Kevin Foster
M.Sc. to Ph.D. Rollup

Thesis Summary

Abstract

Artists frequently imply the shape and form of an object using an accurate natural-science style employing pen-and-ink. This style is useful because it allows for fast-interpretation of images, it can help focus the viewers attention on specific details and can provide extra information about the object(s). Given these positive properties, this style is important to simulate. This thesis presents a complete set of methods to render two types of 3D surfaces in natural-science pen-and-ink styles. It examines stroke placement, style and variation, while considering interior strokes and silhouette (contour) strokes.

Table of Contents

Approval Page ii

Abstract iii

Acknowledgements iv

Table of Contents v

1 Introduction 1
- 1.1 Natural Science Pen-and-Ink Rendering 2
  - 1.1.1 Silhouette Strokes 4
  - 1.1.2 Interior Strokes 4
- 1.2 Research Goals and Motivation 5
- 1.3 Contributions 6
  - 1.3.1 Polygonal Meshes 6
  - 1.3.2 Implicit Surfaces 7

2 Background 8
- 2.1 Precise ink-based illustration 8
- 2.2 Stroke Extraction: Polygonal Meshes 10
  - 2.2.1 Definition of a Silhouette 10
  - 2.2.2 Silhouette Extraction Methods 11
  - 2.2.3 Interior mark extraction 13
  - 2.2.4 Shape measures in mesh models 13
- 2.3 Stroke Extraction: Implicit Surfaces 14

3 Pen-and-Ink Polygonal Meshes 16
- 3.1 The Silhouette Stroke Extraction System 16
  - 3.1.1 Results 18
- 3.2 The Interior Stroke Extraction System 19
  - 3.2.1 Results 22

4 NPRBT: The Non-Photorealistic Blobtree 28
- 4.1 NPRBT System Architecture 28
- 4.2 NPRBT Results 30

Bibliography 36

Chapter 1

Introduction

Historically, work in computer graphics has focussed on developing tools, rendering methods and hardware to create realistic images (Figure 1.1). Such images have many applications, but the idea that *realistic images are always the most desirable style of image*, does not always hold.

Consider the crosswalk street-sign in figure 1.2. Pedestrians wondering where to cross the street do not require information from the sign such as the colour of the person's clothes or the time of day when the image was created. The only information that a pedestrian needs to know is that they are allowed to cross between the lines painted on the street. The non-photorealistic interpretation of this, shown in figure 1.2, successfully delivers this information by focussing the viewer's attention on the important details without confusing them with extra information. Use of non-photorealistic images can also be found in medical textbooks and technical manuals (natural science and technical illustration), hospitals (medical imagery), building and electrical drafting (blueprints and circuit diagrams) and in computer animation. This widespread use can be explained because of 3 main reasons. *Non*-photorealistic images can (1) focus a viewer's attention on specific detail that the artist wishes to convey, (2) convey extra information which is impossible to include in photorealism and (3) add a special expression or feeling to the image.

It is for this reason that interest and research in using computers for Non-photorealistic rendering (commonly referred to as NPR), is increasing. Systems developed in the field of NPR enable creation non-photorealistic images faster than with traditional approaches,

Figure 1.1: A an example of graphics development for photorealism: *id Software*'s 1st person games over the past 15 years. Images are organized from the earliest on the left to the most recent on the right.

Figure 1.2: A crosswalk street-sign.

facilitate tools to help artists, and can be used to preform many non-photorealistic operations that cannot be done by a human, such a creating accurate visual interpretations of MRI and cat-scan data.

1.1 Natural Science Pen-and-Ink Rendering

The medium of pen and ink consists solely of black ink marks on white paper. No tone or colour is used in these images. Despite this simplicity, an artist can use different applications of ink to create stunning interpretations of objects in a variety of styles. These images can be anywhere between extremely accurate and loose interpretive drawings. In my thesis, I will focus on accurate natural-science style images.

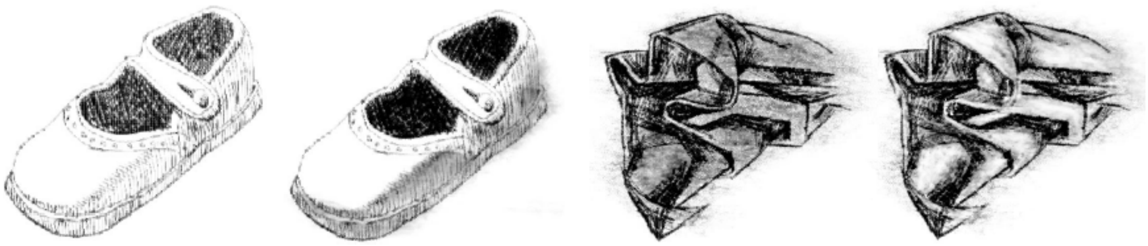

Figure 1.3: Images generated by a Non-Photorealistic rendering system.

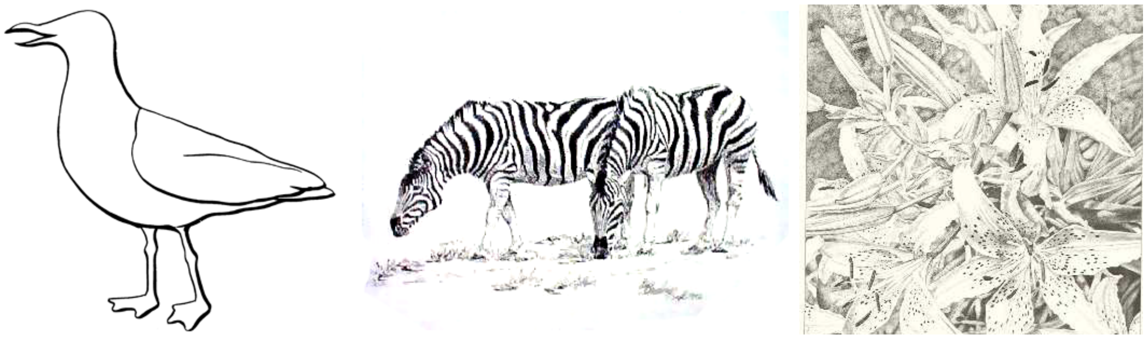

Figure 1.4: Images of various natural-science pen and ink styles.

