Surface Drawing

Steven Schkolne* Peter Schröder†

Caltech Department of Computer Science
Technical Report CS-TR-99-03

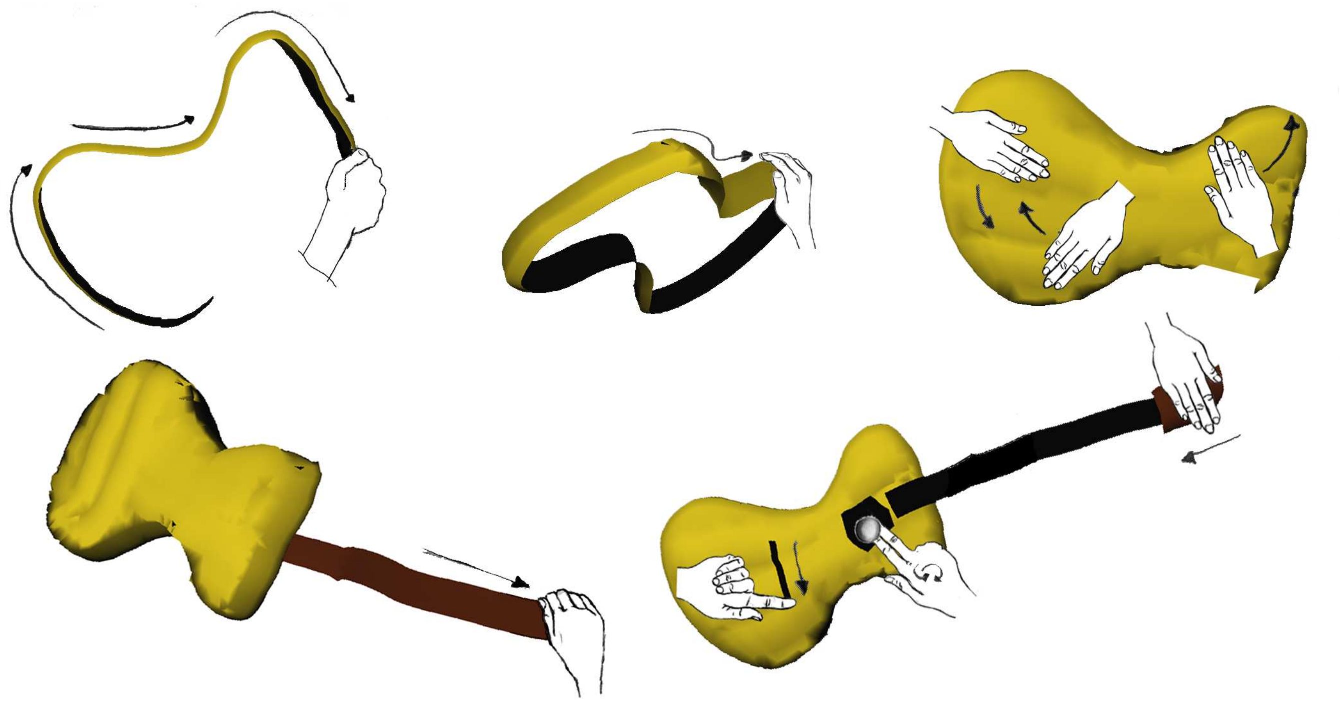

Figure 1: *A guitar is Surface Drawn by forming its shape with the hand.*

Abstract

We present Surface Drawing, a medium which provides direct control over the creation of a wide range of intricate shapes. Surface Drawing addresses several key issues in creative expression and perceptual thinking by providing a direct link between the motions of the hand and the forging of shapes. Surfaces are created by moving a hand, instrumented with a special glove, through space in a semi-immersive 3D display and interaction environment (the Responsive Workbench). This technique allows both novices and experts to create intricate forms without the perceptual constraints of a rigid mathematical structure, large toolset, or a reduction of modeling to editing. In Surface Drawing the design space can be freely explored during the modeling process without the need to plan the construction of the final shape. In particular it supports unconstrained erasing and buildup of new geometry. This is achieved through the use of a novel incremental construction method for triangulated meshes, the Cookie Cutter algorithm. It allows the user to freely grow, join, and erase surfaces based on hand motions. We report on our experiences with the system and present results created by artists and designers exploring problems in industrial design, character design, and fine art.

*ss@cs.caltech.edu
†ps@cs.caltech.edu

1 Introduction

The state of the art in three-dimensional modeling can represent a variety of complex smooth surfaces. When artists and designers create shapes with these techniques, we observe that often the method of surface representation restricts the ways in which a shape can be modeled. Consider traditional spline-based modeling [8]. While splines can represent many different shapes, a user often has to think of the most efficient way to represent a shape before beginning the modeling process. Changing directions midway through the design process often requires such a drastic change in the underlying placement of patches that the user must return to the drawing board, essentially starting from scratch [9].

The traditional shape modeling approach of the graphics and CAGD communities is fundamentally different from the approach taken by the artistic community. Consider the perhaps simplest of all modeling tools: the pencil. It is an extraordinarily effective conduit for artistic expression. Part of the reason for this is its simplicity. Another reason for its success is the close relation between an artist's perception and action and the forms the pencil produces. This link yields direct control over all aspects of form. Surface Drawing, which we introduce, provides direct control over three-dimensional space in the same way a pencil commands two-dimensional space.

The key to Surface Drawing is the use of motions of the human hand to describe shapes, as in Figures 1 and 4. As the hand is moved in a semi-immersive environment, a surface is grown by adding the shape of the hand at each sampling interval. The versatility and natural understanding of the hand, combined with the simplicity of this process, allows for an intuitive modeling process in which the user can think perceptually about a shape while constructing it. While the resulting surface might not meet certain analytic criteria, such as curvature continuity, forms can be constructed with gestural, emotive qualities that are difficult to achieve with traditional surface modeling tools. This system has both the ease of use that novices expect and the control that experts demand. We emphasize that the goal of our work is the creation of tools which support conceptual design and the artistic process. These methods are distinct from those focusing on exact specification of geometry, as needed in traditional CAD.

To better understand why Surface Drawing works, we introduce the concept of *perceptual thinking*. This idea describes how thought and interface are intimately related. By reducing the distance between the actions of the modeling system and the perceptions and actions of a user, the cognitive overhead of a modeling system is greatly reduced, freeing up resources to think about the form being created.

