EUROGRAPHICS 2006 / E. Gröller and L. Szirmay-Kalos
(Guest Editors)

Volume 25 (2006), Number 3

Sweep-based Freeform Deformations

Seung-Hyun Yoon[†] and Myung-Soo Kim[‡]

School of Computer Science and Engineering, Seoul National University, Seoul 151-744, Korea

Abstract

We propose a sweep-based approach to the freeform deformation of three-dimensional objects. Instead of using a volume enclosing the whole object, we approximate only its deformable parts using sweep surfaces. The vertices on the object boundary are bound to the sweep surfaces and follow their deformation. Several sweep surfaces can be organized into a hierarchy so that they interact with each other in a controlled manner. Thus we can support intuitively plausible shape deformation of objects of arbitrary topology with multiple control handles. A sweep-based approach also provides important advantages such as volume preservation. We demonstrate the effectiveness of our technique in several examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations

1. Introduction

Since its introduction by Sederberg and Parry [SP86], the freeform deformation (FFD) has established itself as one of the most powerful shape design methods for freeform objects. A user of FFD starts with an existing object and changes its shape. This contrasts with the use of sweeps, which allow a designer to create three-dimensional objects from scratch instead of modifying existing shapes. In this paper, we combine these two well-known shape design tools, and propose a new technique for the sweep-based freeform deformation of existing three-dimensional objects.

In conventional FFD [SP86, Coq90, MJ95], the user deforms the shape of an object using control lattices which define trivariate volumes enclosing the parts of an object to be deformed. This requires a hierarchy of multiple lattices which is rather difficult to specify because of its three-dimensional structure. This difficulty motivates us to consider variants of FFD which are based on a one-parameter family of affine transformations [Bar84, CR94] or on a coordinate frame that moves along the axis of an object [LCJ94].

Figure 1 explains the basic idea of our approach. It shows the Utah teapot represented as a union of three deformable parts: the body, the spout and the handle. Each part is approximated by a sweep surface. The boundary vertices of each deformable part are then bound to appropriate cross-sections of the sweep surfaces. By deforming these surfaces, the corresponding parts of the teapot change their shapes. An interesting feature of the Utah teapot is its multiple control handles, and the deformation of one handle influences the others to maintain the consistency of the teapot's topology. There certainly need to be constraints on the allowable interactions between the three different parts of the Utah teapot. Figure 1(b) shows a deformation of the teapot body and the hierarchy of different sweep surfaces, which automatically changes the shape of the handle and the spout so as to maintain the topology of the teapot model as the body changes its shape. Figure 1(c) shows the result of a deformation of the handle, and the hierarchy of sweep surfaces which makes the body and spout change their shapes as the user bends the handle. In this paper, we will address these interaction problems and show how to set up the necessarily complicated interaction rules between different deformations.

Using existing FFD methods, it is difficult, or at best tedious, to support interactions among multiple deformations. Even commercial modeling packages may only support a few limited symmetric interactions. And some FFD methods [Bar84, LCJ94] provide insufficient degrees of freedom to specify arbitrary interactions among multiple deforma-

† shyun@3map.snu.ac.kr
‡ mskim@cse.snu.ac.kr

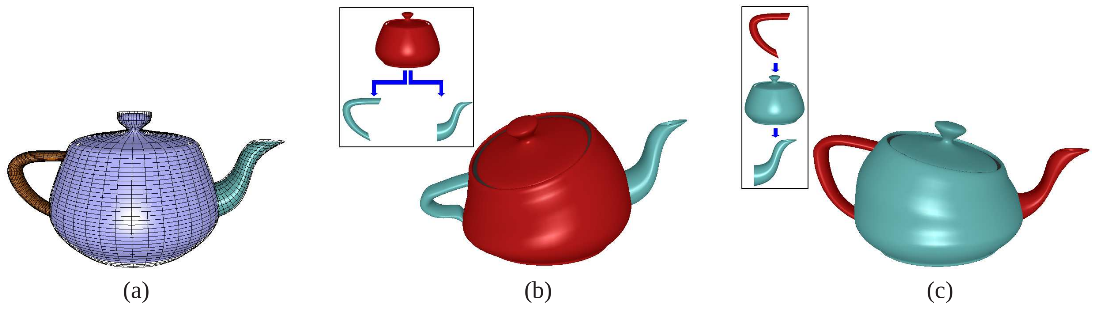


Figure 1: *Interactions among deformations:(a) teapot model and control sweep surfaces, (b) deformation of the body, and (c) deformation of the spout and handle.*

tions. Our new FFD approach uses sweeps to support interactions among different deformable parts in a natural way.

Our technique works by approximating the deformable parts of an object using control sweep surfaces which are constructed by interpolating some key cross-sections. An object is represented using multiple deformable sweep surfaces which can interact because they are organized in a hierarchy. An effective way of binding a sweep surface to other sweep surfaces within such a hierarchy is to bind key cross-sections to cross-sections of the parent part (there are more details in Section 5). Using this simple binding mechanism, we can construct a hierarchy between a number of control sweep surfaces and then deform them simultaneously.

A sweep-based representation provides various additional important advantages, which we itemize in the following summary of the main contributions of our work:

- **Sweep-based FFD** provides an intuitive and effective control mechanism for modifying and editing three-dimensional shapes.
- **Direct control** of the deformation of three-dimensional objects is achieved.
- **Interactions among deformations** are fully supported.
- **Hierarchy-based interaction** of deformations can easily be organized, which facilitates the preservation of object topology.
- **Volume-preserving interaction** is possible because a sweep-based representation greatly simplifies volume computation; both total and relative volumes can be preserved.