Some examples of images in this style can be seen in figure 1.4 the use of strokes with varying width, dots used to create the appearance of strokes and sets of small marks to create texture. The precise drawings that natural-science artists create are an excellent way to understand structures and details of various subjects. The primary purpose of such drawings is to make shape characteristics of the subject visible, to lessen the possibility of misinterpretation, and to please the eye in terms of proper artistic handling of the subject. Traditionally, precise drawings are produced in three main steps: 1) Shape characteristics (i.e. folding regions, surfaces areas, volumes, curvatures) are accurately identified and measured; 2) regions related to the shape measures are lightly outlined using pencils, and 3) there regions are filled with pen strokes, with a gesture that conveys a careful constructed look [Sim92, Hod89, Raw87].

1.1.1 Silhouette Strokes

Natural-science artists place a great deal of attention on the contours (silhouettes) of objects. A general principle in drawing states that “a good outline means good perception of mass” [Cra00]. This is achieved in two main processes [Cra00, Hod89, Whi94]:

- by emphasizing the placement of the subject”s outline outside the silhouette boundary of its form rather than within it.
- by carefully controlling the various characteristics or qualities of the line, in particular the suggestion of movement which is achieved by drawing long line segments with various degrees of linear weight and emphasis.

These processes are visible in the seagull image in figure 1.4. The weight and emphasis variation depends on the subject matter and on the information that the illustrator wants to present (ie. shape features, depth-cues, focus of attention).

1.1.2 Interior Strokes

A great deal of focus is also spent on creating marks and strokes on the interior of the surface. Interior strokes can be implemented to represent virtually any shape if used properly [Sim92, Hod89]. As an individual primitive, interior strokes are either lines or dots, black, narrow, and consistent in thickness. In clusters, they create a cumulative effect resulting in tone values which helps to ensure proper revealing of the subject’s shape characteristics. To this end, the main challenge facing the illustrator is on representing tri-dimensionality in drawings, given the fact that strokes are essentially 2D marks placed in a plane. To overcome this limitation, illustrators follow two processes similar approach to those used for contour lines:

- by increasing stroke width at certain curvature angles, junctions, creases [Hod89, Raw87].
- by carefully adjusting the stroke spacing and adjusting the stroke direction to control cumulative effects of strokes.

If stippling is not used, strokes can follow same direction of the contour, the primary or secondary principal direction of curvature or an arbitrary direction. Observe carefully placed and stylized interior strokes in the images of the zebra and the tiger lily in figure 1.4.

1.2 Research Goals and Motivation

The fundamental goal of this research is to provide a comprehensive set of systems to generate accurate pen-and-ink surfaces for various types of 3D surfaces. This research is valid for three main reasons. First of all, the pen-and-ink natural science style is used in many types of situations (as listed previously) and thus is an important style to reproduce. Secondly, use of the systems can save time and money in the artistic pipeline because the systems presented here can create images in seconds which often take hours to draw by highly-trained artists. Finally, the systems are 3D and built to run interactively, giving the user freedom to view an object at any angle. Furthermore, in the case of the system for implicit surfaces, scenes can be rendered faster than using many photorealistic approaches, which means that it is useful for shape design and prototyping.

1.3 Contributions

For polygonal meshes and implicit surfaces, this thesis will present methods to create high-quality pen-and-ink images efficiently. New work will be presented for almost every step in the following pipeline:

1. Stroke extraction: methods to extract accurate strokes from the surface (interior and silhouette)
2. Shape measures: functions which evaluate different metrics on the surfaces
3. Stroke rendering: ways to render the strokes using the shape measures in a certain style

Throughout these steps, the systems will simulate rules followed by natural-science pen-and-ink artists. To this end, silhouette (contour) strokes and interior strokes will be rendered with separate methods which each follow unique strategies to emulate the effects created by these artists.

1.3.1 Polygonal Meshes

For polygonal meshes, this thesis will contribute both to silhouette extraction and stylization and to interior stroke extraction and stylization. For silhouette extraction, this thesis will present (1) a new method to remove errors and artifacts for silhouette strokes, along with (2) a stylization step and (3) a long stroke creation (chaining) step. For interior mark extraction, this thesis presents (1) a new method to use morphometric shape variables to apply ink to edges in the polygonal model and (2) methods to simulate various interior mark-making styles with this approach.

1.3.2 Implicit Surfaces

For implicit surfaces, this thesis will contribute a system called the NPR Blobtree (NPRBT) which renders Jungle Blobtree implicit surfaces quickly while simulating the natural-science illustration style described previously. There are only three main works that have been previously presented which examine Non-Photorealistic Rendering techniques for implicit surfaces, by Bremmer and Highes[BH98], by Elber[Elb98] and by Akleman[Akl98]. Bremer, Hughes and Elber's systems are presented only for simple blending implicit surfaces. The new system in my thesis will be a combination and extension of these previous works which is compatible with the Blobtree. Specific contributions for silhouette extraction and stylization include: (1) silhouette extraction methods will be accelerated and improved to handle CSG operations and high-frequency areas, (2) various ways to apply pen-and-ink styles to these strokes will be presented and (3) methods to specifically extract CSG strokes (that do not fall on the silhouette) will be explored. Specific contributions for interior stroke extraction and stylization include: (1) a simple pipeline to apply strokes to a surface, (2) a method to curve strokes to follow surface curvature and (3) new approaches to extract longer strokes interior strokes for static Blobtree surfaces.

Chapter 2

Background

Research for Non-Photorealistic Rendering is blossoming. Many systems are now available which work with various types of data (such as a 3D surface, an image, a set of stroke positions or a method to generate such positions) which can stylize this data in many non-photorealistic styles such as water-color, pen and ink or charcoal. This rendering can be accurate, as is commonly seen in technical illustration, or it can be loose and sketchy, as seen in drafting images and in some types of art. Furthermore, the system can be a realistic physical simulation, can be based on observations of real life phenomena, can be an ad-hoc visual approach or can be some combination of all of these.

2.1 Precise ink-based illustration

Different approaches for placing small pen-and-ink primitives over all types of 3D models have been proposed. Winkenbach and Salesin [WS] present a convincing method to render pen-and-ink strokes on polygonal models and parametric surfaces. They introduce “stroke textures” for polygonal meshes, which allows for procedural accumulation of strokes to create tone and texture, and “controlled density hatching” for parametric surfaces, which removes and adds strokes on the parametric surface to maintain a selected tone or effect. Their system also demonstrates effects such as silhouette generation, scale-dependant rendering and shadow casting.

Several systems build on this technology of stroke textures [WS] including Praun et al. [PHWF01] who introduce “tonal art maps” which organize pre-rendered strokes as a sequence of mip-mapped images. Strokes within the images are scaled to attain appropriate stroke size and density at various resolutions. This method, an early example of using hardware multitexturing for NPR, uses lapped textures [EP00] to place seamless textures on the surface. Lapped texturing is a method of repeatedly placing small texture patches on a surface in a way so that joins between the textures aren’t visible until it is entirely covered.