We represent our surfaces as meshes, and develop the Cookie Cutter algorithm for incrementally constructing a surface as it is being created.

Our principal contributions are (a) the formulation of the principle of perceptual thinking, (b) the Surface Drawing method, which leverages perceptual thinking, and (c) the Cookie Cutter incremental surface construction algorithm which we use in our implementation of the Surface Drawing method. These topics are presented as follows:

- **Section 2:** Understanding modeling through *perceptual thinking*. Specific guidelines for creative perceptual thinking are presented. Previous work is evaluated using these guidelines.
- **Section 3:** The Surface Drawing paradigm is described in detail.
- **Section 4:** The Cookie Cutter algorithm, an interactive incremental surface construction technique, is presented.
- **Section 5:** Description of the ways in which the hand is sampled and the tools we provide for modeling.
- **Section 6:** Applications of Surface Drawing to industrial design, character design, and fine art are presented.
- **Section 7:** We discuss the benefits and shortcomings of the present work. Future areas of research are proposed.

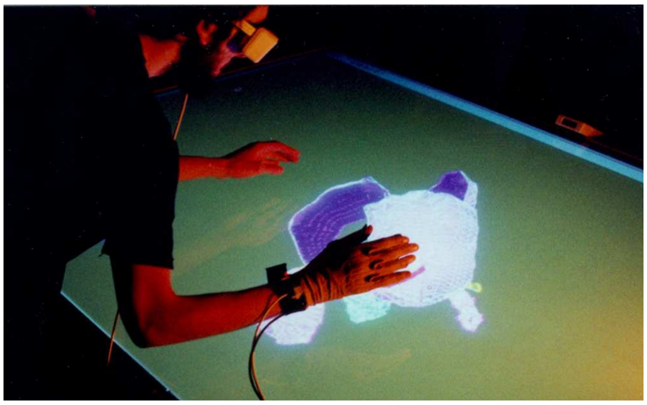

Figure 2: *Perceptual interaction with a surface in the semi-immersive environment of the Responsive Workbench.*

2 Perceptual Thinking

In his seminal text *Visual Thinking* [4], Arnheim describes the importance of perception in our thought processes:

> The cognitive operations called thinking are not the privilege of mental processes above and beyond perception, but the essential ingredients of perception itself.

Arnheim finds vision to be the most important of the senses, and shows how thinking through images pervades society. McKim [17] applies the concept of visual thinking to the problem-solving process. He cites the difficulties of materials and techniques that draw attention away from the thinking process, and the need for direct interaction to support rapid ideation.

Visual thinking can be expanded to include all of the senses (including proprioception) in what we call *perceptual thinking*. Perceptual thinking is a process in which understanding is achieved through direct perception, without being translated into a higher linguistic or mathematical form. In a problem solving environment, application of perceptual thinking minimizes the distance between perception and the artifact being created.

Consider the example of describing a set of points on a two-dimensional grid for later display. A weak perceptual interface for this task would be to write down coordinates for these points or enter them with a keyboard, only allowing the user to see them after the input is complete. A strong perceptual interface would be to draw the points on a piece of paper, or allow the user to enter them with a touch-sensitive computer display. The latter form of interaction gives the user a much stronger aesthetic control over the points being described.

Applying these principles, we develop the following guidelines for an interactive modeling system:

Invisible mathematical structure The behavior of the modeling system should be based upon simple physical interactions that are understood via direct perception. The user should think primarily in terms of the model being created, instead of a structure that must be created to support that model. Any mathematical structure that exists must support arbitrary changes in the model in a way that is transparent to the user.

Direct primitives Presenting an artist with a series of complex primitives (sphere, cube, cone, etc.) forces thought in terms of these primitives. Unless the solution space inherently consists of these primitives (packing oranges or stacking bricks) they will only take the user a step away from the form of a model. Primitives should be the essential constituent parts of the modeling space.

Full dimension The interface should make it easy for the user to observe the object being modeled in the number of dimensions it has. The modeling tool should have easy control over all of its degrees of freedom.

Small toolset The modeling operations should consist of a few tools. If the number of such tools is large, thought must be used to decide which tool to use, instead of contemplating the object being modeled.

Direct creation Requiring an artist to create a simple object and then edit it repeatedly to get the shape they desire forces many unnecessary intermediate perceptions which have little to do with the final product. We seek tools that are capable of sophisticated creation during the early stages of design.

Sensory completeness The visual, haptic, and other senses should be used to better understand the model being created.

2.1 Consider the Pencil

To better understand what a perceptual, creative computer modeling tool might be, consider an exemplary traditional tool: the pencil. The movements of the hand across the page are closely tied to the resulting lines that are displayed. This single tool can be used to make a range of shapes, from simple to intricate. The pencil does not require the user to work within a mathematical structure. The pencil does not force images to be understood through primitives that are not related to the drawing task. The pencil allows highly complex shapes to be directly created without tedious editing. The pencil allows the creation and viewing of drawings in their full dimension. Users are presented with direct control over the lines they are making, and nothing more. The pencil is successful because it allows users to think perceptually about a drawing as it is being constructed.

Even when presented with sophisticated modeling tools, artists often use pencils to think about models before specifying them with software. This process forces a three-dimensional problem to be thought about in two dimensions. Since the object being created is three-dimensional, this thought process would ideally take place in three dimensions. We present such a solution in this paper.

2.2 Related Work

Modeling is one of the oldest problems in computer graphics [22]. Most successful commercial systems for three-dimensional modeling are based on the use of tensor-product spline surfaces [8]. These surfaces have nice curvature properties, but unfortunately require a very complex toolsuite for effective manipulation [2]. Such techniques force artists to think in terms of mathematical structures which they often find counterintuitive. Once such complex systems are in place, it is difficult to make changes to parts of a model without restructuring. Current interfaces for spline-based modelers are typically based around two-dimensional views and two-dimensional input via a mouse. Consequently, the three-dimensional structure of the product being created is not an inherent part of the modeling process.

Another well-studied modeling technique is the use of deformations. Global deformations, introduced by Barr [5], apply functions to deform space, and carry an underlying surface with them. Local deformations, such as the free-form deformations developed by Sederberg and Parry [21] allow users to make small, detailed changes to an object. These techniques are limited because they require users to start with simple shapes and work them into more complex forms. The sequence of changes is dictated by the underlying mathematics and is not an inherent part of the final form.

