required steps, s
Compute the angle of each step, $\theta_i = \frac{2||\vec{t}||}{nrs}$
for each step j from 0 to $s-1$ **do**
  for each point p in the tool's bounding box **do**
    $M = 0$
    for each swirl i from 0 to $n-1$ **do**
      $M \mathrel{+}= \mu_{2r}(||p - c_{ij}||) \log M_{i,j}$
    end for
    $p = (\exp M)p$
  end for
end for

The point c_{ij} denotes the center of the i^{th} swirl of the j^{th} ring of swirls. For efficiency, a table of the basic-swirl centers, c_{ij}, and a table of the rotation matrices, $\log M_{i,j}$, are precomputed. We have a closed-form for the logarithm of the involved matrix, given in equations 10 and 11, saving an otherwise expensive numerical approximation.

$$\begin{aligned} \vec{n} &= \theta_i \vec{v}_i \\ \vec{m} &= c_{i,j} \times \vec{n} \end{aligned} \tag{10}$$

$$\log M_{i,j} = \begin{pmatrix} 0 & -n_z & n_y & m_x \\ n_z & 0 & -n_x & m_y \\ -n_y & n_x & 0 & m_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{11}$$

Note that for the sake of efficiency, we handle these matrices as mere pairs of vectors, $(\vec{n}, \vec{m})$. Once M is computed, we use a closed-form for computing $\exp M$. Since the matrix M is a weighted sum of matrices $\log M_{i,j}$, the matrix M is of the form of Equation 11, and can be represented with a pair, $(\vec{n}_M, \vec{m}_M)$. If $\vec{n}_M = 0$, then $\exp M$ is a translation of vector $\vec{m}_M$. Else, if the dot product $\vec{m}_M \cdot \vec{n}_M = 0$, then $\exp M$ is a rotation of center $\vec{c}$, angle θ axis $\vec{v}$, as given by Equation 12.

$$\begin{aligned} \vec{c} &= \begin{cases} (1, \frac{n_y - m_z}{n_x}, \frac{n_z + m_y}{n_x}) & \text{if } |n_y| \le |n_x| \\ & \text{and } |n_z| \le |n_x| \\ (\frac{n_x + m_z}{n_y}, 1, \frac{n_z - m_x}{n_y}) & \text{if } |n_z| \le |n_y| \\ (\frac{n_x - m_y}{n_z}, \frac{n_y + m_x}{n_z}, 1) & \text{otherwise} \end{cases} \\ \theta &= ||\vec{n}_M|| \\ \vec{v} &= \vec{n}_M/\theta \end{aligned} \tag{12}$$

Finally, in the remaining cases, we denote $l = \|\vec{n}_M\|$, and we use Equation 13. See Appendix C for efficiency.

$$\exp M = I + M + \tfrac{1-\cos l}{l^2} M^2 + \tfrac{l - \sin l}{l^3} M^3 \tag{13}$$

Symmetrical objects can be easily modeled by introducing a plane of symmetry about which the tool is reflected (see Figure 5).

4. Results

We have implemented swirling-sweepers in C++ using OpenGL®, on a Pentium® 2400Mhz with 1GB of RAM. This implementation works in real-time. The computational time is a function of the magnitude of the input vector, because this determines the number of substeps. Small vectors will produce extremely fast deformations. In order to preserve the sampling of the de-

formed surface, we use the mesh update algorithm proposed in [2], adapted for sweeping space deformations. Results are shown in Figure 5.

Limitations In our implementation, the tool must be of significant size compared to the density of the mesh. In Figure 5, we compare the shapes' volume with unit spheres on the right. The shapes volumes are respectively 101.422%, 99.993%, 101.158% and 103.633% of the initial sphere. This error is the result of accumulating smaller errors from each deformation. For instance 80 swirling-sweepers have been used to model the alien. The small errors are due to the finite number of steps, and to our choice of shape representation.