Elber [Elb98] provides one of the main bases for this thesis with a techniques for creating small strokes across freeform implicit and parametric surfaces. Strengths of his system included the ability to draw the strokes in various directions, such as principal directions of curvature, the ability to group strokes in several configurations and several interesting stroke styles.

Deussen and Strothotte [DS00] present a system which generates pen-and-ink images of 3D trees. Their system uses a hybrid pixel/geometric approach to handle the geometry of the tree skeleton and leaf particles. The main strength of this system is the ability to generate pen-and-ink images at greatly different levels of detail in several different styles. Level of detail is used for several purposes by artists including depth perception, focussing the viewers attention and to add style and feel to the image.

An important pen-and-ink style is stippling, a style that involves precise positioning of small dots such that their density produces tone and implies shape and texture. Recent investigation into this style has focused on the geometric relation between the stippled dots [DHvOS00] and on interactive direct volume illustration systems [LME$^+$02]. The system from Deussen et al. [DHvOS00] provides an automatic method to place stipples based on halftoning images and a user-driven painting interface which generates convincing stipples extremely quickly. Using this interface, the user needs only to “paint” the regions on the surface where stipples exist. The system automatically places the individual stipples using a relaxation method based on Voronoi diagrams to evenly distribute the stipples, randomly jitter the stipples and to shape the stipples.

Secord [Sec02] presents two techniques, one real-time using pre-computed information and a slower high-quality iterative technique, for generating stippled drawings using weighted Voronoi centroidal diagrams. This system places small arbitrarily-shaped primitives on grayscale images to achieve a convincing stippling effect and requires very-little user input. More recently, Pastor et al. [OEMP03] attached stipple particles to the surface of the model, using a point hierarchy to control the stipple shading density.

The approach taken in this thesis for implicit surfaces is similar to most of these works—particles are applied to the surface and then selection as to which particles to be drawn is executed. The polygonal model system uses a different approach—it uses existing mesh edges to create strokes on a fixed basis across the mesh.

2.2 Stroke Extraction: Polygonal Meshes

Many methods have been presented that extract silhouettes and interior strokes from 3D polygonal meshes.

2.2.1 Definition of a Silhouette

The silhouette for a smooth surface is defined as follows: a point X on a surface with normal $\mathbf{N}_X$ is on the silhouette for an eye position E if the angle between $\mathbf{N}_X$ and $(X - E)$ is 90 degrees. This definition includes interior silhouettes as well as the object’s outline.

For polygonal models, the silhouette set is defined as the set of edges in the mesh which share a front-facing and a back-facing polygon. For any polygon in the mesh, X is set to be the midpoint of the face. Assuming the normal $\mathbf{N}(X)$ is pointing away from the surface, if $\mathbf{N}(X) \cdot (X - E) > 0$, the polygon is back-facing and if $\mathbf{N}(X) \cdot (X - E) < 0$, the polygon is front-facing.

The simplest approach to find these silhouette edges is the brute-force approach. This method iterates through each edge in the polygonal mesh and finds silhouettes by determining the orientation for the polygons that the edge belongs to as described in the previous paragraph. While this approach is easy to implement, there are many faster methods to find silhouette edges.

2.2.2 Silhouette Extraction Methods

Many methods have already been presented which extract silhouette from polygonal meshes more efficiently than with the brute force approach. These methods use object-space or image-space approaches (or a combination of both of these) and produce either individual silhouette edges or chains of joined silhouette edges. The composition of the silhouette chains varies from system to system too-some systems use edges from the polygonal model directly in the silhouette while others create new edges on the mesh.

Markosian et al. [MKT$^+$97] use probabilistic testing to find silhouettes. This method chooses which edges to test for silhouettes based on a list of edges that were silhouettes in previous rendering steps. The overall effect of this is that silhouettes can be extracted very quickly, but some may be missed. However, this is the point of the paper as accuracy and detail is deliberately traded to make the system real-time.

Gooch et al. [GSG$^+$99] present a system for interactive technical illustration of polygonal models. This system includes methods to colour the interior of the surface with specialized shading methods and several methods (two hardware and one software) to draw silhouette curves and creases. The hardware methods (1) draw front faces and back faces using filled modes and line modes in a special way to highlight silhouettes and (2) use environment maps with edge lines at the silhouette (this achieves an interesting artistic effect). Their software method is more elegant. Their system projects the vertex normals for each edge onto a 3D sphere called a "Gauss Map" and stores the arcs created by the two vertex normals from an edge. Silhouettes for a certain angle can be found by finding the arcs which intersect a plane through the origin of the sphere. This method gains efficiency by storing arcs in a hierarchy which allows for quick culling of regions that cannot contain a silhouette.

Hertzmann and Zorin [HZ] present a system which calculates hatch marks and silhouettes. Their system, like Gooch's [GSG$^+$99] uses another method to quickly cull edges which cannot be silhouettes. This method is based on geometric duality and removes silhouettes by limiting the problem to intersecting a plane with a surface. However use of the system requires some abstract math which converts points between geometric space and space on a hypercube and must use a oct-tree subdivision hierarchy.

Sander et al.[SGG$^+$00] use cone maps to bound the front-facing and back-facing areas of the mesh to quickly find areas that contain silhouettes.

The "edge-buffer"[BS00], extracts all silhouettes efficiently with an edge adjacency information graph. Every time the scene is rendered from a different angle or the model moves, front/back-face bits are set in the buffer and silhouette edges are quickly extracted using bit-wise logical operators.

2.2.3 Interior mark extraction

Marks on the interior of the silhouette are another component that can be used to illustrate an object. They can be used to illustrate depth, form, texture, curvature and other surface properties. Methods to evaluate surface metrics and lighting equations have been explored greatly for both surface types, although once again there is less work that actually uses these methods for a NPR approach for implicit surfaces.

2.2.4 Shape measures in mesh models

Researchers have explored the use of lighting parameters/equations for measuring and rendering models in the context of technical and scientific illustration [GGS$^+$98, GSG$^+$99, HS99]. The approach used in this thesis is to extract and render shape features by calculating local shape measures directly at the 3D mesh, with no need for either illumination or surface reflectance information. For polygonal meshes, shape measures are usually estimated at every vertex of the model, taking into account some local properties of the adjacent triangles to each vertex, such as triangle angles, areas, and edge lengths [Boi95, YKM99, GIHL00]. In NPR, the focus has been mainly on shape measures related to principal direction of curvature to guide the stroke placement process [GIHL00, HZ, RKS00, PHWF01]. The approach used in this work is to use geomorphological shape measure calculation schemes [Eva72, MH93, SSM02, Woo96]. They provide a large and computationally stable collection of shape measures. These methods are adapted to work directly with 3D triangle meshes, organized in a modified edge-buffer data structure [BS00]; the main resulting benefit is that now various shape measures can be directly calculated at each edge of the model using only the information of its two adjacent faces. This modified edge-buffer is now able to efficiently extract and render edges related to silhouettes and to different shape measures as well.

