Computational Aesthetics in Graphics, Visualization and Imaging (2005)
L. Neumann, M. Sbert, B. Gooch, W. Purgathofer (Editors)

Sketch Based Construction and Rendering of Implicit Models

B. Wyvill, K. Foster, P. Jepp, R. Schmidt, M. C. Sousa and J. A. Jorge[2]

Department of Computer Science, University of Calgary, Canada
[2] Departamento de Engenharia Informática, Instituto Superior Técnico, Portugal

Abstract
We present an implicit modeling system as a tool for creating a wide range of aesthetic models. Because of their ability to form blends and produce both organic shapes as well as man-made objects, implicit surfaces are a good medium for artists seeking new ways to experiment with 3D modeling. Implicit models can be created using our sketch-based modeling tool Shapeshop *and also by using a procedural interface. Further, we exploit the differential properties of implicit surfaces to explore new techniques for rendering hierarchical, skeletal implicit models in several pen and ink styles. Our method extracts and stylizes silhouette strokes, lines following local shape features, such as those caused by CSG junctions and abrupt blends, and short interior marks to reveal basic form. In this approach we use a particle system as a basis for the stroke extraction method.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation, I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations

1. Introduction

The aim of this research is to provide artists and designers with new tools for building and rendering interesting models in an artistic fashion. In this paper we describe some of the interactive and procedural methods we have developed to provide these tools in an implicit modeling system. Our approach is based on a hierarchical implicit modeling system, the *BlobTree* [WGG99]. Models can be built interactively (Shapeshop), or with the Python language to enable the manipulation of a wide variety of implicit primitives. The nodes in the *BlobTree* represent a large number of operations including different types of blends, CSG operations and warps. Besides conventional rendering techniques, we have also designed algorithms for portraying features of complex hierarchical implicit models as strokes, which produces an aesthetic rendering of those models. NPR rendering can be used to produce pleasing images, and the system includes a number of parameters to allow fine control over the resulting image. We take advantage of functionally defined implicit surfaces to produce pen and ink style renderings. The renderer's view is very general since it uses black box functions to decide on the placement of strokes in regions representing surface features. Furthermore the type of each stroke can be guided by the surface geometry. The black box approach enables the development of the rendering software to be independent of the *BlobTree* , making it possible to use other implicit formulations. Implicit surfaces enjoy many advantages over other modeling techniques, particularly in applications requiring a wide range of topologies and blends. Indeed, we exploit these features in artistic rendering. In fact the *BlobTree* has a number of features that make it very useful for creating aesthetic models as detailed below. Implicit modeling techniques lend themselves to representing both man made (figure 1) and natural objects (figure 4) and with pen and ink rendering styles they provide artists with useful tools for producing aesthetically pleasing models and images.

The methods presented in this research allow users to exploit the hierarchical features of the *BlobTree* to encode model complexity. They can construct a *BlobTree* from predefined parts, procedurally defined models via a Python interface or models generated from sketch input. Unlike other sketch–based approaches, our models can be separated into animatable parts as the *BlobTree* is also a scene graph [WGG99].

The benefits of the system are that through the interactive interface we provide artists with a means to exploit blending and other features of implicit modeling. Moreover, the procedural interface allows users to create aesthetic models and

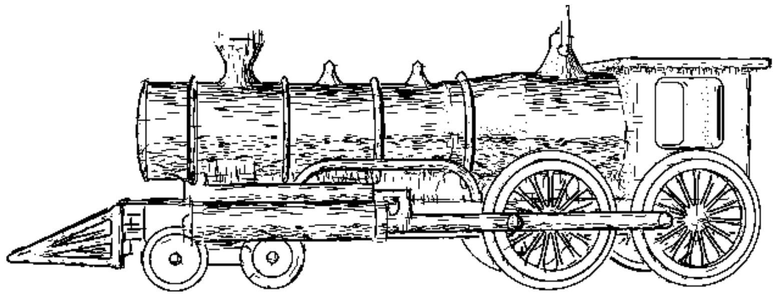

Figure 1: *Pen and ink rendering of a model steam engine.*

animations algorithmically. In addition, pen and ink style rendering is provided alongside photorealistic rendering, providing artists with useful tools for producing aesthetically pleasing models and images.

In the following sections we describe some of the main contributions that inspired this work, some details of the *BlobTree* , the sketch-based modeling interface, and the pen and ink rendering system. We conclude with examples and a brief discussion.