Alternative methods of surface representation, such as Grimm's manifold work [11], and Szeliski and Tonneson's oriented particles paradigm [23] are promising in their ability to represent a great variety of shapes, but their interactive potential for intricate shapes has not been demonstrated.

Some recent work has focused on different methods of interaction to solve some of the above problems. Zeleznik's SKETCH paradigm [27] utilizes two-dimensional gestural input to describe and place objects in three-dimensional space. While SKETCH makes efficient use of a two-dimensional input device, many of its complicated gesture sequences would not be necessary if the user had a three-dimensional interface. SKETCH allows users to work primarily in three orthogonal directions. While this is useful for the creation of many manufactured parts, SKETCH does not immediately extend to intricate organic forms.

Two-dimensional input is used to create three-dimensional figures in the work of Han [13] and van Overveld [25]. These vision-based techniques try to infer three-dimensional structure from a series of two-dimensional motions, requiring the user to think in two dimensions about a problem that is inherently three-dimensional. They do however provide means to leverage traditional drawing expertise.

The HoloSketch work of Deering [6] and the 3-Draw system of Sachs et al. [19] allow users to work in three-dimensional space, but they allow only the placement of lines and, in Deering's case, primitives such as transformed spheres. These approaches work well for models that are made of theses primitives but do not readily extend to the larger class of all surfaces.

Finally, volume sculpting, as presented by Galyean [10] and Wang [26] comes closest to meeting our guidelines. Their techniques are limited by the choice of initial volume which must contain the finished shape. Shapes must be formed by repeated subtraction steps which are somewhat indirect. In Surface Drawing we instead opt for an additive process, providing for the direct creation of intricate geometry.

Figure 3: *An artist drawing a figure holds a pencil so that the region of contact between the pencil and paper is a line.*

3 Surface Drawing

The goal of the Surface Drawing approach is to extend the traditional system of drawing lines with a pencil to the creation of freeform surfaces. As discussed in Section 2.1 the pencil is an excellent tool for perceptual thinking. The key to understanding a pencil is to analyze the way in which it is used by experts. Figure 3 shows how a trained artist holds a pencil. Note that the contact between the pencil and the paper (two-dimensional space) is in the form of a *line*. Surface Drawing is an extension of this metaphor where the contact between the user and three-dimensional space is a *plane*. There is an interesting connection between this view of drawing and manifold theory (see [1]). Drawing with a pencil is much like adding locally one-dimensional coordinate patches to a one-manifold. Modeling surfaces is the process of making *two*-manifolds, and Surface Drawing does this via the natural extension of adding *two-dimensional coordinate patches*. In short, we view drawing as the development of a manifold with codimension one.

We can summarize our paradigm with the following definition:

Surface Drawing: A method for creating shape in which surfaces are created by moving a locally two-dimensional object through three-dimensional space.

A locally two-dimensional object that is very versatile and intimately entwined with human perception is the inside surface of the hand. Our implementation of Surface Drawing uses the motion of the hand to define portions of surfaces (see Figure 4). Notice that the hand is moving in the tangent plane of the surface being created, much as a pencil moves in the tangent line of a curve drawing. A few hand motions can be combined rapidly to form a much more elaborate surface, such as the guitar depicted in Figure 1.

To realize these ideas, we need an interactive environment which supports the sensing of an oriented plane at a point in three-dimensional space. The traditional mouse/monitor interface

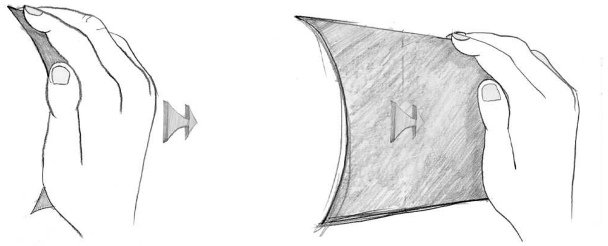

Figure 4: *A simple drawing motion produces a small surface patch.*

scheme would not be fully dimensional. A system which does support five degrees of freedom on input and three-dimensional viewing of objects is the Responsive Workbench [15, 14] (see Figure 2). Its horizontal top surface (measuring 1.8 x 1.35 m) displays alternating left/right stereo images which are viewed through magnetically tracked shutter glasses. The position and orientation of the interface tools (stylus and glove) are also sensed with magnetic trackers. The hand configuration is sensed with a CyberGlove.

In this fully dimensional environment, there is a perceptual connection between the form of the hand as it moves through space and the surface that is generated. This perceptual creation, along with the perceptual viewing of the model, allows shapes to be understood at a deep level while they are created.

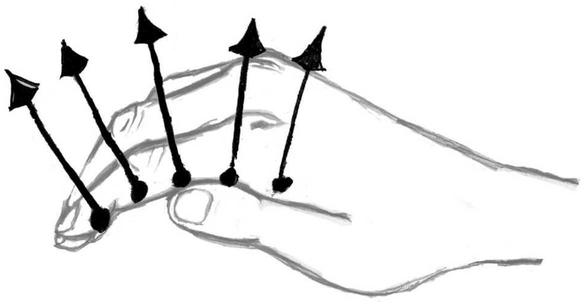

Figure 5: *In the primary drawing mode, we sample position and orientation at five locations along the index finger and palm.*

4 Surface Construction

Data from the tracker and glove is used to create a mesh of triangles that describes the surface being drawn. From the user's perspective, the hand is acting as a guide for a plane. To construct a surface efficiently, we view the hand as a group of samples consisting of position and orientation (see Figure 5 for a depiction of the samples we use). Placing these samples as points in three space without connecting them does not give a strong sense of the underlying surface being formed. We present a method, which we call the Cookie Cutter algorithm, that incrementally constructs a surface from these samples.

This problem is similar to the well-studied surface reconstruction problem (for a review see [18]). Three key differences exist between the traditional reconstruction setting and ours:

- **Input Data:** normal vectors, in addition to positions, are given;
- **Locality:** since the sample points are added interactively, global information about the sample set cannot be used;
- **Speed:** the construction must be performed within tight delay and update limits to sustain the illusion of virtual reality.