5. Conclusions and future work

We have presented swirling-sweepers, a new volume-preserving space deformation that uses the sweepers formulation: a combination of matrices raised to powers of scalar functions. Combined with the original sweepers, the volume of a shape can be increased, preserved or decreased. We believe there are many more useful sweeper operations yet to be discovered, for instance more complex volume-preserving tools, topology changing tools, surface area preserving tools, or surface smoothing tools. We also believe our technique is adaptable to volume preserving animation.

6. Acknowledgments

Many thanks to Sui-Ling Ming-Wong for carefully proof-reading this paper.

A. Constant volume basic swirl

Let us prove that a basic swirl preserves the volume. We will show that the determinant of the Jacobian, $\det J$, is equal to 1 everywhere. A rotation of angle 2θ around $\vec{n}$ and its powers in ϕ can be modeled with a quaternion:

$$q^\phi = (\cos(\phi\theta), \sin(\phi\theta)\vec{n}) \tag{14}$$

Hence the swirl deformation of a point $p = (x, y, z)^T$ is

$$f(p) = q^\phi p \overline{q^\phi} \tag{15}$$

We can assume without loss of generality that the rotation is centered at the origin. To express the Jacobian, we need the three partial derivatives of f. Let us denote $\vec{\mathrm{x}}$, $\vec{\mathrm{y}}$ and $\vec{\mathrm{z}}$ the vectors $(1,0,0)^T$, $(0,1,0)^\top$ and $(0,0,1)^\top$. The first partial derivative of f is

$$\frac{\delta f(p)}{\delta x} = \frac{\delta q^\phi}{\delta x} p \overline{q^\phi} + q^\phi \vec{\mathrm{x}} \overline{q^\phi} + q^\phi p \frac{\delta \overline{q^\phi}}{\delta x} \tag{16}$$

The reader can verify the quaternion equality:

$$\frac{\delta q^\phi}{\delta x} = -\theta \frac{\delta \phi}{\delta x}(\sin(\phi\theta), -\cos(\phi\theta)\vec{n}) \tag{17}$$

Using 17, the leftmost term of the right side of equation 16 is a quaternion:

$$\begin{aligned} & \tfrac{\delta q^\phi}{\delta x} p \overline{q^\phi} \\ = \;& \theta \tfrac{\delta \phi}{\delta x}(-\sin(\phi\theta), \cos(\phi\theta)\vec{n}) * p * \overline{q^\phi} \\ = \;& \theta \tfrac{\delta \phi}{\delta x}(-\cos(\phi\theta)\vec{n} \cdot p, -\sin(\phi\theta)p + \cos(\phi\theta)\vec{n} \times p) * \overline{q^\phi} \\ = \;& \theta \tfrac{\delta \phi}{\delta x}(-\vec{n} \cdot p, \cos(2\phi\theta)(\vec{n} \times p) - \sin(2\phi\theta)(\vec{n} \times p) \times \vec{n}) \end{aligned}$$

The reader can verify similarly that the rightmost term of equation 16 is also a quaternion:

$$\tfrac{\delta q^\phi}{\delta x} p \overline{q^\phi} = \theta \tfrac{\delta \phi}{\delta x}(\vec{n} \cdot p, \cos(2\phi\theta)(\vec{n} \times p) - \sin(2\phi\theta)(\vec{n} \times p) \times \vec{n})$$

The partial derivative of f in x is a vector:

$$\tfrac{\delta f(p)}{\delta x} = 2\theta \tfrac{\delta \phi}{\delta x}(\cos(2\phi\theta)(\vec{n} \times p) - \sin(2\phi\theta)(\vec{n} \times p) \times \vec{n}) + q^\phi \vec{\mathrm{x}} \overline{q^\phi}$$