2. Previous Work

In one of the earliest works on implicit surfaces, Ricci used line drawings to visualize constructive solid models [Ric73]. His system extracts lines following the intersection of planar cross-sections of the surface and also performs hidden line removal (HLR). Bremer and Hughes [BH98] and Akleman [Akl98a, Akl98b] presented approaches for artistic stroke extraction from implicit surfaces by stepping along the surface in various directions. The systems differ in that Bremer and Hughes' method renders silhouette strokes in a pen and ink style and Akleman's approach extracts strokes in arbitrary directions and simulates paint interacting with paper to create expressive painterly renderings.

None of these approaches trace strokes by following important shape features such as those resulting from constructive solid geometry (CSG) operations. Furthermore, with the exception of Akleman [Akl98a, Akl98b], these methods are presented only for simple blending surfaces.

Traditional 3D modeling interfaces have a steep learning curve and the standard interaction metaphors are closely tied to the underlying mathematical shape representations. Users must manipulate 3D shapes using abstract concepts such as control points. Sketch-Based Modeling interfaces attempt to simplify 3D modeling by replacing these unnatural interaction techniques with direct controls based on 2D gestures. The goal is to provide artists and designers with 3D modeling tools that are as flexible and efficient as a pencil and paper, and hence more practical for experimentation in early design stages.

The SKETCH [ZHH96] system is an early example of sketch-based modeling. 3D shapes could be created by sketching 2D gestures that were automatically recognized. For example, drawing three perpendicular lines generated a cube. These types of controls have been further explored in the Chateau [IH01] and GiDES++ [JSC03] systems.

Another category of sketch-based modeling research follows the footsteps of the Teddy [IMT99] system. In this system the user sketches a 2D outline which is *inflated* into a rounded 3D shape. Sketch-based editing operations such as cutting and extrusion allows the user to quickly create simple 3D shapes. The underlying shape representation in Teddy is a triangle mesh. This has been extended to employ quadratic implicit surface approximation [IH03] combined with subdivision, resulting in smoother surfaces. A variety of other implicit representations have been applied, including variational implicit surfaces [KHR02,AJ03], convolution surfaces [TZF04], and spherical implicit functions [AGB04]. These systems largely mimic the Teddy interface and are capable of smoothly blending between surfaces, although they do not scale to complex models. Other systems based on interpolating parametric surfaces [CSSJ05] and volume datasets [ONNI03] have explored additional types of sketch-based shape creation and interaction techniques.

3. The *BlobTree*

The *BlobTree* [WGG99] paradigm has been introduced as a method of organizing implicit surface modeling in a manner that enables global and local operations to be exploited in a general and intuitive fashion. In the BlobTree, an implicit surface model is defined using a tree data structure that combines implicit model primitives as leaf nodes, and arbitrary operations as interior nodes. Evaluation of the potential function is then obtained by traversing the tree structure. Currently supported interior nodes include blending, controlled blending, bounded blending, constructive solid geometry (CSG), precise contact modeling (PCM) [Gas93] and spatial warping.

Previous attempts at interactive shape modeling with implicit surfaces have largely been limited to offset surfaces.

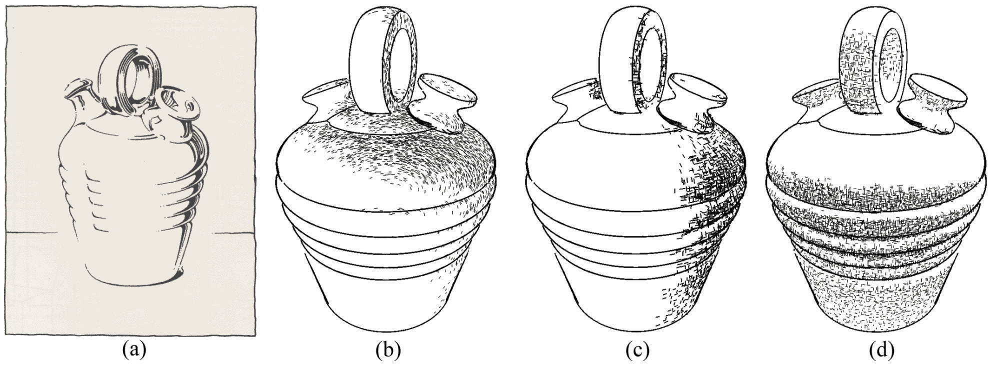

Figure 2: *The jug. (a) An illustration from* Rendering In Pen and Ink *by Arthur L Guptill (see figure 58 on page 36 of [Gup76]). (b-d) Renderings from our system using various stroke directions and a moving point light with strokes in the (b) contour direction and the (c, d) principal directions of curvature.*