Approaches to reconstruction that use three-dimensional triangulation such as Edelsbrunner's alpha shapes [7] and Amenta's crust algorithm [3] suffer from high computational costs. The Cookie

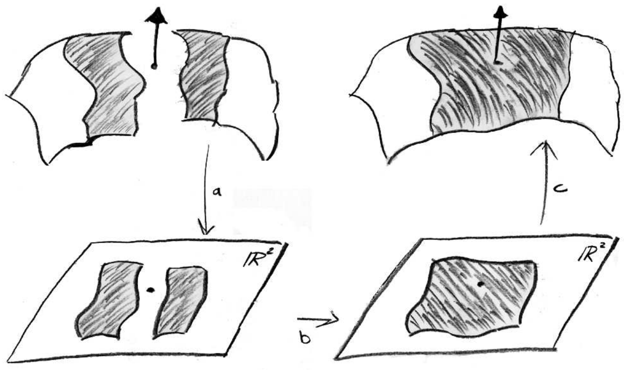

Figure 6: *An overview of the Cookie Cutter algorithm. A surface patch (shaded) near the new sample is identified. Then it is (a) projected onto $\mathbf{R}^2$, (b) retriangulated, and (c) moved back to the original mesh.*

Cutter technique avoids this overhead by retriangulating in two dimensions. An overview of the construction process is shown in Figure 6. To add a new sample to the mesh, the affected surface neighborhood is removed and retriangulated, taking the new data into account. By adding the notion of orientability to our system, this algorithm also allows self-intersecting surfaces of high curvature to be formed.

The following section describes this process in detail.

4.1 The Cookie Cutter Algorithm

The incremental Cookie Cutter algorithm takes an existing mesh of samples, edges, and triangles, $M = (S, E, T)$, and changes it to reflect the addition of a new sample x. A sample is a position in $\mathbf{R}^3$ and a corresponding unit normal vector (direction), which we write $x = (x_p, x_d)$. After receiving x, the Cookie Cutter algorithm performs the following seven steps:

1. **Find neighborhood:** identify a neighborhood of samples, $N_s(x) \subset S$ to which the new sample x should be added;

2. **Find surface region:** the triangles $N_t(x) \subset T$ that correspond to this neighborhood, along with a number of *boundary edges* $N_e(x) \subset E$ describing the boundary of the surface region being removed are identified;

3. **Ensuring non-degenerate projection:** if any of the triangles $N_t(x)$ will be flipped upon projection into the plane defined by x_d, the neighborhood $N_s(x)$ is reduced, and $N_t(x)$ as well as $N_e(x)$ are updated;

4. **Cut:** the triangles $N_t(x)$ are removed from the mesh;

5. **Project:** each sample in $N_s(x)$ and each boundary edge in $N_e(x)$ are projected into the plane defined by x_d;

6. **Triangulate:** the projection of $N_s(x)$ is triangulated using a two-dimensional Delaunay triangulation with some modifications;

7. **Unproject:** each resulting Delaunay triangle is mapped back to $\mathbf{R}^3$ by the implicit association through $N_s(x)$.

Each of these seven steps is now described in detail.

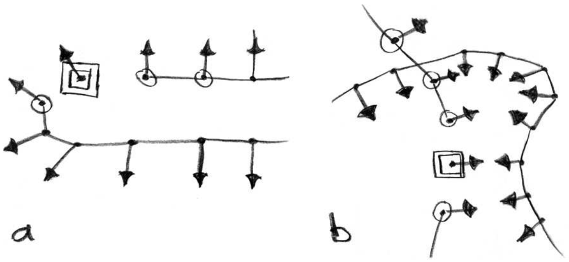

Figure 7: *A neighborhood (circled points) is taken around a new sample (in square). Using a simple dot product test, neighborhoods on (a) surfaces with high curvature and (b) intersecting or almost-touching surfaces can be formed correctly.*

Find neighborhood Given a new sample x, its sample neighborhood $N_s(x)$ consists of all samples s^i that satisfy the following two conditions:

$$\|x_p - s_p^i\| < d_{\max} \text{ and } |x_d \cdot s_d^i| > \cos(\theta_{\max})$$

where $\| \cdot \|$ is the standard Euclidean metric. Samples satisfying these conditions are shown as heavy dots in the example of Figure 8. In practice, we found $d_{\max} = 5cm$ and $\theta_{\max} = 60^o$ to work well. These conditions choose a neighborhood that is near the new sample both in terms of position and orientation. As shown in Figure 7, this mixed distance/orientation criterion allows us to deal effectively with regions of high curvature or multiple surfaces intersecting. We can also handle surface parts with opposite orientations that are arbitrarily close to one another.

Find surface region The triangle neighborhood $N_t(x)$ consists of triangles whose vertices are all in $N_s(x)$. $N_t(x)$ (shaded in Figure 8) represent the set of triangles that might be changed when $N_s(x)$ is retriangulated. While finding $N_t(x)$, an array of signed edges, which we call *boundary edges* (denoted $N_e(x)$), is filled with edges of triangles that contain only two points in $N_s(x)$. Boundary edges are represented as arrows in Figure 8 (a). A boundary edge is a segment of the boundary between the surface neighborhood of x and the remainder of M, and the set of boundary edges around a given sample forms our cookie cutter. We orient each boundary edge using the third vertex of the triangle that was found to contain only two neighborhood samples. This orientation is used to prune triangles formed outside of the boundary in the retriangulation process and to ensure a non-degenerate projection.

Ensure non-degenerate projection In cases of high curvature and sample noise, projection onto the tangent plane of x can flip triangles (see Figure 9). In practice, this situation occurs quite rarely. However, when it does happen, it causes a hole in our mesh. We check for this problem by storing an orientation t_d^i for each triangle in T. Each triangle normal t_d^i is chosen using the orientation of M. If $t_d^i \cdot x_d < 0$ for any triangle in T, $d_{\max}$ is reduced and $N_s(x)$, $N_t(x)$, and $N_e(x)$ are recomputed.

Cut The cutting step removes the triangles in $N_t(x)$ from the mesh M.