Let us introduce $\vec{q}_x$ and $\vec{u}$ for the sake of simplicity:

$$\begin{aligned} \vec{q}_x &= q^\phi \vec{\mathrm{x}} \overline{q^\phi} \\ \vec{u} &= 2\theta(\cos(2\phi\theta)(\vec{n} \times p) - \sin(2\phi\theta)(\vec{n} \times p) \times \vec{n}) \end{aligned}$$

The partial derivative in x shortens to:

$$\frac{\delta f(p)}{\delta x} = \frac{\delta \phi}{\delta x}\vec{u} + \vec{q}_x \tag{18}$$

The two other partial derivatives of f are obtained by substituting x for y or z. Since a rotation preserves lengths and angles, we can write:

$$\vec{q}_x = \vec{q}_y \times \vec{q}_z$$

Let us develop the determinant of the Jacobian:

$$\begin{aligned}\det J &= \frac{\delta f}{\delta x} \cdot (\frac{\delta f}{\delta y} \times \frac{\delta f}{\delta x}) \\ &= 1 + \vec{u} \cdot (\frac{\delta f}{\delta x} q_x + \frac{\delta f}{\delta y} q_y + \frac{\delta f}{\delta z} q_z)\end{aligned}$$

We can assume without loss of generality that $\vec{n} = \vec{\mathrm{x}}$. This provides expressions for the rotated canonic set:

$$\begin{aligned}\vec{q_x} &= \vec{\mathrm{x}} \\ \vec{q_y} &= \cos(2\phi\theta)\vec{\mathrm{y}} + \sin(2\phi\theta)\vec{\mathrm{z}} \\ \vec{q_z} &= \cos(2\phi\theta)\vec{\mathrm{z}} - \sin(2\phi\theta)\vec{\mathrm{y}}\end{aligned}$$

This assumption also provides a simple expression for the double cross product:

$$(\vec{n} \times p) \times \vec{n} = y\vec{\mathrm{y}} + z\vec{\mathrm{z}}$$

We will now use the fact that the tool is spherical. We model the field function ϕ as a function of the distance to the origin, $d(p)$. The field can be partially derived:

$$\frac{\delta\phi(d(p))}{\delta x} = \frac{\delta\phi}{\delta d}\frac{\delta d(p)}{\delta x} = \frac{\delta\phi}{\delta d}\frac{x}{d(p)}$$

With this, the determinant of the Jacobian becomes:

$$\begin{aligned}& \det J \\ = \;& 1 + \tfrac{\delta\phi}{\delta d}\tfrac{2\theta}{d(p)} \\ & (\cos()\vec{\mathrm{x}} \times p - \sin()(y\vec{\mathrm{y}} + z\vec{\mathrm{z}}))\cdot \\ & (x\vec{\mathrm{x}} + (y\cos() - z\sin())\vec{\mathrm{y}} + (y\sin() + z\cos())\vec{\mathrm{z}}) \\ = \;& 1\end{aligned}$$

Thus $\det J$ is equal to 1 everywhere. Therefore the deformation stretches space with no expansion nor compression.

B. Swirl angle

The image of a point p in the center of a circle of swirls is given by Equation 7. Since the point is at the center, one can substitute ϕ_i for $1/2$

$$f(p) = \bigoplus_{i=0}^{n-1}(\tfrac{1}{2} \odot M_i)p \tag{19}$$

The speed of this deformation at p is given by the logarithm

$$\vec{v} = \textstyle\sum_{i=0}^{n-1} \frac{1}{2}\log(M_i)p \tag{20}$$

Since M_i is a rotation matrix, this simplifies (see Equation 11).

$$\vec{v} = \tfrac{\theta}{2}\textstyle\sum_{i=0}^{n-1}(\vec{v_i} \times (p - c_i)) \tag{21}$$

By taking the norm:

$$||\vec{v}|| = \tfrac{\theta}{2}\textstyle\sum_{i=0}^{n-1}||p - c_i|| \tag{22}$$