2.3 Stroke Extraction: Implicit Surfaces

There are three works that address silhouette and interior stroke extraction and stylization for implicit surfaces[BH98, Akl98, Elb98]. Akleman[Akl98] presents a system specialized to extract long strokes for oil painting. This approach uses a particle system and an integrator to find long flowing strokes along the surface in various directions. Furthermore, the system can create both accurate and artistic renderings.

Bremer and Hughes[BH98] use a hybrid raytracer/integration/OpenGL system to extract silhouettes and some interior strokes. To do this, their system creates random rays from the scene's eye and collides them with the implicit surface. For successful intersections, a small interior mark is placed on the surface. To extract the silhouette, the system uses an integrator find the silhouette of the surface from this point. Finally, silhouette strokes are extracted using another integration method which estimates the silhouette direction and visible portions of these silhouettes are rendered in an ink-line style.

Elber[Elb98] presents a system which renders short strokes on the interior of the surface. To accomplish this, his system distributes particles on either parametric and implicit surfaces, evaluates shape measures and then extracts strokes directly from the particles. These strokes can be grouped and directed in various orientations to convincingly convey details of the surface. A variety of stroke types, such as scribbles, straight lines and jagged lines, can be applied to the particles and his system allows for transparency. Although this system does not specifically extract long silhouette strokes, strokes can be grouped near the silhouette of a surface.

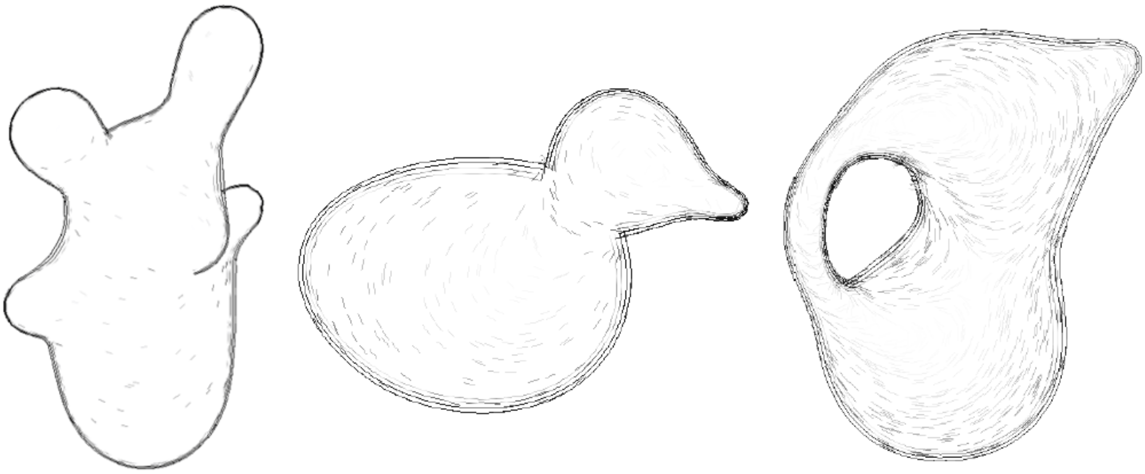

Figure 2.1: Several surfaces rendered with Bremmer and Hughes' system.

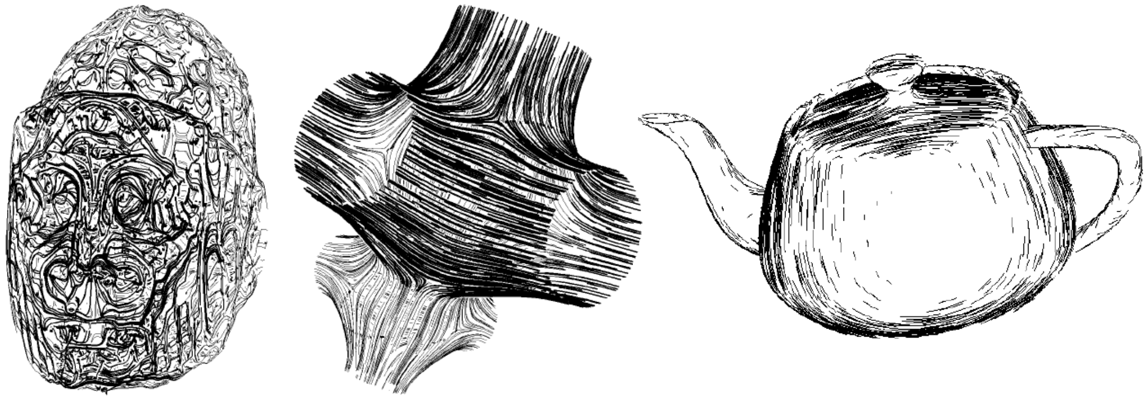

Figure 2.2: Results from Elber's system. The left two images are implicit surfaces. The right image is a parametric surface with strokes grouped around the silhouette.

Chapter 3

Pen-and-Ink Polygonal Meshes

This chapter provides an overview of a pen-and-ink system for polygon meshes. Key goals of this system is good rendering rates, efficient shape measures calculation and visual quality resembling traditional precise pen stroke drawings. Input to the silhouette system can be any 3D polygonal model, although the system produces better results for dense meshes. The interior stroke extraction system can take as input dense 3D triangle meshes with arbitrary topology.

The system described in this chapter is broken into two parts:

1. the *Silhouette Stroke Extraction System*: responsible for extracting silhouettes, correcting errors in the silhouettes and stylization
2. the *Interior Stroke Extraction System*: responsible for extracting and stylizing strokes on the interior of the surface

3.1 The Silhouette Stroke Extraction System

For silhouette strokes, this thesis will contribute the *Silhouette Stroke Extraction System*. There has already been significant research in non-photorealistic rendering focusing on quality silhouette extraction and rendering, in particular for 3D polygonal mesh-based silhouette line stylization algorithms [IHS02, MKT$^+$97, NM00, SP]. Such algorithms are usually organized in four main steps:

1. Extraction of individual silhouette edges from the mesh.
2. Linkage of silhouette edges together to form long, connected paths, or chains.
3. Removal of silhouette errors and artifacts from the chains.
4. Stylization of the strokes.