Project After correctly calculating a neighborhood, the local surface region (cookie) is ready for retriangulation. Basis vectors on the tangent plane of x are chosen, and each point in $N_s(x)$ is given coordinates in terms of these vectors. The edges $N_e(x)$ that define the boundary are also projected onto the tangent plane at x. The orientation of any boundary edge $e^i \in N_e(x)$ can flip upon projection, in a fashion similar to that of a triangle flipping. A dot product test similar to that for triangles is performed. However, this time test

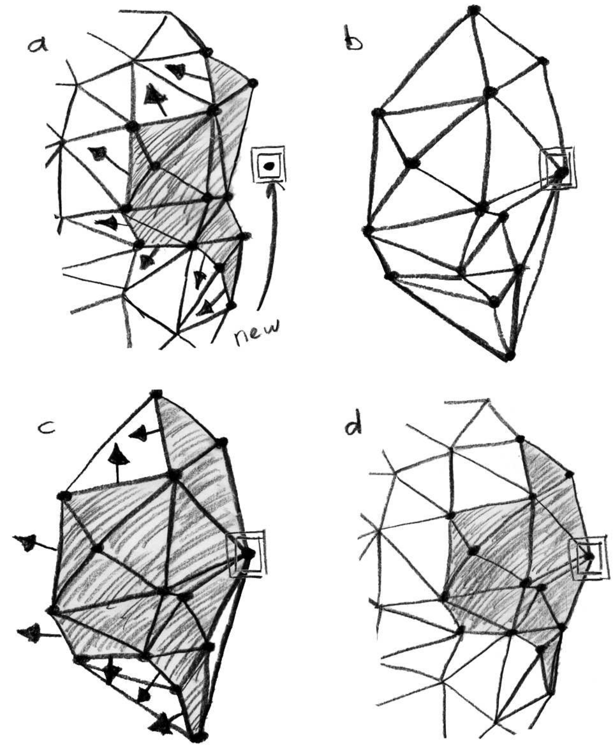

Figure 8: *A portion of the mesh is cut out and replaced by a mesh region that contains the new sample. (a) After the addition of a new sample, the samples in the neighborhood (heavy dots) and boundary edges (with arrows) are identified. The triangles in the neighborhood $N_t(x)$ (shaded) are removed. (b) The sample neighborhood is retriangulated. (c) Triangles (shaded) that are neither on the wrong side of a boundary edge, nor have large circumcircle, are identified. (d) The identified triangles are added to the mesh.*

failure is dealt with by changing the orientation of the associated boundary edge to ensure correct behavior.

Triangulate Figure 8 (b) shows an example of a two-dimensional Delaunay triangulation [16, 12]. Note that it may contain triangles that are outside of a boundary edge (see Figure 8 (c)). These are removed to preserve the cookie cutter outline of the neighborhood that was originally removed from the mesh. After this cleanup the triangles are ready to be reinserted into the original mesh (see Figure 8 (d)). A problem with traditional Delaunay triangulations is

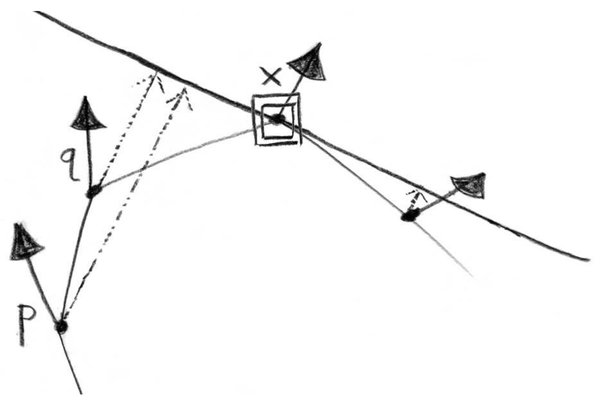

Figure 9: *In this cross-section of a mesh, the triangle containing p and q is flipped when it is projected onto the tangent plane of the new sample x*

that the resulting triangulation is always convex. Users often make planar regions with concave boundaries which should be reflected in the mesh. Removing triangles (such as the triangle in the lower right-hand corner of Figure 8 (c)) whose associated circumcircles have prohibitively large radii (in practice we used 2.5cm) allows the boundaries of meshes to be concave.

Unproject The vertices of triangles in the retriangulation are all associated with samples in the original mesh through the tangent plane projection. We map each of these samples back to their original locations in $\mathbf{R}^3$, moving the new triangulation into the three-dimensional space in which the user is working.

4.2 Discussion

The key elements of the Cookie Cutter approach are sample neighborhood construction and boundary computation. Using orientation information to prune the Euclidean neighborhood allows planar subsets to be found in many difficult regions, such as neighborhoods with intersecting or almost-touching surfaces. Storing a boundary allows a very efficient local retriangulation process. The Delaunay triangulation is $O(n \log n)$ with a typically small n. The other steps are all $O(n)$ with the same small n. The construction of the sample neighborhood $N_s(x)$ is optimized with a regular grid spatial data structure.

Unlike the surface reconstruction setting, where the data can be assumed to come from a clean surface, the Cookie Cutter algorithm has to deal with "messy" samples. This is due to the inherent noise in the tracking system and the unsteadiness of users' hands in this environment. While the triangulation algorithm is provably correct in two dimensions (see [20]), this proof does not extend to the three-dimensional case. Despite these issues, the algorithm manages to form clean surfaces the majority of the time. Holes and nonmanifold topology occur about one percent of the time (one bad vertex per 100 vertices). While these situations can be easily fixed by a user, automatic mending will be treated in future research. Noise also causes the meshes to be quite bumpy at times. Thus, we apply a simple Laplacian mesh smoother [24] as a post-process.

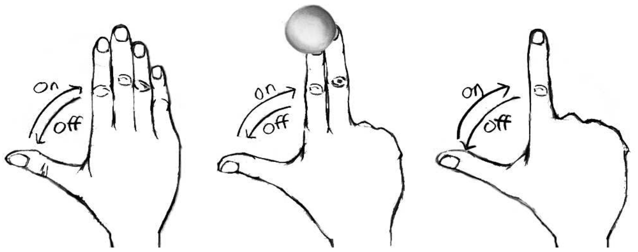

Figure 10: *Three hand positions are identified: (a) the user draws with the index finger and palm; (b) an eraser is placed at the end of the index and middle fingers; (c) fine details are added with the tip of the finger. In all three situations, the thumb is used to activate and deactivate the operation.*

5 Implementation

The interface to our Surface Drawing implementation consists of two methods for adding geometry, an eraser to remove geometry, and a simple manipulation tool. The three tools that affect geometry are accessed with three hand configurations (see Figure 10). The manipulation tool is assigned to a magnetically tracked stylus.

Primary input The choice of hand samples is important in allowing geometry to be placed effectively. We sample the index finger and the palm of the inside surface of the hand (see Figure 5). Normals are taken as the normals of the outer surface of the hand. The user moves the hand as if running it across the object being created, in a direction perpendicular to the plane along which the finger bends (see Figure 4). This allows the description of surfaces with a considerable range of curvature. With this system, drawing can be easily started and stopped by pressing the thumb against the index finger (see Figure 10 (a)). Since the elastic tension in the CyberGlove naturally holds the thumb in the "off" position, users rarely draw accidentally. We find that users learn this on/off mechanism quite readily.