The rest of this paper is organized as follows. In Section 2, we briefly review some representative FFD techniques. Section 3 introduces our sweep-based approach to freeform deformation. In Section 4, we propose three control techniques for sweep surfaces and in Section 5 we discuss interactions among deformations of different parts of an object. Experimental results are presented in Section 6. Finally, we conclude this paper in Section 7.

2. Related Work

Freeform deformation (FFD) techniques employ different types of control lattices to construct three-dimensional volumes that surround the objects to be deformed. Sederberg and Parry [SP86] used parallelepiped control lattices. Coquillart [Coq90] extended the types of control lattice to include cylindrical lattices and lattices located on surfaces. Using the Catmull-Clark subdivision scheme, MacCracken and Joy [MJ95] further extended the capability of FFD by introducing lattices of arbitrary topology.

Barr [Bar84] proposed a simple deformation technique for stretching, twisting, bending and tapering solid primitives. This method is essentially the application of a one-parameter family of affine transformations to the cross-sections of an object along its axis. Chang and Rockwood [CR94] clarified the underlying affine structure of this approach in a systematic way. Lazarus et al. [LCJ94] presented a general deformation scheme that is based on a coordinate frame that moves along the axis of an object. Singh et al. [SF98] proposed a deformation technique that uses a parametric curve and influence function. Hyun et al. [HYC*05] considered the deformation of a human body model using sweep surfaces.

These methods all provide some sort of intuitive handle on the deformation of an object. In this paper, we further extend these results so as fully to utilize the underlying affine structures of the moving frames. Our approach offers effective ways of controlling the deformation of an object while preserving a visual resemblance to its original shape. An important advantage of this approach is that we can support various types of interaction among different parts of an object.

Recently, some new deformation techniques have been proposed. Hua and Qin [HQ03] presented a modified FFD in which they employ a scalar field as the embedding space instead of a volume. The vertices of an object are parameterized by the level-set of a scalar field and follow its deformation. Angelidis et al. [ACWK04] introduced a volume-preserving operator as a shape deformation tool. Simplified polygonal meshes can also be used as control structures [SK00, CO03], as can continuous parametric surfaces [FMP96, COJ*02]. Yoshizawa et al. [YBS03] presented a skeleton-driven mesh deformation technique in which they used a Voronoi-based skeletal mesh as a control structure for freeform models. Ju et al. [JSW05] generalized mean-value coordinates from closed 2D polygons to closed triangular meshes and applied them to mesh deformation. Using a rigid motion-invariant mesh representation, Lipman et al. [LSLC05] proposed an interactive mesh editing and shape interpolation technique.

To improve the controllability of FFDs, a number of direct manipulation techniques have been proposed. Hsu and Kaufman [HK92] introduced the direct manipulation of FFDs, and later Hu et al. [HZTS01] presented an explicit solution to this problem using constrained optimization. Chang et al. [CLKH98] also proposed the direct manipulation of a generalized cylinder, which is somewhat similar to our sweep surface.

3. Sweep-based Freeform Deformation

Sweeps are a procedural modeling technique for representing three-dimensional freeform objects [PT97]. In this section, we describe our sweep-based FFD.

Sweep surface from a continuous motion. A sweep surface generated by a continuous motion provides a nice control structure for FFDs. Let $\{X_i\}$ be a set of key cross-sections. Each key cross-section X_i is associated with a local transformation $T_{i-1,i}$ (represented by a 4×4 matrix) from the previous key cross-section X_{i-1}. Thus, when a key cross-section X_j changes its position and orientation, all the following key cross-sections $X_k (k > j)$ are automatically updated by a series of relative transformations. This arrangement facilitates intuitive deformations such as bending and twisting.

Our sweep surface is generated by interpolating these key cross-sections $\{X_i\}$:

$$\begin{aligned} S(\theta,t) &= C(t)+R(t)\cdot O_t(\theta) \\ &= \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} r_{11}(t) & r_{12}(t) & r_{13}(t) \\ r_{21}(t) & r_{22}(t) & r_{23}(t) \\ r_{31}(t) & r_{32}(t) & r_{33}(t) \end{bmatrix} \cdot \begin{bmatrix} r(\theta,t)\cos\theta \\ r(\theta,t)\sin\theta \\ 0 \end{bmatrix}, \end{aligned}$$

where $O_t(\theta)$ represents a star-shaped cross-sectional closed curve and $C(t)$ and $R(t)$ describe its position and orientation respectively.

Figure 2(a) shows a set of key cross-sections and Figure 2(b) shows the sweep surface generated by interpolating them. As shown in Figure 2, our sweep surface represents a time-variant star-shaped cross-section by a scalar radius function $r(\theta,t)$. This sweep surface can be used as a control structure for bending, twisting and tapering. Moreover, to achieve local deformations, we can change the cross-sectional shape of a sweep surface by modifying the scalar radius function $r(\theta,t)$.

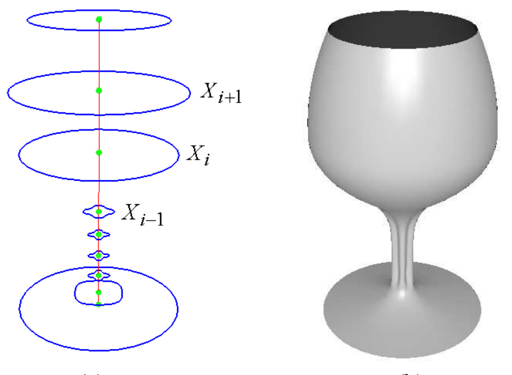


Figure 2: *Sweep surface:(a) key cross-sections, (b) resulting sweep surface.*

If the sweep surface $S(\theta,t)$ has no self-intersection, it bounds a volume of magnitude

$$\int A(t)\left\langle N(t), C'(t)\right\rangle dt, \qquad (1)$$

where $A(t)$ is the area of the cross-section $O_t(\theta)$ (see [STV03]) and $N(t)$ is a unit normal vector of the moving cross-sectional plane, which appears as the third column vector of $R(t)$. In Section 5, we will show how to use this simple integral formula to solve an interaction problem while preserving (both absolute and relative) volumes among multiple deformations.