Step 4 involves two main sub-processes: (1) smoothing the stroke by fitting splines or using an interpolation/approximation scheme and (2) creating line quality attributes in the stroke such as width and brightness. Although there is a great deal of work which extracts and stylizes silhouettes from polygonal meshes (*steps 1,2 and 4*), there are few examples that attempt to correct errors and artifacts that can be created when this extraction takes place (*step 3*).

Figure 3.1 shows examples of silhouette errors and artifacts. These occur because of numerical instability and unsuitable edges from the polygonal mesh (the mesh is a discrete approximation of a surface). Observe figure 3.1(b), where four frames illustrate different combinations of artifacts. Silhouettes for these images have been calculated for an angle other than that displayed. From this image, it is apparent that edges taken directly from the mesh are not ideal to construct the silhouette. The quality of the stroke stylization process and subsequent rendering results are compromised due to these errors. In this thesis, a method to correct these errors and artifacts, along with an automatic stroke smoothing step, will be presented. Also, a method to extend the Edge-Buffer system to create complete stroke chains (step 2) will be presented.

To remove artifacts, silhouette chains are interpreted from the point of view of low and high-frequency portions of the silhouette curve. The low-frequency data is the correct

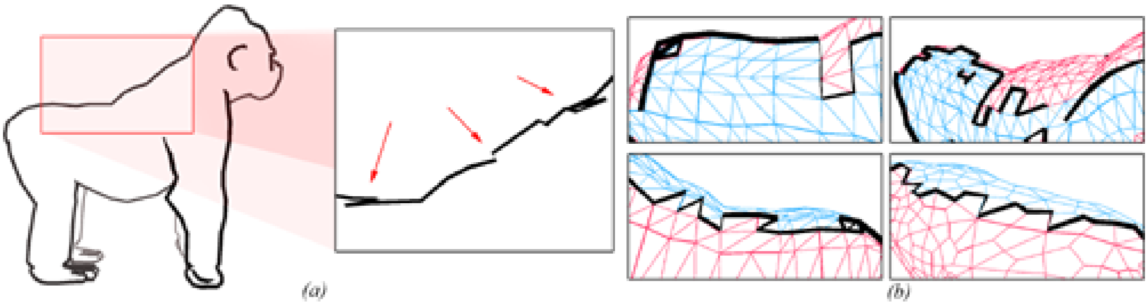

Figure 3.1: **(a)** The silhouette of an ape mesh with highlighted errors. **(b)** Four images showing various silhouettes and underlying mesh that generated them. The silhouettes are presented at a perturbed view to provide a better understanding of the cause of the errors. Shaded polygons are back-facing.

silhouette path and the high-frequency data is the errors. Thus, the solution comes from removing the high-frequency errors from the chain.

3.1.1 Results

The *Silhouette Stroke Extraction System* works effectively for most meshes and can generate error free strokes with minor user input (Figures 3.2 to 3.4). Furthermore, the system achieves fast computation rates including preprocessing (building the edge-buffer) and rendering (chaining, multiresolution filtering, and stroke stylization).

The error removal method presented gains speed over other silhouette error correction methods because the system does not need to identify errors to remove them. Thus, error condition/correction cases are not required. Error-removal for smaller meshes with 4000 faces can be done in about 1 millisecond, while larger meshes with 45000 faces take 60 milliseconds on average. However, the system does not guarantee error-free silhouettes. Although removing errors for most strokes is easy, error removal from some strokes extracted from smaller meshes may fail because important features of the silhouette may be removed with the errors.

In these cases, the system presents a tradeoff between feature-preservation and quality of filtering (directly related to user input). This is mainly a problem with smaller meshes because the raw strokes from these meshes provide insufficient detail to remove high-frequency (see figure 3.5).

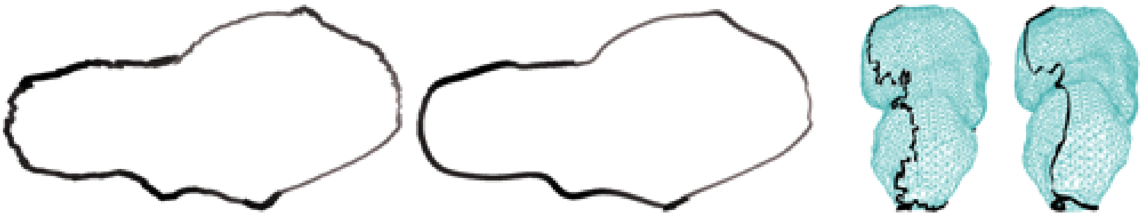

Figure 3.2: From left to right: Original silhouettes from an asteroid, the results of processing and alternate views of the strokes with the mesh.

Figure 3.3: From left to right: Original silhouettes from an inner-ear, the results after processing, and alternate views of the strokes with the mesh.

3.2 The Interior Stroke Extraction System

This section introduces the *Interior Stroke Extraction System* which generates accurate strokes on the interior of 3D meshes efficiently. The key goals of this system are good rendering rates, an efficient scheme for shape-measures calculation and visual quality resembling traditional precise pen stroke drawings (figure 3.6).

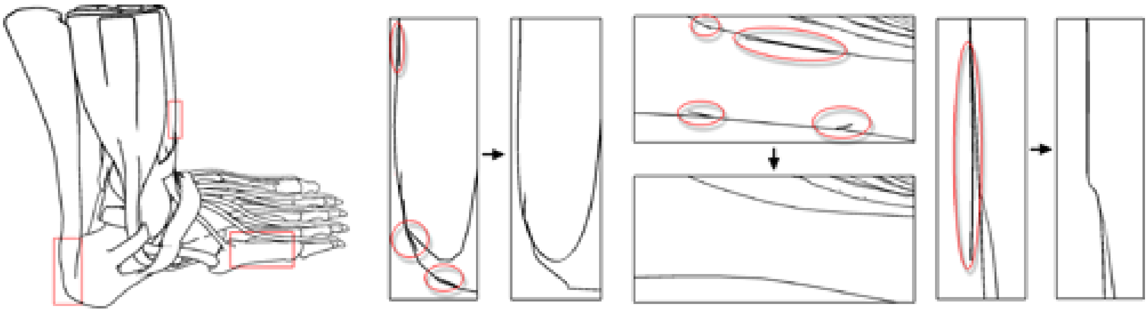

Figure 3.4: Removing silhouette errors on large meshes is more important when zooming in on the mesh. Here, we circle the errors on three enlarged areas on the foot and provide our corrected strokes. Note that the errors are removed and the strokes are still very accurate to the mesh.