Eraser Users often want to correct or remove regions of a surface. Rather than introducing complex editing semantics, we provide the user with only a simple spherical eraser tool. It is moved through space, deleting geometry in its wake. We remove all the samples and their associated triangles that are within the sphere of the eraser. The user switches from draw mode to erase mode by bending the metacarpophalangeal joints on the 4th and 5th fingers while keeping the other fingers straight (see Figure 10 (b)). Erasing is also activated pressing the thumb against the index finger. Since geometry can be added with ease, this simple means of editing is sufficient.

Manipulation The user frequently needs to work on the model from different views, requiring the ability to translate and rotate. In a 3D semi-immersive environment it is quite natural to accomplish this with a virtual stick which can be "poked" into the object. The object then follows the position and orientation of the stick, providing a direct manipulation ability. In our implementation this was accomplished by providing a tracked stylus, which the user holds in the subdominant hand while creating geometry with the dominant hand.

Detail input Users often want to add small details to their models. Our current system allows users to add small pieces of surface with the tip of the index finger. This mode is accessed by bending the 3rd, 4th, and 5th fingers while keeping the index finger straight (see Figure 10(c)). We sample the hand as before but only use the samples from the distal phalanx.

5.1 Discussion

In our experiments we found that this interface was learned quite readily. One of the nice features of this system is that there is no stored interface state. The hand configuration affects how the drawing happens, and the user is never asked to perform a mode switching action requiring a movement away from the object. The manipulation task, which does not require high precision, is assigned to the subdominant hand. Moving geometry is necessary for both drawing and erasing, and this two-handed interface ensures that the stick is always accessible.

6 Results

The prototype system presented in this paper has been used by four practicing artists and designers, ranging from computer modeling experts to fine artists with no prior exposure to semi-immersive technologies. In this section we present some of the models created by them and discuss observations made during the use of the Surface Drawing tool.

The iris in Figure 12 demonstrates the control of surface form and flow that Surface Drawing yields. The delicate bend of the petals was under complete control of the artist at every point during the creation process. The basic form was created in half an

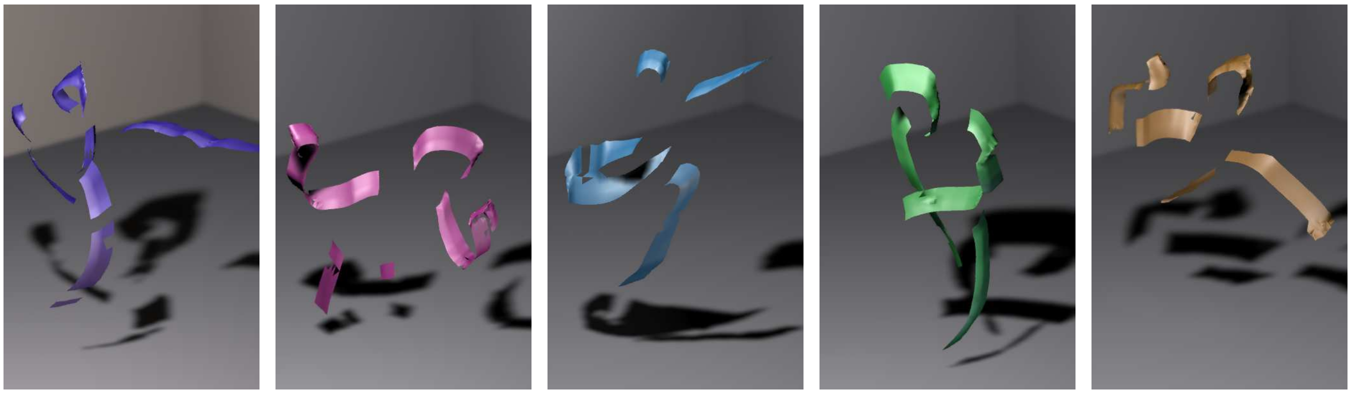

Figure 11: *A series of three-dimensional gestures, each drawn in less than one minute*

hour. The final shape was acheived through a series of localized refinements. A similar model could have been created with a more traditional NURBS modeling tool, but the process would require initial planning of the layout of patches. Surface Drawing can be started and finished without such planning. After placing the general spline patches, a NURBS user would then have to push and pull control vertices (often with unexpected results [9]) to achieve the final form. The final Surface Drawing was refined through direct, predictable manipulations by the hand.

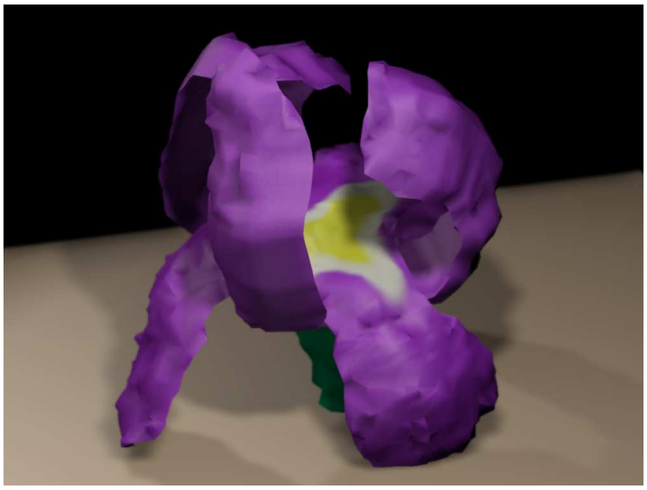

Figure 12: *An articulated iris.*

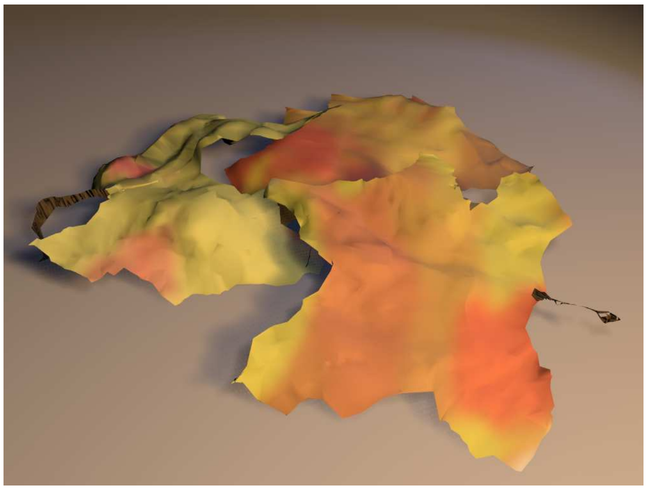

Figure 13: *Fallen leaves.*

The leaves in Figure 13 are another example of an intricately formed surface whose undulations were intuitively grasped during the creative process.