Since the centers are equidistant to p

$$||\vec{v}|| = \tfrac{\theta}{2}nr \tag{23}$$

Therefore the angle is

$$\theta = \tfrac{2||\vec{v}||}{nr} \tag{24}$$

C. Exponential

Applying the exponential of the matrix to a point does not require to compute the exponential of the matrix explicitely. Let us define the matrix M with a pair of vectors, $(\vec{n}, \vec{m})$.

$$\begin{aligned}\exp(M)p &= p + (\vec{m} + \vec{n} \times p)b + (\tfrac{\vec{n}\times\vec{m}}{l^2} - p)a \\ &\quad + \vec{n}((\vec{n} * p)a + (\vec{n} * \vec{m})(1 - b))\tfrac{1}{l^2} \\ \text{where } l &= \|\vec{n}\| \\ a &= 1 - \cos(l) \\ b &= \tfrac{\sin(l)}{l}\end{aligned} \tag{25}$$

References

[1] M. Alexa. Linear combination of transformations. In *Proceedings of SIGGRAPH'02*, Computer Graphics Proceedings, Annual Conference Series, pages 380–387. ACM, July 2002.

[2] A. Angelidis, G. Wyvill, and M.-P. Cani. Sweepers: Swept user-defined tools for modeling by deformation. In *Proceedings of Shape Modeling and Applications*. IEEE, June 2004.

[3] F. Aubert and D. Bechmann. Volume-preserving space deformation. *Computers & Graphics*, 21(5):625–639, 1997.

[4] M. Botsch and L. Kobbelt. Multiresolution surface representation based on displacement volumes. In *Proceedings of Eurographics*, pages 483–491. Eurographics, September 2003.

[5] M. Desbrun and M.-P. Cani. Animating soft substances with implicit surfaces. In *proceedings of SIGGRAPH*, pages 287–290. ACM, August 1995.

[6] G. Dewaele and M.-P. Cani. Interactive global and local deformations for virtual clay. In *Proceedings of Pacific Graphics*, pages 131–140. IEEE, October 2003.

[7] N. Foster and R. Fedkiw. Practical animation of liquids. In *proceedings of SIGGRAPH*, pages 23–29. ACM, August 2001.

[8] G. Hirota, R. Maheshwari, and M. C. Lin. Fast volume-preserving free form deformation using multi-level optimization. In *Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 234–245. ACM, June 1999.

[9] J. C. Platt and A. H. Barr. Constraint methods for flexible models. In *Proceedings of SIGGRAPH*, Computer Graphics Proceedings, Annual Conference Series, pages 279–288. ACM, August 1988.

[10] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preseving free-form solids. In *Proceedings of Solid Modeling*, pages 361–372. ACM, May 1995.

[11] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH*, Computer Graphics Proceedings, Annual Conference Series, pages 151–160. ACM, August 1986.

[12] J. Teran, S. Blemker, V. N. T. Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Eurographics/SIGGRAPH Symposium on Computer Animation'2003*, pages 68–74. ACM, 2003.

[13] F. Thomas and O. Johnston. *The illusion of life.* Hyperion, 1981.

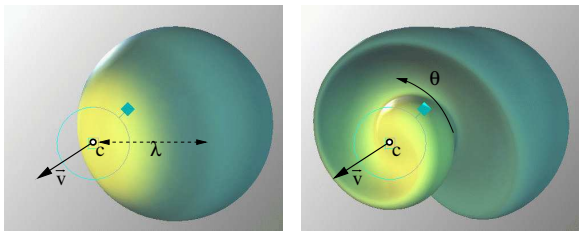

Figure 1. The effect on a sphere of a swirl centered at c, with a rotation angle θ around $\vec{v}$. The two shapes have the same volume.