Control sweep surface construction. Before changing the shape of an object, its deformable parts must be approximated by sweep surfaces. We start by selecting the parts to be deformed and creating initial control sweep surfaces which enclose those parts. Figure 3(a) shows the leg of the well-known Armadillo model and an initial control sweep surface.

In the second step, key cross-sections are computed by cutting the polygonal model of the selected part with planes. The center of each key cross-section is computed, and then radii from the center to the boundary vertices of the cross-section are sampled. Figure 3(b) shows a key cross-section with sampled radii. A control sweep surface can then be constructed by interpolating these key cross-sections. The centers of the key cross-sections are interpolated by a cubic B-spline curve $C(t)$ using chord length parametrization, and their orientations (represented by unit quaternions) are linearly interpolated and normalized to form a rotation matrix $R(t)$. The sampled radii on each key cross-section are interpolated by a B-spline function $r(\theta,t)$. Figure 3(c) shows the control sweep surface constructed by interpolating the computed key cross-sections.

In the third step, we can manually edit the positions and orientations of these key cross-sections, insert additional key

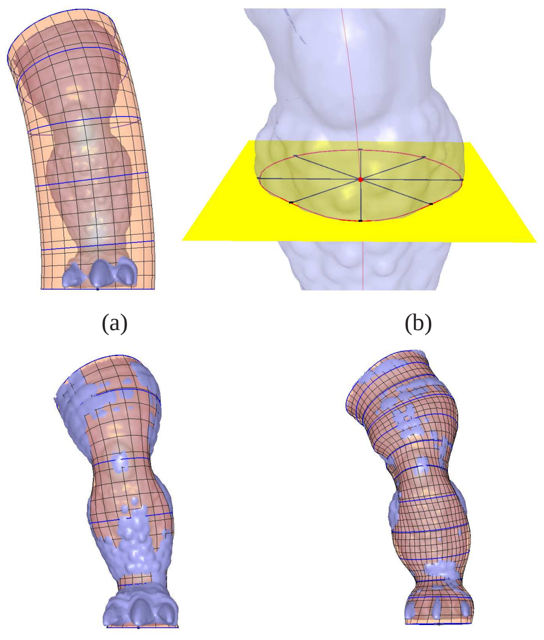


Figure 3: *Control sweep surface construction:(a) the selected part of the Armadillo leg and the initial control sweep surface, (b) a key cross-section and sampled radii, (c) an intermediate control sweep surface, and (d) the final tightly fitting control sweep surface.*

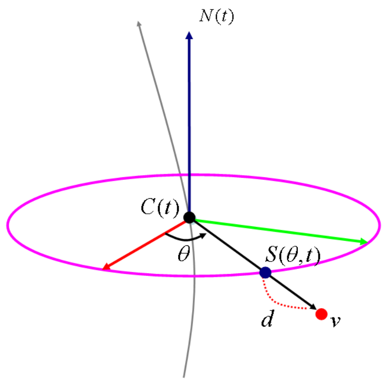


Figure 4: *Binding a vertex to a control sweep surface.*

cross-sections if necessary. Then the second and third steps are repeated until a tightly fitting surface has been generated.

Figure 3(d) shows a control sweep surface constructed in this iterative way including the insertion of some additional key cross-sections with manually adjusted orientations. Figures 5(b) and 6(b) show control sweep surfaces for all the deformable parts of a dinosaur and a bunny model.

Vertex binding and deformation. Once a control sweep surface has been constructed, the vertices in the part of the original model that is to be deformed are bound to that surface. To do this, we first compute the cross-sectional plane that contains a vertex v by solving

$$< v - C(t), N(t) >= 0,$$

where $< \cdot, \cdot >$ signifies an inner product, and $N(t)$ represents a unit normal vector of the moving cross-sectional plane. The local coordinate of the vertex v in that plane is then determined from $\hat{v} = R(t)^T(v - C(t))$. The circular binding angle θ is $arctan(\hat{v}_y/\hat{v}_x)$. Finally, the signed distance d from the control sweep surface is computed as $d = \|\hat{v}\| - \|S(\theta,t) - C(t)\| = \|v - C(t)\| - \|S(\theta,t) - C(t)\|$. Figure 4 shows an example of this binding procedure in which a vertex v is bound to a cross-section of a control sweep surface using the binding parameters (θ, t, d).

During a deformation, the user can change the shape of the control sweep surface to which all the vertices of the deformable part are bound. Then all the vertices of that part can be reconstructed using the following equation, and will then track the deformation of the control sweep surface.

$$v = C(t) + R(t) \begin{bmatrix} (r(\theta,t) + d) \cdot cos\theta \\ (r(\theta,t) + d) \cdot sin\theta \\ 0 \end{bmatrix}. \qquad (2)$$

Figures 5(c) and 5(d) show the deformation of a dinosaur model and Figures 6(c) and 6(d) show the deformation of the neck and ears of a bunny model, all of which we achieved by editing the positions and orientations of a few key cross-sections.

4. Sweep Surface Control

In this section, we will introduce three types of control technique for sweep surfaces: editing key cross-sections; directly controlling a point on a sweep surface; and controlling the position and orientation of an arbitrary cross-section. We start with the editing of key cross-sections and move on to the higher-level control techniques, which provide more intuitive and effective deformations.