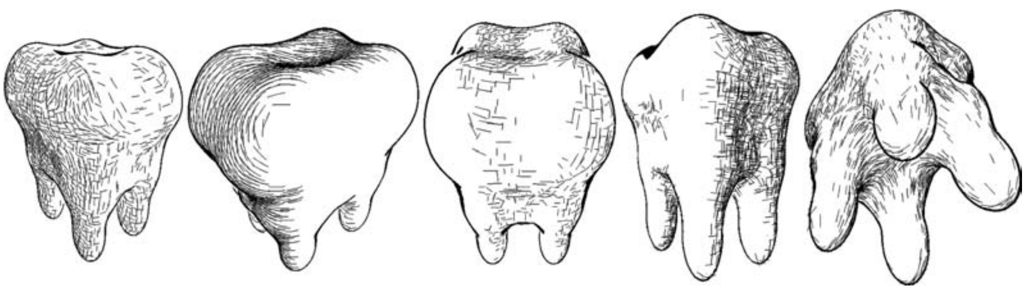

Figure 3: *Five angles of a tooth model rendered using various stroke direction and placement techniques.*

Essentially the user was forced to manipulate some underlying primitive, using trial-and-error to approach the desired surface. This modeling workflow could be frustrating. We have recently developed an implicit representation of sweep surfaces that permits direct specification of the surface. Sweep representations have been shown to be a powerful and expressive shape modeling tool in existing parametric modeling software. Our implicit formulation inherently produces a closed volume and handles self-intersection in a coherent manner. In addition, we have developed several sweep-endcap techniques that increase the expressive potential of these surfaces. By permitting direct manipulation of the sweep profile, we enable the designer to quickly specify the shape of desired parts of a model. These parts can be quickly assembled using blending and CSG operators. The underlying BlobTree model inherently supports a construction history and provides an avenue for animating the models.

For the techniques described in this paper, the potential function is treated as a *black box*. This means that the method is general and can apply to any implicit model definition where the gradient is computable everywhere (although it does not have to be continuous). The stroke extraction methods provided in this paper rely on the vector field created by the gradient $\nabla f(\mathbf{x})$ of the implicit surface and on field evaluations of $f(\mathbf{x})$ for a surface with implicit value *iso*. The gradient of an implicit surface extends everywhere $f(\mathbf{x}) > 0$. When used exactly on the surface, the gradient is perpendicular to the surface. For a complex object, away from the isosurface, the gradient may not point directly to the surface, however the direction is generally acceptable for our particle and stroke extraction methods that follow.

4. Sketch-Based Modeling with the BlobTree

Two-dimensional sketches form the basis for 3D shape creation in ShapeShop. Based on the input mouse samples, we fit a smooth 2D variational contour. A variety of 3D shapes can be generated from these closed curves.

4.1. Creating Shapes from Sketches

The ShapeShop system (see [SWCSJ05]) supports three basic types - “blobby” shapes, linear sweeps, and surfaces of revolution.

To create a “blobby” shape, the closed 2D contour (figure 5a) is projected onto a plane through the origin parallel to the current view plane, and then inflated in both directions

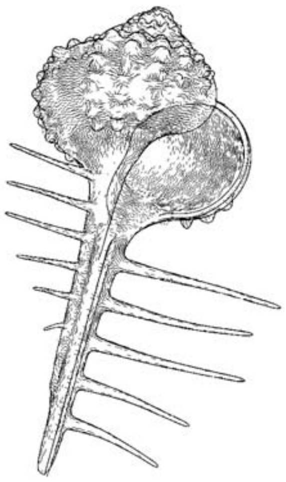

Figure 4: *An example of a complex natural object (Murex Cabritii sea shell) modelled with our system and rendered in pen and ink.*

(figure 5b). After creation, the width of the primitive can be manipulated interactively (figure 5c). The inflation width is functionally defined and could be manipulated to provide a larger difference between thick and thin sections. One advantage of an implicit representation is that holes and disjoint pieces can be handled transparently.