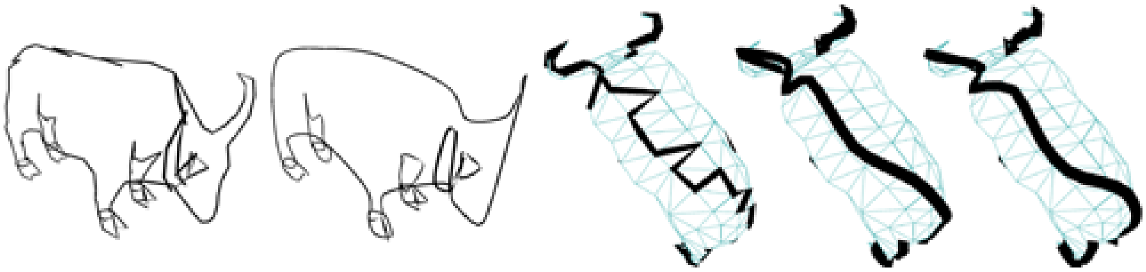

Figure 3.5: Left to Right: A silhouette from a low resolution ox mesh, processed strokes with global Cubic B-Spline filters, an alternate view of the original and processed strokes, and processed strokes with global Chaikin filters. Note that the corrected strokes do not adhere well to the original mesh.

Figure 3.6: Real precise drawings: (left) Portion of American Portrait by William Henson, (middle) archaeological artifact by Emily S. Damastra,(right) image of woman by Humberto Costa Sousa.

Typical NPR approaches to procedurally generate strokes as individual primitives and to placing them directly on 3D models involve processes that can be costly. These include distributing strokes across the surfaces, evaluating hand-gestures functions (i.e. pressure, slanting, waviness), linking strokes in chains, fitting curves to stroke sequences, among others. The approach presented here does not follow any of these approaches and instead embodies the following three main strategies:

1. One stroke per mesh edge: Each stroke has the same length and location of its corresponding edge, and is modelled and rendered individually (i.e. no chaining). This strategy provides rendering at reasonable rates with temporal coherence, as the strokes are fixed to their edges on the model, and are not redistributed for each frame.

2. Edge-based shape measures: The method calculates shape measures at every mesh *edge*, using only information from its two adjacent faces. This is achieved by extending the edge-buffer data structure [BS00] and by adapting shape measure calculation schemes from geomorphology [SSM02].

3. Pen stroke thickness and styles: The system automatically adjusts the thickness of each stroke as a function of surface curvature estimated at the edge; the user controls the parameters of stroke style for placing different types of pen marks and for achieving ink distribution visual effects.

A sample session of our system is provided in Figure 3.7. The user can use several shape measures (positive mean curvature, negative mean curvature, slope steepness, dihedral angle and slope aspect) to apply ink and can modify the way that ink is applied to the edge.

3.2.1 Results

The *Interior Stroke Extraction System* achieves fast computation rates including pre-processing (building the edge-buffer and calculating shape measures) and rendering (automatic stroke thickness adjustment and interactive pen marking). Figures 3.8 to 3.11 show results using the system. In the caption for each figure, the effects achieved for different shape measures selection, thresholds, and pen-marking styles are provided.

Running times, provided in table 3.2.1, were gathered from a 2.65 GHz Pentium IV with OpenGL/ATI Radeon 9700 graphics. The measured frame rate provides the user with an acceptable level of interactivity for exploring and illustrating various shape measures on mesh models. The user is able to quickly select and threshold clusters of feature edges related to certain shape measures, to adjust the parameters of stroke styles and thickness distribution effects. The system has good temporal rendering coherence with only some temporal aliasing occurring at the silhouette edges as new strokes are added based on the

Model	$\triangle$s	*edges*	*preproc.*	*render*
Rihard Jakopič	108,507	56,610	6	1
Broken Adze	118,676	59,339	6	1
Female Head	154,650	78,560	8	1
Gargoyle	206,982	103,740	11	2
Hip	265,084	136,099	14	2
Fossil Skull	284,458	146,123	15	4
Preformed Adze	401,060	200,529	22	4
Hammerstone	725,828	362,915	235	7

Table 3.1: Average times (in seconds) for pre-processing and rendering the models presented in this paper.

silhouette extraction and automatic thickness adjustment. The system allows for some level of control when the object is viewed from far away and in close-up. The user can adjust the style parameters of the pen marks and also select depth-dependent ink distribution effects.

The system only guarantees good strokes when used on models defined by very dense meshes. Use of less detailed meshes can result in poorly placed strokes which are distracting. Furthermore certain types of models made of many flat-surfaces (i.e. modern buildings) might produce unwanted artifacts in placement of the strokes.