Editing key cross-sections. We now consider how to apply various deformations such as bending, twisting, and tapering to sweep-based models by editing their key cross-sections. During a deformation, a user can select one key cross-section X_i and edit its local position and orientation $T_{i-1,i}$ with respect to the previous key cross-section X_{i-1}. These changes affect consecutive key cross-sections $\{X_k\}$, $k = i, i+1, \cdots, n$, and update their global positions and orientations $\{T_{0,k}\}$. Alternatively, we can edit a key cross-section X_i independently of other key cross-sections. We can also easily edit the radius function $r(\theta,t)$ of key cross-sections, in addition to transforming them. The edited key cross-sections are interpolated and the corresponding sweep surface is constructed. Figure 7 shows the results of these editing operations.

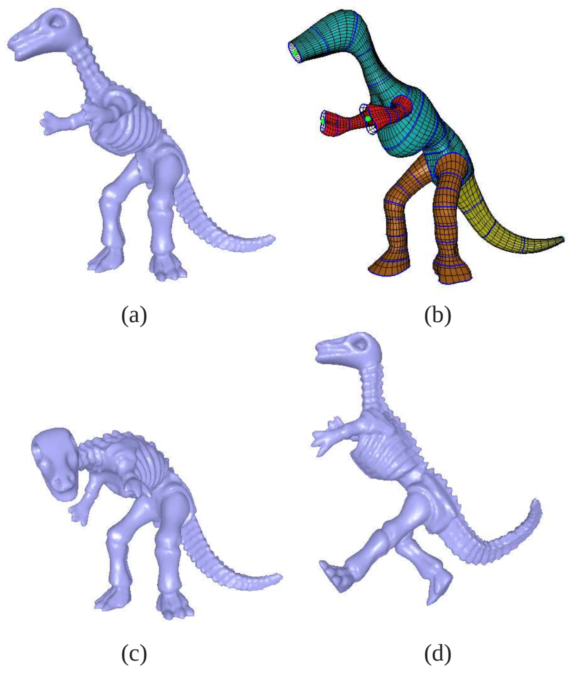


Figure 5: *Sweep-based deformations:(a) dinosaur model, (b) control sweep surfaces, (c) and (d) deformations of the dinosaur model.*

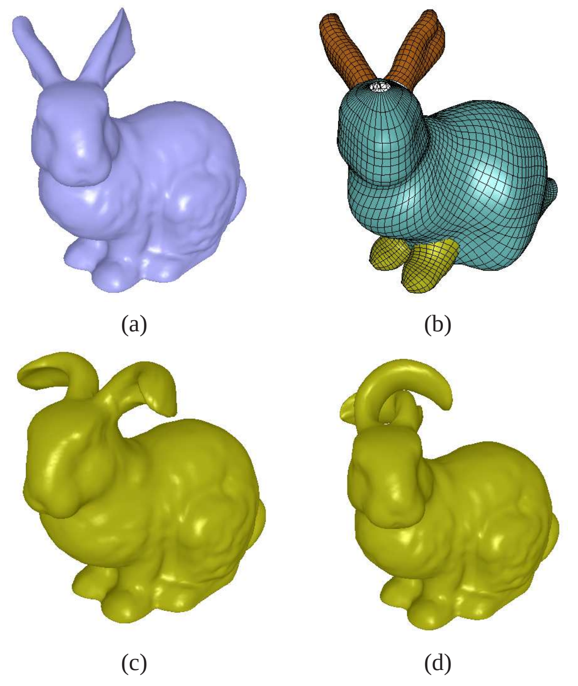


Figure 6: *Sweep-based deformations:(a) bunny model, (b) control sweep surfaces, (c) and (d) deformation of the ears and neck.*

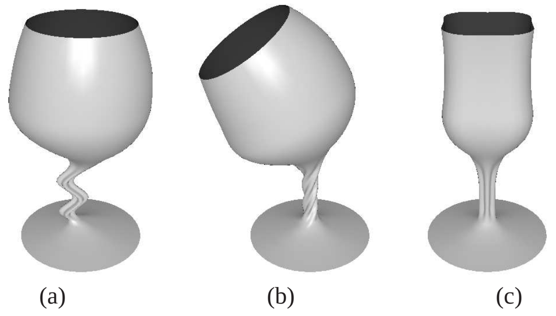


Figure 7: *Editing key cross-sections by changing their: (a) position, (b) orientation and (c) radius.*

Direct control of a surface point. Sometimes a user wants to deform the shape of an object directly by editing points on its boundary. When the user picks and manipulates a vertex v with binding parameters (θ, t, d), the corresponding cross-section moves and rotates, resulting in a new configuration, and $S(\theta, t)$ must change to accommodate this offset. The basic idea of our technique is to edit appropriate key cross-sections so that the sweep surface passes through the changed target point. However, the relation between a point on a sweep surface and its key cross-sections is highly nonlinear, and there are many possible ways to modify a sweep so that its surface contains a specified point.