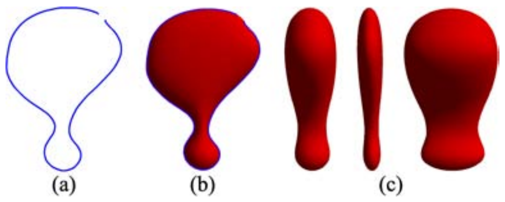

Figure 5: *Blobby inflation converts the 2D sketch shown in (a) into the 3D surface (b) such that the 2D sketch lies on the 3D silhouette. The width of the inflated surface can be manipulated interactively, shown in (c).*

Linear sweeps (figure 6a) are created in the same way as blobby shapes, with the sweep axis perpendicular to the view-parallel plane. The initial length of the sweep is proportional to the screen area covered by the bounding box of the 2D curve, but can be interactively manipulated. Surfaces of revolution (figure 6b) are created by revolving the sketch around an axis lying in the view-parallel plane. Revolutions with both spherical and toroidal topology can be created.

Recent sketch-based modeling systems (see [AJ03]) have largely ignored traditional solid modeling primitives such as linear sweeps and surfaces of revolution, focusing on blobby-inflation primitives instead. This restricts the types of models that can be sketched. For example, CAD models often involve many shapes that are easily represented with sweep surfaces but would be difficult to create using blobby inflation. In particular, surfaces of revolution cannot be reproduced with blobby inflation.

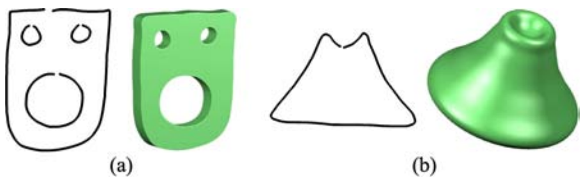

Figure 6: *Sketched 2D curves can also be used to create (a) linear sweeps and (b) surfaces of revolution.*

4.2. Shape Editing

Our underlying shape representation is a true volume model, thus cutting operations can be easily implemented using CSG operators (figure 7). Users can either cut a hole through the object or remove volume by cutting across the object silhouette. Once a hole is created users may transform it interactively, and manipulate properties such as the depth of a cutting operation. Cut regions are represented internally as linear sweeps, so no additional implementation is necessary to support cutting in the BlobTree.

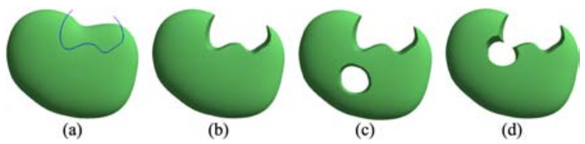

Figure 7: *Cutting can be performed (b) across the object silhouette or (c) through the object interior. Holes can be interactively translated and rotated. Intersection with other holes is automatically handled, as shown in (d).*

We allow users to blend new blobby primitives to the current volume via oversketching. To this end, the corresponding operator has a user-controlled parameter that controls the amount of blending. Furthermore, blended volumes can be transformed interactively (figure 8). Also, interactive blending allows complex shapes to be sketched incrementally using simpler pieces which are merged into a coherent smooth surface.

4.3. Surface detailing

Any BlobTree primitive can be used to add surface detail based on sketches. In an initial experiment, ShapeShop supports "surface-drawing". Rays through the 2D sketch are intersected with the current implicit volume, and the computer intersection points are used to define new implicit primitives that lie on the surface. Currently we create a tube-like surface which is blended to the existing volume (figure 9).

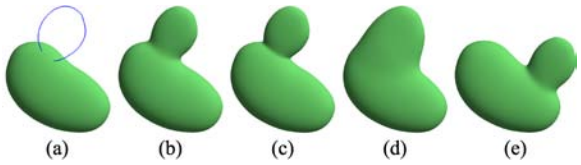

Figure 8: *The sketch-based blending operation (a) creates a new blobby inflation primitive (b) and blends it to the current volume. The blending strength is parameterized and can be interactively manipulated, the extreme settings are shown in (c) and (d). The blend region is recomputed automatically when the blended primitives move, as shown in (e).*