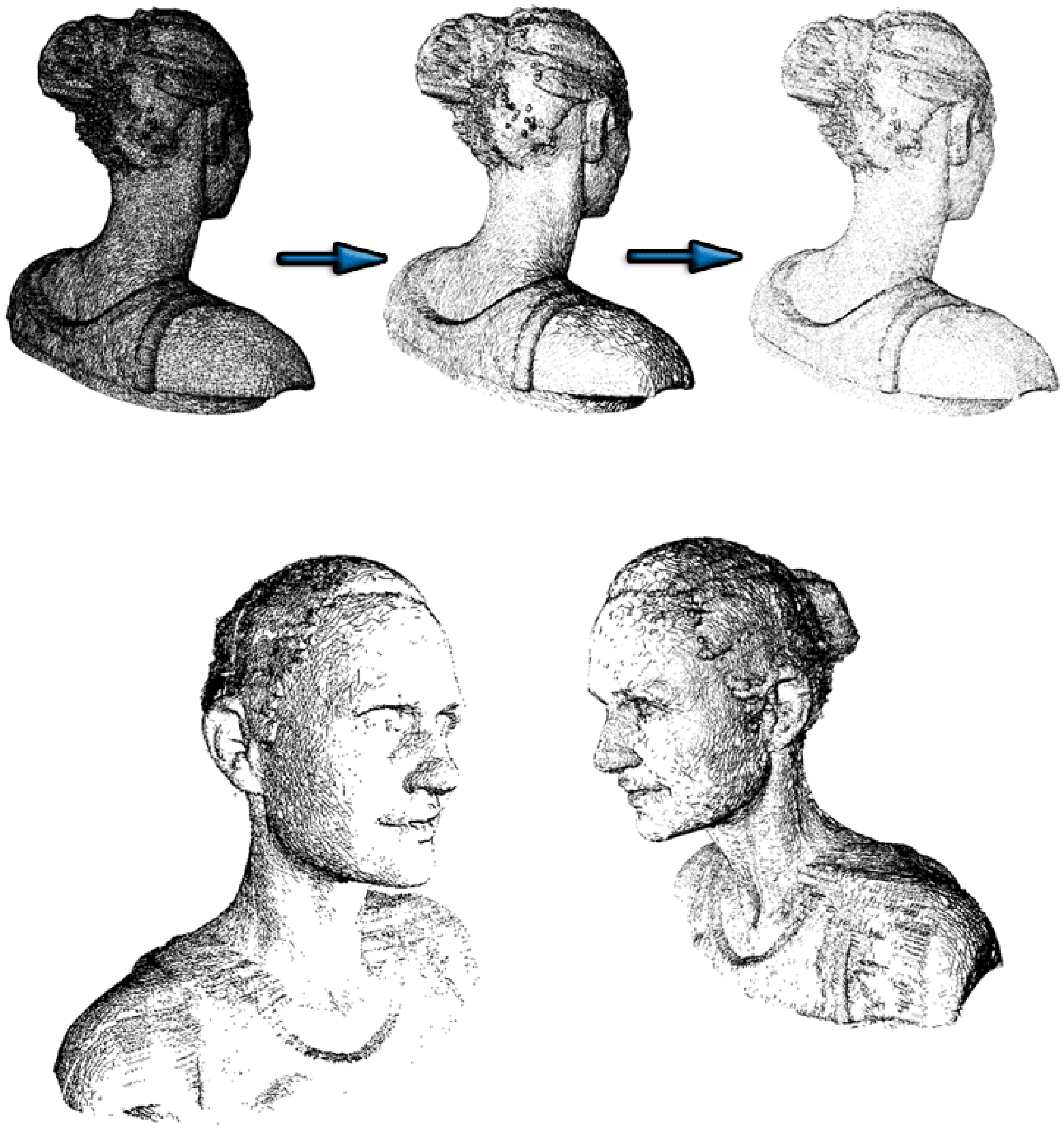

Figure 3.7: **Female head** (built from laser scans); *Top-row, left:* starting with the wire-frame mesh, filled strokes are placed at slope steepness and positive mean curvature; Notice how various anatomic and hair features are revealed. Next, filled with serrated marks are used, resulting in a soft stippling effect. *Bottom images:* Side views of model rendered with filled strokes using the setup used for top-right image. *Model source: Cyberware.*

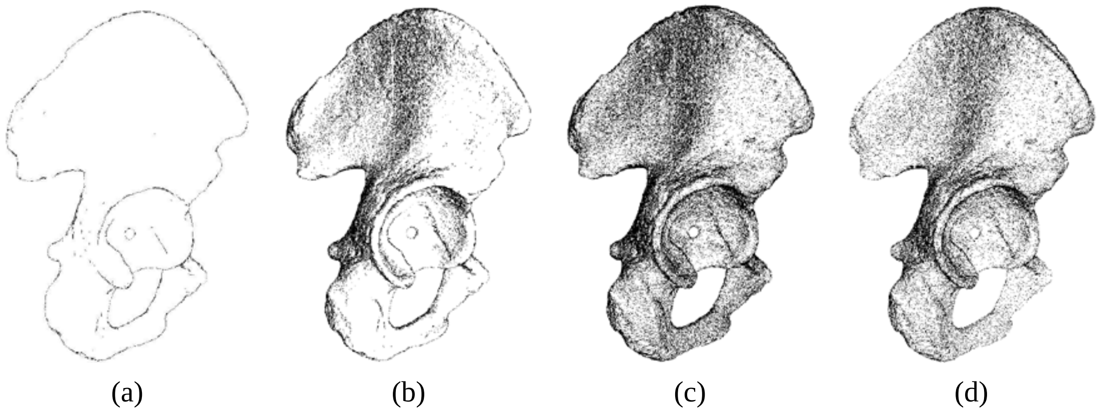

Figure 3.8: Using the interior stroke extraction system for medical illustration of a *hip* model built from laser scans. (a) First silhouettes are rendered with thickness variation as a function of curvature; (b) Concave formations are revealed by placing filled strokes in locations with negative mean curvature; (c) Creases and convex formations are revealed by placing filled strokes in locations with positive mean curvature; (d) Finally, the view-depth effect is applied to improve the depth perception.*Model source: Cyberware.*

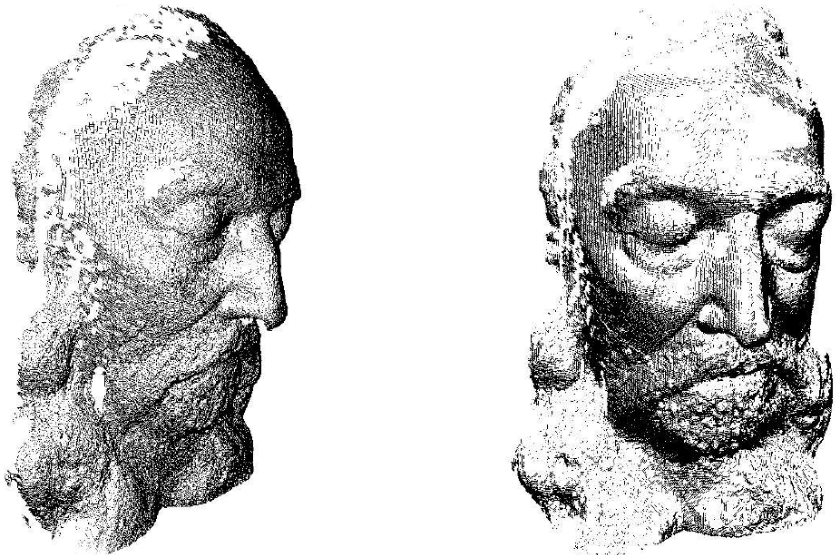

Figure 3.9: A mask of artist Richard Jakopic (built from range images); Left: Covering the model with filled strokes on locations of negative mean curvature and positive mean curvature and slope aspect. Notice how facial features are properly revealed. Model provided by Danijel Skocaj, University of Ljubljana Computer Vision Lab.

Figure 3.10: A gargoyle built from range images; Left: Stroke placed in areas of high steepness. Top-Right: Placing strokes for all other shape measures, with emphasis on slope aspect. Notice how the directional variation of strokes reveal the curved shapes of the statue. Model provided by Rich Pito, University of Pennsylvania GRASP Lab.