Figure 8 shows a strategy based on an infinitesimal editing operation. The user picks a point v (in red) on a sweep surface and moves it to a point v' (in blue), which generates a difference vector Δv at the vertex v. Our strategy is to divide Δv into a linear component Δt and an angular component Δq, and then to start a numerical target tracking process. We first select the lower key cross-section X_i, and consider a sphere centered at the origin p_i of X_i, with radius $\|v - p_i\|$. The vertex v' is then projected on to the sphere, where it generates a new vertex v''. The linear component Δt is the difference $v'' - v'$, and the rotational axis is computed as the cross product $(v'' - p_i) \times (v - p_i)$ and then normalized. The magnitude of Δq, which determines the rotation of the cross-section, is computed as follows:

$$\|\Delta q\| = arccos\left(\frac{< v - p_i, v'' - p_i >}{\|v - p_i\| \cdot \|v'' - p_i\|}\right).$$

By applying both the linear and angular difference vectors to the key cross-section X_i, the vertex v approaches the target position v' when a new sweep surface is constructed. To achieve convergence, we repeat this procedure until the distance from v to v' is reduced within a given tolerance. For stability, we propagate control to the k lower key cross-sections $X_i, \cdots, X_{i-k+1}$, starting from the lowest key cross-section X_{i-k+1} and ending at X_i. In general, each key cross-section has 6 degrees of freedom and there are a number of possible ways to bring the sweep on target, of which our

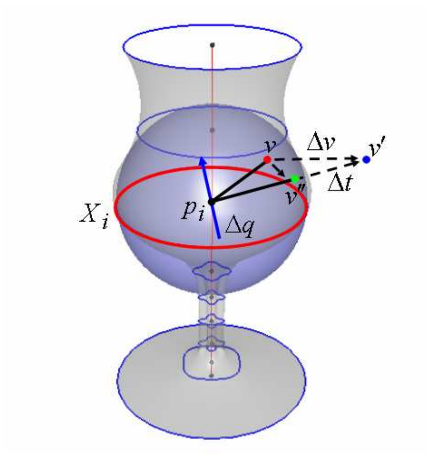
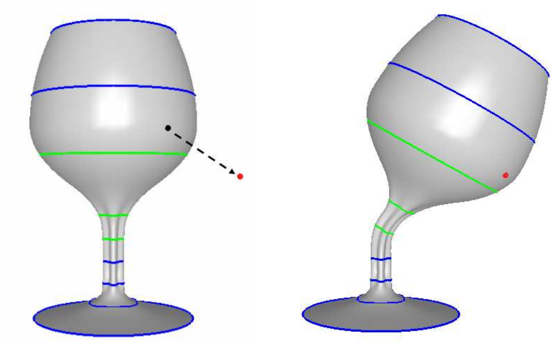


Figure 8: *The separation of a difference vector* Δv*.*

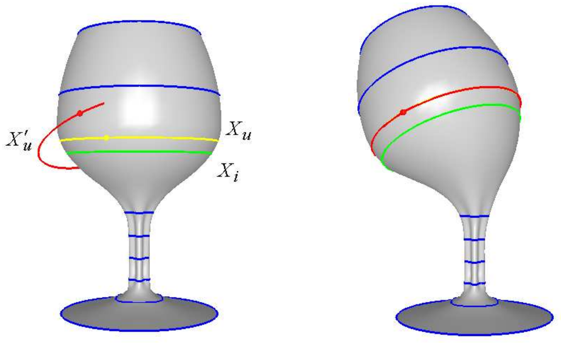


(a) (b)

Figure 9: *Direct control of a sweep surface.*

technique is only one. Alternatively, we can use just one of the difference vectors to achieve other types of control. Figure 9(a) shows a point selected by the user v (in black), the target position v' (in red), and the affected key cross-sections (in green). Figure 9(b) shows the result of the control process, in which both the linear and angular components of the change have been combined.

Direct control of an arbitrary cross-section. We now propose another direct control technique for a sweep surface. (Later, we will use this technique to support hierarchy-based interactions among different parts of an object.) During a deformation, the user selects a point on a sweep surface and changes the position and orientation of the cross-section that contains that point. In Figure 10(a), X_u is the selected cross-section and X'_u is the new cross-section. We edit the key cross-section X_i which is closest to X_u, assuming that X_u will follow X_i to a large extent, and repeat this process until the transformational distance between X_u and X'_u is reduced within a given tolerance. Figure 10(b) shows the direct con-



(a) (b)

Figure 10: *Direct control of an arbitrary cross-section.*

trol of the cross-section X_u using this method. Although we only edited X_i in Figure 10, this method can be extended to the editing of multiple key cross-sections, as discussed above.

5. Interactions among Deformations

We will now consider the problem of simultaneously controlling the interaction between a number of different deformations of an object using multiple control handles. As a simple example, we will edit the Utah teapot, using separate handles to control the body, spout and handle. Each of these components is approximated by a sweep surface, and its shape is changed by controlling that surface. We need to maintain the consistency of the model's topology during deformations. For example, when the user bends the handle, the body should follow. When the body is modified, the handle and the spout need to track the deformation of the body. We require a mechanism to specify the hierarchy of these deformations, and rules to control their interactions (see Figure 1).

Hierarchy of sweep surfaces. We now show how we specify a hierarchy of sweep surfaces. In the teapot example of Figure 1(b), the body is a root node, and the handle and the spout are child nodes. A hierarchy is then specified by binding key cross-sections of child nodes to cross-sections of their parent node. In the teapot example, there are two types of child node. One is a loop, for the handle, and the other is a branch, for the spout. As shown in Figure 11, both the first and last key cross-sections of the handle node (loop type), namely X^{handle}_{first} and X^{handle}_{last}, are bound to cross-sections of the body, namely $X^{body}_{handle\text{-}first}$ and $X^{body}_{handle\text{-}last}$. But for the spout node (branch type), only the first key cross-section of the spout, X^{spout}_{first}, is bound to cross-section $X^{body}_{spout\text{-}first}$ of the body. The binding of a key cross-section is a simple extension of the vertex binding technique introduced in Section 3, and can be represented by the binding parameters $(\theta, t, d, \hat{q})$. For example, the three binding parameters