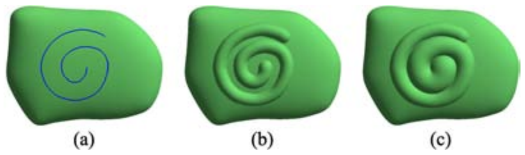

Figure 9: *Surface-drawing is specified by a 2D sketch, as shown in (a). Blended skeletal implicit point primitives are placed along the line at intersection points with the model, shown in (b). In (c) the radius of the points is increased and then tapered along the length of the 2D curve.*

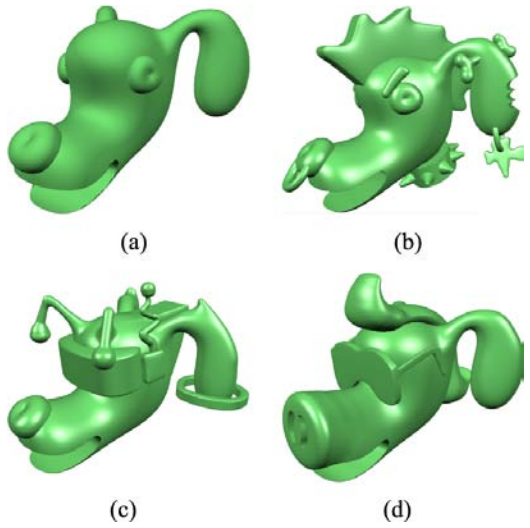

Figure 10: *Design variations on a basic character model (a). Hierarchical implicit volume modeling makes model refinement quick and fluid. For example, the pig nose in (d) was simply sketched over top of the existing dog nose, and then the new volume was blended to the old. The user is not required to deal with geometry issues. Blended features such as "spock ear" points (c), or CSG bite-mark in ear (b), can be easily removed to return to basic ear.*

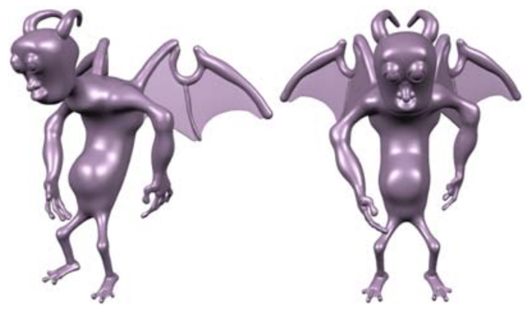

Figure 11: *Gremlin model created using 64 primitives and 38 composition operators. Model components were sketched individually and then assembled. Implicit blending operators support easy experimentation with assembly of different parts. The user can mix-and-match different model components with little effort.*

Surface drawing with implicit volumes is a very flexible technique (figure 10). Any pair of implicit primitive and composition operator can be used as a type of "brush" to add detail to the current surface. For example, creases could be created by subtracting swept cone primitives by using CSG operations. In addition, since each surface-drawing stroke is represented independently in the BlobTree hierarchy, individual surface details can be modified or removed through our existing modeling interface.

4.4. Sketch Interface

Our sketch-based modeling interface was designed primarily to work on large interactive displays, such as the touch-sensitive SmartBoard. These input systems lack any sort of modal switch (buttons). In some sense, this is desirable as pencils also lack buttons, however tasks commonly initiated with mode-switching (such as keypresses or right mouse buttons) must be converted to alternate schemes, such as gestures or 2D widgets.

Since many 2D widgets can be cumbersome to use with large-display input devices (which frequently exhibit low accuracy and high latency), we borrowed stroke-based widget interaction techniques from CrossY [AG04].

5. Rendering *BlobTree* Models in a Pen and Ink Style

BlobTree models can be rendered directly by ray tracing or polygonization [WGG99]. A method for rapid visualization of implicit surfaces was presented by Witkin and Heckbert in [WH94], in which oriented particles are placed on the implicit surface. The particles are forced to lie on the surface and are distributed using an attraction-repulsion method. We have extended this method so that as the particles move, the system can identify whether they are on, or close to, an area of interest. Areas of interest are classified in terms of a pen and ink drawing style; in this context these are outlines