Both of these works began as rough drawings. Portions of the figure were erased and recreated to accentuate the desired affect. The eraser allows partial removal followed by reworking.

The character of the leaning bird in Figure 16 emerged during the modeling process. The model is leaning in an exaggerated fashion. This pose was not initially planned, but rather was the result of an insight that occured during the modeling process. Because the artist was working intimately with the shape, this way of representing a graceful and playful character emerged. The model was conceptualized and Surface Drawn in 45 minutes.

The aesthetic subtlety of the wings of the soaring bird in Figure 16 would be quite difficult to plan and specify in advance. Surface Drawing allowed these wings to be created with the natural language of their motion. The graceful emotional nature of these models demonstrates the potential of Surface Drawing for three-dimensional character design.

Surface Drawing is a powerful tool for capturing gestural form, as indicated by the three-dimensional gesture drawings shown in Figure 11. These figures were drawn by an artist viewing a live model. Each drawing took one minute to complete. These works show the strength of Surface Drawing in capturing emotive qualities in a manner similar to that of line gesture drawing. The short modeling time of these figures demonstrates that Surface Drawing can easily be used as a rapid conceptualization tool.

The furniture in Figures 14 and 15 was made by a designer seeking to explore new furniture ideas. The gentle curve of the chair's seat was modified until the designer found the desired balance. The creation of the sofa was a somewhat more time consuming process. These shapes demonstrate how a considerable number of different forms are tied together in each piece in a three dimensional harmony. This complex relation was felt and understood easily in the Surface Drawing framework since the creative process was not slowed by the need to manipulate complex tools or understand the nuances of an underlying mathematical machinery.

Finally, the human torso in Figure 16, created by an artist with two weeks of Surface Drawing experience, is the most intricate Surface Drawing done to date. After performing some gestural sketches, the artist drew a torso from a live model. Once the basic gestural form was Surface Drawn to the artist's liking, he further refined the model by adding musculature.

7 Discussion

Our experience to date with Surface Drawing has demonstrated that this paradigm is easily understood by artists and non-artists alike.

Figure 14: *Stretched metal chair.*

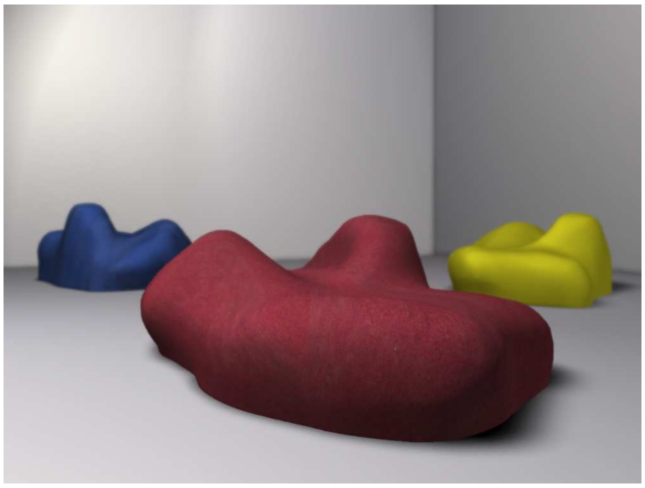

Figure 15: *Fuzz bubble sofa.*

We found that the tool was quickly comprehended because it is inherently related to fundamental perceptual abilities.

In the following paragraphs we revisit the modeling guidelines we laid out at the beginning of this paper and discuss how well the current implementation of Surface Drawing follows them.

Invisible mathematical structure There is no underlying structure on our mesh. The user does not have to think about coordinate systems or control vertices. The only structure that is present is orientation. With tracker noise and a modest sampling rate, this is necessary to enable surfaces to touch without merging. Shading the two sides of surfaces differently gives a direct perceptual cue to this orientation, and we found that the concept was understood by users.

Direct primitives No predefined surfaces are used as starting points for modeling, nor are they combined to form more complex objects. The primitive in this approach, from the user's point of view, is a small locally two-dimensional patch which, as manifold theory indicates, is the elemental component of a surface.

Full dimensions The tracked hand moving in space together with the stereo view give our implementation full dimension.

Small toolset Our entire interface for manipulating and creating geometry consists of four modes/tools: primary input, erasing, manipulation, and detail input. These modes are easily accessible via a change in hand position, and do not require any operations, such as selecting a menu, before a tool can be used.

Direct creation Our system provides a highly versatile method for creating form. Changing models during the early stages of ideation is more a process of creation than editing. In our system there is essentially no dependence of the final shape on the history of its creation, fully supporting users changing their minds as they model.

Sensory completeness The use of the hand allows the sense of proprioception to be used. The visual image of the model matches the positions of the hand, forming a sensory loop. However, since the hand moves through empty space there is no haptic feedback. The latter is an integral part of sketching on paper. Current force feedback technology is much too constraining for the Surface Drawing environment but will be of great interest as it matures.

8 Summary and Future Work

In this paper we have presented the idea of perceptual thinking and proposed a number of principles which reflect this idea in the context of three dimensional modeling. Surface Drawing is a method which satisfies these principles, and we described an implementation using a semi-immersive environment to achieve the full dimension requirement. The surfaces are realized as meshes which are built incrementally with the help of the Cookie Cutter algorithm.

We have begun the exploration of the potential of Surface Drawing with our prototype implementation and a small group of artists and designers as users. Many of the underlying ideas were developed directly out of the experience of using the system. There are many possible directions for future work suggested by these explorations.