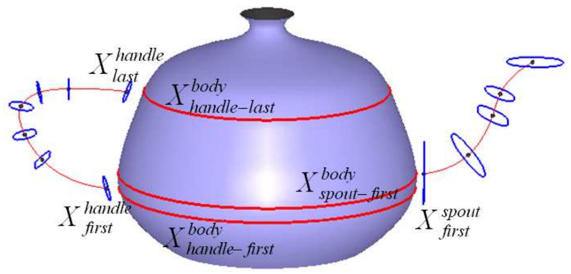


Figure 11: *Binding of key cross-sections.*

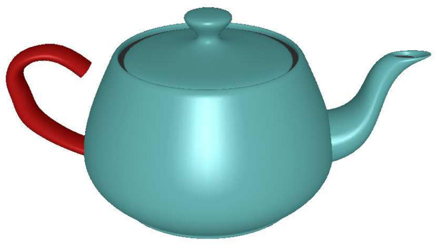


Figure 12: *Broken topology during a deformation.*

$(\theta^{handle}_{first}, t^{handle}_{first}, d^{handle}_{first})$ represent the displacement of the center of the key cross-section X^{handle}_{first} from the body sweep surface, as expressed in Equation (2), and the fourth binding parameter $\hat{q}^{handle}_{first}$ represents its relative orientation to the cross-section $X^{body}_{handle\text{-}first}$ of the body sweep surface. Using this binding information, the bound key cross-sections change their positions and orientations when the sweep surface of their parent node is deformed. In the teapot example of Figure 1(c), the handle is a root node, the body is a child node and the spout is a grandchild node. The rest of the interactions can be realized in a similar fashion.

Hierarchy-based interactions. Once the cross-sections of all child nodes have been bound to the sweep surface of their parent node, following the hierarchical structure, we need to solve the problem of interactions among multiple deformations.

In the teapot example, when the user changes the shape of the body, the handle and the spout should follow the deformation of the body. The first key cross-section X^{spout}_{first} of the spout is simply updated using its binding parameters $(\theta^{spout}_{first}, t^{spout}_{first}, d^{spout}_{first}, \hat{q}^{spout}_{first})$, and then the positions and orientations of the rest of the key cross-sections X_j, $j = 2, \ldots, n$, are automatically updated using the local transformations $T_{j-1,j}$ from the previous key cross-section X_{j-1}. The control sweep surface of the spout node is then reconstructed by interpolating the updated key cross-sections, and all vertices bound to that sweep surface are also reconstructed using their binding parameters. For the handle, the problem becomes more complicated. We can easily compute the new positions and orientations of the two key cross-sections X^{handle}_{first} and X^{handle}_{last} from their binding parameters. But this may result in an undesirable deformation because the intermediate key cross-sections are not considered. So then we compute the difference between the new transformation of the last key cross-section X^{handle}_{last} and its old transformation. Finally, we apply our direct control technique to the last key cross-section X^{handle}_{last}, using the difference between these transformations. When the direct control technique is applied, all key cross-sections except the first one X^{handle}_{first} are updated iteratively, and the result is a natural deformation of the handle. Figure 1(b) shows a deformation of the teapot body in which the handle and the spout change their shapes automatically so as to maintain the topology of the teapot model as the body changes its shape.

In the case of the deformation of the spout, we do not need to take any additional action, because there is no topological constraint on the spout and it can deform independently. But when we modify the handle, we need to deform the body as well, so that the loop-type topology of the handle and the body is maintained. Figure 12 shows the broken topology that occurs when we do not consider the simultaneous control of deformations. To maintain the teapot's topology, we need to apply our direct control technique to the cross-section $X^{body}_{handle\text{-}last}$ of the body to which the last key cross-section X^{handle}_{last} of the handle is bound (see Figure 11). When a user deforms the control sweep surface of the handle node, the last key cross-section X^{handle}_{last} changes its position and orientation. From the new values we can compute the target position and orientation of the cross-section $X^{body}_{handle\text{-}last}$ of the body node. The difference between the target and the previous transformation of the cross-section $X^{body}_{handle\text{-}last}$ is then computed. Finally, using the direct control technique for an arbitrary cross-section which we introduced in Section 4, we can control the sweep surface of the body so that its cross-section $X^{body}_{handle\text{-}last}$ coincides with the target position and orientation, thus maintaining the topological constraints on the teapot model. Figure 1(c) shows the result of a deformation of the handle, in which the body and spout change their shapes automatically as the user bends the handle.

Volume-preserving interactions. The volume of a sweep surface is given in Equation (1). Using this integral formula, we have developed two simple interaction rules which can be applied to deformations of an object with multiple control handles. The first is a **total volume preservation** rule which maintains the overall volume of an object during a deformation. When the deformable part of an object changes its shape, it generates a change ΔV in the volume of this part. This difference ΔV is uniformly divided between each control handle, and each radius function $r(\theta, t)$ is updated

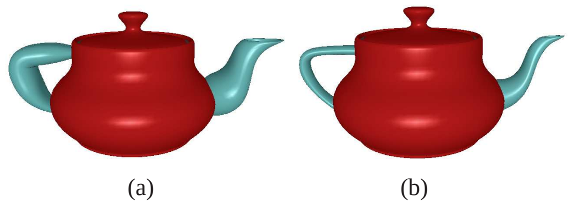


Figure 13: *Volume-based interactions:(a) total volume preservation, (b) volume ratio preservation.*

to reflect to the proportion of the volume distributed to that handle. First, the target volume of each control handle is computed. We then increase or decrease the sampled radii of each key cross-section and repeat the process until the target volume is reached. Figure 13(a) shows the result of a deformation of the teapot using the volume-preservation rule, in which the spout and the handle increase their volumes as the body shrinks. The second interaction rule is a **volume ratio preservation** rule, which maintains the ratio of the volume of a child node to that of its parent node. This rule is implemented using a similar update of the radii of key cross-sections, embedded within an iteration process. Figure 13(b) shows the result of a deformation using the volume ratio-preservation rule, where the ratio between the volumes of the spout and handle, and that of the body, are preserved.