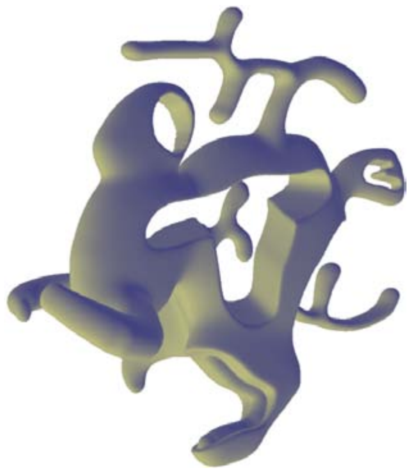

Figure 12: *This "3D Doodle" was sketched absent-mindedly by one of the authors after looking through images of Salvador Dali paintings. Total sketching time was only a few minutes. Experimental 3D modeling at this speed is essentially impossible in existing commercial tools.*

such as silhouettes, contours and discontinuous regions. After identification of an area of interest, outlines and feature strokes are extracted using techniques based on the work of Bremer and Hughes [BH98]. Shape measures are then used to position and stylize other strokes on the interior of the surface. Strokes can be created in many styles with two hidden line removal methods, each entailing specific tradeoffs.

5.1. The particle system

Stroke position is determined using particles placed on the implicit surface. Random rays are created and intersected with the surface to initialize a predefined number of particles n on the surface, hereafter termed $P_i, i \in [1,n]$, with corresponding positions $\mathbf{x}_i$. As ray-intersection tests are computationally expensive, our method only performs this step once during preprocessing. Particles are then distributed over the surface using an attractor/repulser method. The attractive force $\bar{F_i}$ pulls particles toward the surface, and the repulser force $\bar{R_i}$, repels particles from other particles. The Witkin-Heckbert [WH94] particle system starts with a few seed particles ("ideally, only one per connected component"), that divide to create new particles. Particles are spread across the surface using attraction and repulsion forces until reaching a desired sampling level/distribution.

5.2. Extraction of outlines and feature strokes

Points on the silhouette can be identified by comparing the direction of the surface normal $\vec{n}$ at that point, with the viewing direction $\vec{d}$. If $\vec{n}.\vec{d} = 0$ (or within some threshold) then the particle is on (or close to) the silhouette or contour. This avoids resampling the surface with ray tracing and the numerical integration required to move the particle to the silhouette or contour as with the Bremer-Hughes method. Extraction is started from points identified near the silhouette or contour. The extraction method is based on that of Bremer-Hughes where numerical integration is used to trace the silhouette in the direction of $\vec{n} \times \vec{d}$.

Regions of discontinuity or abrupt change must be found using different techniques. Identification of a region of discontinuity can be calculated directly from the implicit functions. If the function definitions are not available for direct computation, other methods must be used to approximate the discontinuity. We choose not to use the underlying function definition to determine if these points do indeed straddle a discontinuity; by doing this we separate the rendering engine from the implicit modeling system. This means that we can use any implicit modeling system that will return a field function value and a gradient for any point in space. Unfortunately, this black box approach does not distinguish between CSG junctions and abrupt blends. However, this is acceptable because all silhouettes and feature lines are rendered in the same style.

As particles move on the surface the current surface normal is compared with the normal at the previous position. If this comparison indicates a large change in the direction of the normal, the region is identified for further investigation. This means that in the region between the particle's current and last positions there is either a discontinuity or an abrupt change. The singularity is approximated by iteratively converging, using the mid-point and normal information, from points straddling a discontinuous region or where there is an abrupt change as detailed in [FJW*05].

Discontinuities and abrupt blends are extracted using a numerical integration technique inspired by that of Bremer-Hughes. The particles identified as straddling the region are required to trace this outline. Using the cross product of the particles' surface normals, a direction vector can be obtained that approximates the direction of the feature. Our technique applies correctors to ensure accurate tracking along this direction. The results are good and avoid the reliance on the underlying mathematical description of the implicit model. Figure 2 depicts a jug with several feature line strokes along CSG discontinuities generated by our system.

Unfortunately, our technique does not guarantee that all silhouettes and feature line strokes will be extracted from a surface. This is because it relies on particle detection to initialize these extractions. Thus, an adequate coverage of particles on the surface is required to extract all relevant strokes. The models shown here required from 100 to 10,000 particles to extract all strokes.

5.3. Stylization of strokes

Outlines and feature lines are stylized as dark ink-filled strokes that smoothly vary in width using the angled-bisector method presented by Northrup and Markosian [MMK*00].

Particle positions directly identify locations of short interior strokes. Selection of the particles for these strokes is decided using particle measures. Particle measures include distance from the viewer, surface normal angle from a silhouette or contour, mean curvature and lighting.