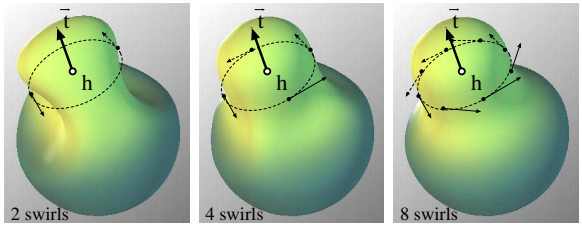

Figure 2. By arranging n basic swirls in a circle, a more complex deformation is achieved. In the rightmost image: with 8 swirls, there are no visible artifacts due to the discrete number of swirls.

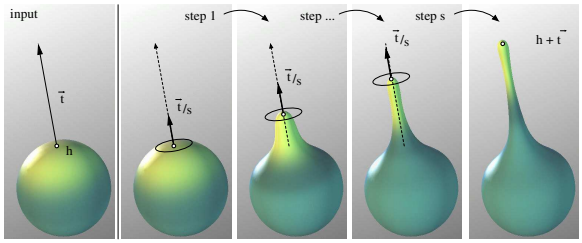

Figure 3. A volume preserving deformation is obtained by decomposing a translation into circles of swirls. 3 steps have been used for this illustration. As the artist pulls the surface, the shape gets thinner. The selected point's transformation is precisely controlled.

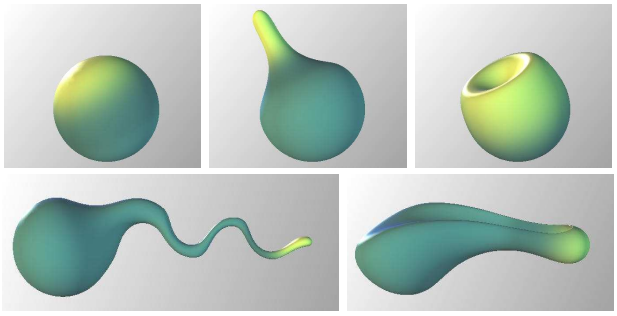

Figure 4. When pushed or pulled, a sphere will inflate or deflate elsewhere.

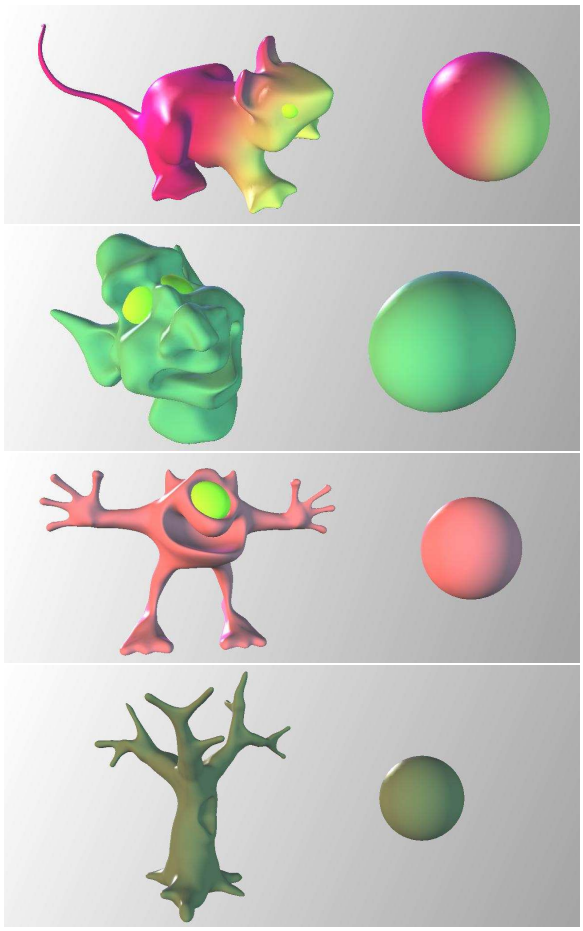

Figure 5. Examples of models "sculpted" with swirling-sweepers. The mouse, the goblin, the alien and the tree have respectively 27607, 25509, 40495 and 38420 vertices. These objects were modeled in less than 30 min by one of the authors. Eyeballs have been added.