6. Experimental Results

We have implemented our sweep-based freeform deformation technique in C++ on a P4-3.2GHz PC with a 2GB main memory. Our deformation technique works in real-time for all the test examples presented in this paper; it is simple to implement and easy to use. The most time-consuming step is to construct control sweep surfaces for all the deformable parts of an object. This usually takes from five to ten minutes. To minimize this construction time, we have developed a user interface which includes a semi-automatic process to compute key cross-sections, and their positions and orientations can subsequently be edited manually. The processing time for vertex binding step depends on the number of vertices to be bound to a control sweep surface. Table 1 lists the number of vertices in each model and the time required for vertex binding.

Figure 14 shows the results of the direct deformation of a bunny model using five control sweep surfaces. In Figures 14(a) and 14(c), the user picks a vertex with the binding parameters (θ, t, d) and specifies its target position, which generates a displacement vector. Our direct control technique is then applied to the point $S(\theta, t)$ on the sweep surface. In Figure 14(b), only the translational component of the displacement vector is considered; while in Figure 14(d), only the rotational component is considered.

Table 1: *Number of vertex and binding time (sec).*

Models	Bunny	Dinosaur	Teapot
Number of vertices	19,226	14,048	12,882
Time for vertex binding	21.714	16.580	13.820

Figure 15 shows the deformation of a teapot model, using the hierarchy-based interaction rule combined with editing of the radius function $r(\theta, t)$ to achieve a local deformation.

We have also extended our technique to support arbitrary interactions among deformable parts. In Figure 16(a), four legs of a chair model are approximated by control sweep surfaces. To facilitate interactions among them, they are virtually interconnected by three auxiliary sweep surfaces (shown in red and cyan). Figure 16(b) shows the deformation of a front leg with no interaction. In Figure 16(c) the user changes the shape of a single leg and both front legs change their shapes because of the presence of a virtual sweep that connects them. In Figure 16(d) three virtual sweeps are involved, and all four legs of the chair become elongated.

7. Conclusions

We have shown that a sweep-based approach provides an excellent control mechanism for deforming three-dimensional objects. Once an object has been approximated with sweep surfaces, it is easy to control shape deformations using a small number of sweep parameters. We have also proposed various control techniques for sweep surfaces, which allow the user to change the shape of an object intuitively and effectively. When an object has multiple deformable parts, it is straightforward to build a hierarchy of deformations using sweeps, and it is also easy to describe the interactions among them. Moreover, our approach supports local deformations by the editing of a radius function $r(\theta, t)$.

The main difficulty in our approach is the construction of the control sweep surfaces, which requires some user intervention. This can be ameliorated by incorporating convenient user interfaces or advanced surface fitting techniques.

In the current implementation, we focused on applying our deformation technique to three-dimensional mesh models. In future work, we will investigate the feasibility of controlling the shape of freeform objects which have other representations, such as implicit surfaces or procedural models, and extend our technique to support more complex interaction rules.

Acknowledgements

The authors would like to thank the anonymous reviewers for their invaluable comments. This work was supported by the Korean Ministry of Information and Communication (MIC) under the Program of IT Research Center on CGVR.