Interior strokes are stylized as either short curved lines or points where the size can easily be altered. Points are used for a stippling style and can be used alone or in conjunction with short strokes. The strokes are created in a two-step process. The first step calculates the stroke direction and the second can curve the stroke based on the amount of surface curvature. The strokes can be created in one of three directions: first and second principal directions of curvature and the contour direction.

6. Results and Discussion

Figure 1 shows a model steam train containing a large number of primitives and different types of blend, CSG and warp nodes. Figure 2 shows a model of an ancient jug, constructed using several CSG and blend operations. Strokes have been placed algorithmically but guided by user defined parameters to produce artistic renderings of the models. From these images it can be seen that our techniques successfully extract and stylize silhouettes, feature lines, and interior strokes from BlobTree implicit surfaces. Moreover, users can control parameters to the pen and ink renderer to generate several styles. Our pen and ink renderer provides interactive rates for simple to medium sized models. Unfortunately, the train (see figure 1) requires more computation time, primarily for silhouette and CSG stroke extraction. This increase in time is due to the complexity of field function evaluations for BlobTree models. An increase of calculation time with tree complexity is generally true for any CSG system, however interactive updates, even for highly complex models, can be achieved during editing using our caching technique [SWG05].

The power of the sweep technique is shown in figure 11. This relatively complex 3D model was created by a non-professional user in about an hour using our direct manipulation interface. A system which exploits caching field values (rather than traversing the *BlobTree* to calculate the field values each time a scene is re-rendered) is employed to attain real time rendering (see [SWG05]). An example of the fast modeling potential of our sketching system is illustrated in figure 12. Modelling this object required only a few minutes. Figure 13 shows a result using the interactive sketching input with stylized NPR output. The bottom image in the figure makes use of stippling to emphasize contrast.

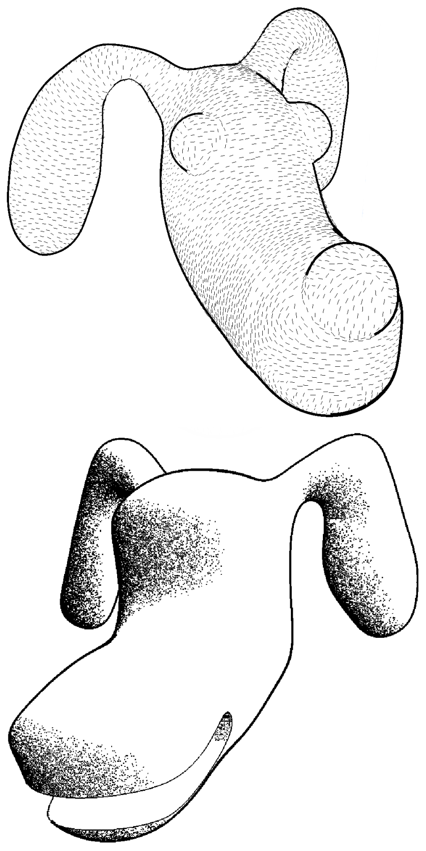
Figure 13: *The dog model was constructed interactively in Shapeshop and rendered using the NPR* BlobTree *. This figure shows (top) short ink marks placed all over the model and (bottom) stipple marks used to emphasize contrast.*

7. Conclusions and future work

The aim of this work has been to exploit particular features of implicit modeling tools to produce a wide range of models and also to make use of pen and ink rendering styles. Our goal is to design tools that facilitate building aesthetic models rendered in different styles. The potential of this approach lies in that it allows combining interaction with procedural algorithms in ways not previously possible. This is only attainable through essential features of the BlobTree system: model definition via a procedural language, defining models via direct manipulation, implicit modeling tools and non-photorealistic rendering techniques. Combining these features in novel ways opens intriguing possibilities yet to be explored by artists to realize the full potential of our approach.

Acknowledgements

This work was partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada, as well as by Portuguese Science Foundation through grant BSAB-458-2004. We would like to acknowledge the help of many of the graduate students in the GraphicsJungle laboratory, particularly Callum Galbraith who built the current implementation of the implicit modeling system.

References

[AG04] APITZ G., GUIMBRETIÉRE F.: Crossy: a crossing-based drawing application. In *Proceedings of ACM UIST 2004* (2004), pp. 3–12.

[AGB04] ALEXE A., GAILDRAT V., BARTHE L.: Interactive modelling from sketches using spherical implicit functions. In *Proceedings of AFRIGRAPH 2004* (2004), pp. 25–34.