- **User interface:** So far we have deliberately focused on using the smallest number of tools possible to create interesting shapes. As our users build more shapes they have expressed a desire to integrate other facilities, e.g., grouping and replication, into the Surface Drawing environment. Future user interface work will address the range of capabilities accessible through the glove, while keeping the cognitive load low.
- **Multiple resolutions:** Many objects have features at multiple resolutions. To more effectively support creation of such objects the Cookie Cutter algorithm will be modified to continually adapt the sampling density of the surface through local subdivision or mesh simplification.
- **Topological inconsistencies:** Occasionally, topological inconsistencies are created in the mesh. These can be troublesome when using Surface Drawn geometry in other systems. We have found some heuristics to work well without limiting the user's perceived freedom of creation, and we will pursue more work in this direction.
- **Cookie Cutter algorithm:** One can think of the Cookie Cutter algorithm as an online reconstruction algorithm for unknown and changing geometry. Developing this potential further could lead to applications in surface reconstruction settings such as those presented by range sensing devices and contact digitization tools.

The original goal of Surface Drawing was to enable rough sketches of shapes to be made in the exploratory phase. However, the results lead us to believe that this tool is capable of much more. We look forward to the future development of this paradigm.

9 Acknowledgments

This work would not have been possible without many members of the Caltech community, most notably David Kremers, who drew the gestures in Figure 11, the iris in Figure 12, the leaves in Figure 13 and the torso and soaring bird in Figure 16. Kremers also provided many suggestions, including the use of the finger to add detail. Khrysaundt Koenig was an invaluable aid in the preparation of this document, most notably in the lighting and rendering of the gestures, iris, leaves, chair and sofa in Figures 11- 15. She also drew all of the human hands which appear throughout this report. Everett Kane, of Art Center College of Design, produced suggestions and constructive criticism in the early stages of the work. Mathieu Desbrun, Dan Fain, Zoë Wood, Al Barr, and Jim Barry graciously provided long conversations which guided the development of this project.

Support for this work was provided by generous industrial gifts of DesignWorks/USA and Alias|Wavefront, and research funding from NSF (ACI-9721349), AFOSR (F49620-96-1-0471), ONR (N00014-97-1-0387), and DOE (W-7405-ENG-48).

References

[1] ABRAHAM, R., MARSDEN, J., AND RATIU, T. *Manifolds, Tensor Analysis, and Applications*. Springer-Verlag, 1988.

[2] ALIAS|WAVEFRONT. Studio 9, 1998.

[3] AMENTA, N., BERN, M., AND KAMVYSSELIS, M. A new voronoi-based surface reconstruction algorithm. In *Computer Graphics (SIGGRAPH '98 Proceedings)* (July 1998), pp. 415–422.

[4] ARNHEIM, R. *Visual Thinking*. University of California Press, 1969.

[5] BARR, A. H. Global and local deformations of solid primitives. In *Computer Graphics (SIGGRAPH '84 Proceedings)* (July 1984), pp. 21–30.

[6] DEERING, M. F. Holosketch: A virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction 2*, 3 (September 1995), 220–238.

[7] EDELSBRUNNER, H., AND MUCKE, E. P. Three-dimensional alpha shapes. *1992 Workshop on Volume Visualization* (1992), 75–82.

[8] FARIN, G. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1997.

[9] FIZTGERALD, G. Design Specialist, DesignWorks/USA, Newbury Park, CA; personal communication, Fall 1997.

[10] GALYEAN, T. A., AND HUGHES, J. F. Sculpting: An interactive volumetric modeling technique. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (July 1991), pp. 267–274.

[11] GRIMM, C. M., AND HUGHES, J. F. Modeling surfaces of arbitrary topology using manifolds. In *Computer Graphics (SIGGRAPH '95 Proceedings)* (August 1995), pp. 359–368.

[12] GUIBAS, L., AND STOLFI, J. Primitives for the manipulation of general subdivisions and computation of voronoi diagrams. *ACM Transactions on Graphics 4*, 2 (Apr. 1985), 74–123.

[13] HAN, S., AND MEDIONI, G. I3dsketch: Modeling by digitizing with a smart 3d pen. In *ACM Multimedia 1997* (1997), Addison Wesley, pp. 41–49.

[14] KRÜGER, W., BOHN, C.-A., FRÖHLICH, B., SCHÜTH, H., STRAUSS, W., AND WESCHE, G. The responsive workbench: A virtual work environment. *IEEE Computer 28*, 7 (July 1995), 42–48.

[15] KRÜGER, W., AND FRÖHLICH, B. The responsive workbench. *IEEE Computer Graphics and Applications* (May 1994), 12–15.

[16] LISCHINSKI, D. Incremental delaunay triangulation. In *Graphics Gems IV*, P. S. Heckbert, Ed. Academic Press Professional, San Diego, CA, 1994, pp. 47–59.

[17] MCKIM, R. H. *Thinking Visually*. Dale Seymour Publications, 1980.

[18] MENCL, R., AND MÜLLER, H. Interpolation and approximation of surfaces from three-dimensional scattered data points. Tech. Rep. 662, Universität Dortmund, December 1997.

[19] SACHS, E., ROBERTS, A., AND STOOPS, D. 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications 11*, 6 (Nov. 1991), 18–26.

[20] SCHKOLNE, S. Surface drawing. Master's thesis, Caltech, 1999. forthcoming.

[21] SEDERBERG, T. W., AND PARRY, S. R. Free-form deformation of solid geometric models. In *Computer Graphics (SIGGRAPH '86 Proceedings)* (Aug. 1986), pp. 151–160.

[22] SUTHERLAND, I. E. Sketchpad – A man-machine graphical communication system. In *Proceedings of the Spring Joint Computer Conference* (May 1963).

[23] SZELISKI, R., AND TONNESEN, D. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (July 1992), pp. 185–194.

[24] TAUBIN, G. A signal processing approach to fair surface design. In *Computer Graphics (SIGGRAPH' 95 Proceedings)* (Aug. 1995), pp. 351–358.

[25] VAN OVERVELD, C. W. A. M. Painting gradients: Free-form surface design using shading patterns. In *Graphics Interface '96* (May 1996), pp. 151–158.

[26] WANG, S. W., AND KAUFMAN, A. E. Volume sculpting. In *1995 Symposium on Interactive 3D Graphics* (Apr. 1995), pp. 151–156.

[27] ZELEZNIK, R. C., HERNDON, K. P., AND HUGHES, J. F. SKETCH: An interface for sketching 3D scenes. In *Computer Graphics (SIGGRAPH '96 Proceedings)* (Aug. 1996), pp. 163–170.

Figure 16: *Three intricate models: soaring bird, human torso, and a whimsical, leaning bird.*