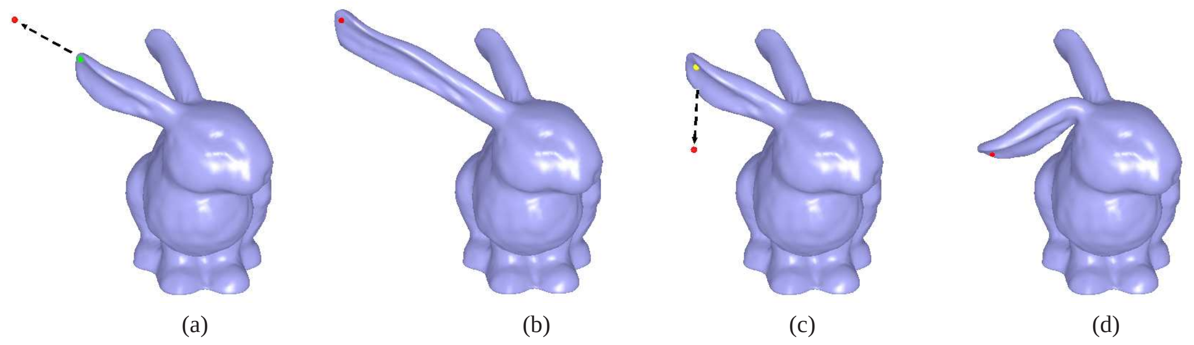


Figure 14: *Direct deformations of a bunny model.*

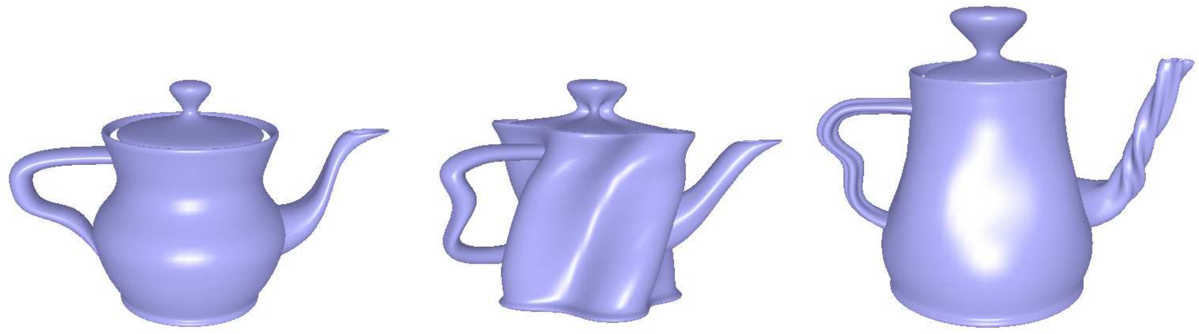


Figure 15: *Deformations of a teapot model.*

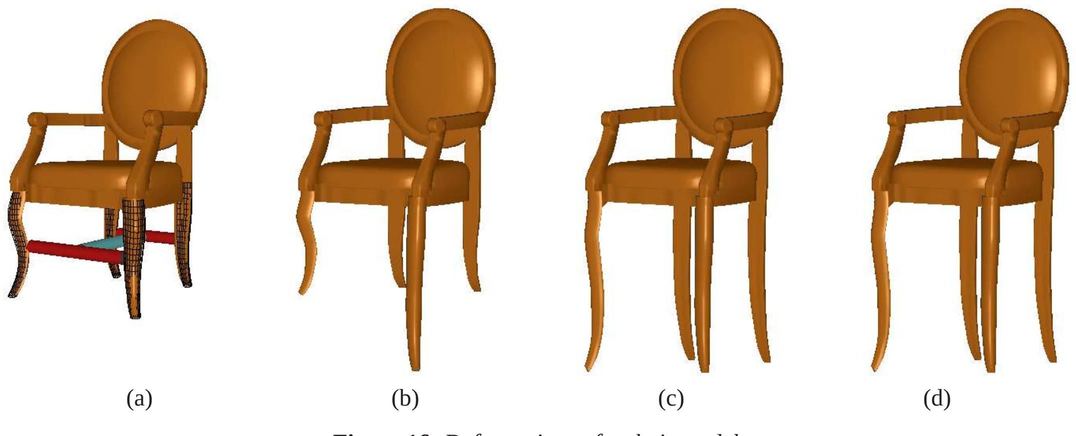


Figure 16: *Deformations of a chair model.*

References

[ACWK04] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-sweepers: constant-volume modeling. *Proc. of Pacific Graphics* (2004), 10–15.

[Bar84] BARR A.: Global and local deformations of solid primitives. *Computer Graphics 18*, 3(1984), 21–30.

[Coq90] COQUILLART S.: Extended free form deformation: a sculpturing tool for 3D geometric modeling. *Computer Graphics 24*, 4 (1990), 187–196.

[CLKH98] CHANG T.-I., LEE J.-H., KIM M.-S., HONG S.-J.: Direct manipulation of generalized cylinders based on rational motion. *The Visual Computer 18*, 5 (1998), 228–239.

[CO03] COBAYASHI K., OOTUSUBO K.: t-FFD: free form deformation by using triangular mesh. *Proc. ACM Symposium on Solid Modeling and Application* (2003), 226–234.

[COJ*02] CHEN B., ONO Y., JOHAN H., ISHII M., NISHITA T., FENG J.: 3D model deformation along a parametric surface. *Proc. International Conference on Visualization, Imaging and Image Processing* (2002), 282–287.

[CR94] CHANG Y.-K., ROCKWOOD A.: A generalized de Casteljau approach to 3D free form deformation. *Proc. SIGGRAPH '94* (1994), 257–260.

[FMP96] FENG J., MA L., PENG Q.: A new free form deformation through the control of parametric surfaces. *Computers and Graphics 20*, 4 (1996), 531–539.

[HK92] HSU W., KAUFMAN H.: Direct manipulation of free form deformation. *Computer Graphics 26*, 2 (1992), 177–184.

[HQ03] HUA J., QIN H.: Free form deformations via sketching and manipulating the scalar fields. *Proc. ACM Symposium on Solid Modeling and Application* (2003), 328–333.

[HYC*05] HYUN D.-E., YOON S.-H., CHANG J.-W., SEONG J.-K., KIM M.-S., JÜTTLER B.: Sweep-based human deformation. *The Visual Computer 21*, 8 (2005), 542–550.

[HZTS01] HU S., ZHANG H., TAI C., SUN J.: Direct manipulation of FFD. *The Visual Computer 17* (2000), 370–379.

[JSW05] JU T., SCHAEFER S. AND WARREN J.: Mean value coordinates for closed triangular meshes. *Proc. SIGGRAPH '05* (2005), 561–566.

[LCJ94] LAZARUS F., COQUILLART S., JANCÈNE P.: Axial deformations: an intuitive deformation technique. *Computer-Aided Design 26*, 8 (1994), 607–613.

[LSLC05] LIPMAN Y., SORKINE O., LEVIN D. AND COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *Proc. SIGGRAPH '05* (2005), 479–487.

[MJ95] MACCRACKEN R., JOY K.: Free form deformations with lattices of arbitrary topology. *Computer Graphics* (1995), 181–189.

[PT97] PIEGL L. AND TILLER W.: *The NURBS Book*, 2nd ed. (1997). Springer.

[SF98] SINGH K., FIUME E.: Wires: a geometric deformation technique. *Computer Graphics* (1998), 405–414.

[SK00] SINGH K., KOKKEVIS E.: Skinning characters using surface-oriented free form deformations. *Proc. Graphics Interfaces* (2000), 35–42.

[SP86] SEDERBERG T., PARRY S.: Free form deformation of solid geometric models. *Computer Graphics 20*, (1986), 151–160.

[STV03] SHEYNIN S., TUZIKOV A., VASILIEV P.: Volume computation from nonparallel cross-section measurements. *Pattern Recognition and Image Analysis 13*, 1 (2003), 174–176.

[YBS03] YOSHIZAWA S., BELYAEV A.-G., SEIDEL H.-P.: Free form skeleton-driven mesh deformations. *Proc. ACM Symposium on Solid Modeling and Application* (2003), 247–253.