[AJ03] ARAÚJO B., JORGE J.: Blobmaker: Free-form modelling with variational implicit surfaces. In *Proceedings of 12th Encontro Português de Computação Gráfica* (2003).

[Akl98a] AKLEMAN E.: Implicit painting of csg solids. In *Proc. of CSG '98* (1998), pp. 99–113.

[Akl98b] AKLEMAN E.: Implicit surface painting. In *Proc. of Implicit Surfaces '98* (1998), pp. 63–68.

[BH98] BREMER D., HUGHES J.: Rapid approximate silhouette rendering of implicit surfaces. In *Proc. of Implicit Surfaces '98* (1998), pp. 155–164.

[CSSJ05] CHERLIN J. J., SAMAVATI F., SOUSA M. C., JORGE J. A.: Sketch-based modeling with few strokes. In *Proceedings of the Spring Conference on Computer Graphics* (2005).

[FJW*05] FOSTER K., JEPP P., WYVILL B., SOUSA M. C., GALBRAITH C., JORGE J. A.: Pen-and-ink for blobtree implicit models. *Computer Graphics Forum (Proc. of Eurographics '05)* (2005), 267–276.

[Gas93] GASCUEL M.-P.: An Implicit Formulation for Precise Contact Modeling Between Flexible Solids. *Computer Graphics (Proc. SIGGRAPH '93)* (August 1993), 313–320.

[Gup76] GUPTILL A. L.: *Rendering in Pen and Ink*. Watson-Guptill Publications, 1976. ISBN is 0-8230-4530-7.

[IH01] IGARASHI T., HUGHES J. F.: A suggestive interface for 3d drawing. In *Proceedings of ACM UIST 2001* (2001), pp. 173–181.

[IH03] IGARASHI T., HUGHES J. F.: Smooth meshes for sketch-based freeform modeling. In *Proceedings of the 2003 symposium on Interactive 3D graphics* (2003), pp. 139–142.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proceedings of ACM SIGGRAPH '99* (1999), pp. 409–416.

[JSC03] JORGE J. A., SILVA N. F., CARDOSO T. D.: Gides++. In *Proceedings of 12th Encontro Português de Computação Gráfica* (2003).

[KHR02] KARPENKO O., HUGHES J., RASKAR R.: Free-form sketching with variational implicit surfaces. *Computer Graphics Forum 21*, 3 (2002), 585 – 594.

[MMK*00] MARKOSIAN L., MEIER B., KOWALSI M., HOLDEN L., NORTHRUP J., HUGHES J.: Art based rendering with continuous levels of detail. In *NPAR 2000* (2000).

[ONNI03] OWADA S., NIELSEN F., NAKAZAWA K., IGARASHI T.: A sketching interface for modeling the internal structures of 3d shapes. In *Proceedings of the 4th International Symposium on Smart Graphics* (2003), pp. 49–57.

[Ric73] RICCI A.: A constructive geometry for computer graphics. In *Computer Graphics Journal* (1973), pp. 157–160.

[SWCSJ05] SCHMIDT R., WYVILL B., COSTA-SOUSA M., JORGE J. A.: Shapeshop: Sketch-based solid modeling with the blobtree. In *Proc. 2nd Eurographics Workshop on Sketch-based Interfaces and Modeling* (2005), Eurographics, Eurographics, pp. 53–62. Dublin, Ireland, August 2005.

[SWG05] SCHMIDT R., WYVILL B., GALIN E.: Interactive Implicit Modeling with Hierarchical Spatial Caching. In *Proc. Shape Modelling International, MIT, USA* (May 2005), IEEE Press.

[TZF04] TAI C.-L., ZHANG H., FONG J. C.-K.: Prototype modeling from sketched silhouettes based on convolution surfaces. *Computer Graphics Forum 23*, 1 (2004), 71–83.

[WGG99] WYVILL B., GALIN E., GUY A.: Extending The CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System. *Computer Graphics Forum 18*, 2 (June 1999), 149–158.

[WH94] WITKIN A., HECKBERT P.: Using particles to sample and control implicit surfaces. In *Proceedings SIGGRAPH '94* (1994), pp. 269–277.

[ZHH96] ZELEZNIK R. C., HERNDON K. P., HUGHES J. F.: Sketch: an interface for sketching 3d scenes. In *Proceedings of ACM SIGGRAPH '96* (1996), pp. 163–170.