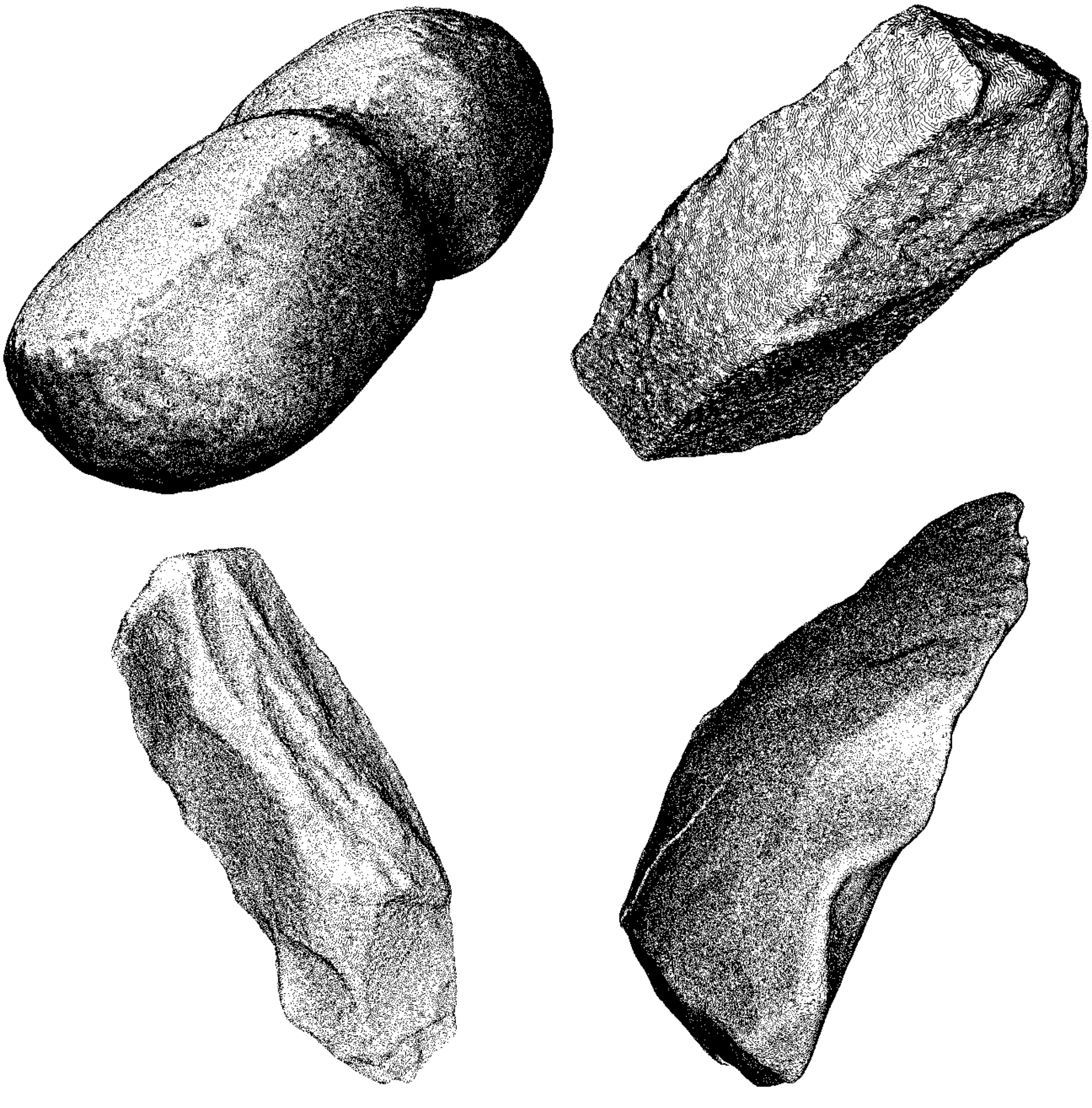

Figure 3.11: Archaeological illustrations of objects built from laser scans. Top-left: a large oval hammerstone with filled strokes revealing slop steepness and locations with negative mean curvature. Top-right: a broken preformed adze rendered with filled strokes for slope aspect; Notices the variations of stroke directions, revealing the irregularity of the shape structure. Bottom-left: another adze with a different stroke style. Bottom-right: a preformed adze with filled strokes for positive and negative mean curvature and the effect of increasing stroke thickness as they get closer to the viewer.

Chapter 4

NPRBT: The Non-Photorealistic Blobtree

In this chapter, the Non-Photorealistic Blobtree (NPRBT) is introduced. The original Blobtree supports creation of implicit surfaces from a wealth of operations including various types of blends, constructive solid geometry (CSG), texturing and warps and rendering with traditional photorealistic approaches. The new NPRBT system is built to render these surfaces in accurate pen-and-ink styles.

A wide range of styles are provided for the implicit system which includes support for transparency, different stroke styles and several different stroke placement techniques.

4.1 NPRBT System Architecture

The NPRBT system is a renderer which accesses surfaces defined in the Jungle Blobtree system. The Jungle Blobtree system handles loading and interpreting the surface, including all mathematical operations such as CSG, warps, different primitives, and control of the $field$ function. The NPRBT system adds a small set of additional functions and also adds the rendering layer which produces the pen-and-ink style.

The NPRBT system is divided into seven main modules, as visible in Figure and 4.1. These modules are:

1. the *Blobtree Interface Engine* which joins the Blobtree system with the NPRBT and provides added shape measure functions including miscellaneous functions to calculate curvature, the hessian and jacobian of the surface.

2. the *stroke positioning system* which controls a set of particles on the implicit surface. These are used as a base for the stroke extraction systems.

3. the *silhouette stroke extraction system*, used to calculate silhouette strokes.

4. the *interior stroke extraction system*, used to calculate interior strokes.

5. *silhouette stylization system* which stylizes silhouettes and performs hidden line removal.

6. *interior stroke stylization system* which stylizes interior strokes and performs hidden line removal.

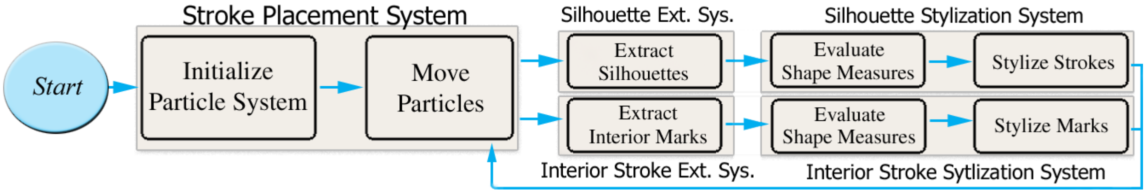

Figure 4.1: The NPRBT system pipeline.

A brief overview of the process of loading an implicit surface to creating a pen-and-ink image of the image is as follows. First, the Jungle Blobtree System loads the surface, creating the surface definition in memory. Then, the events illustrated in figure 4.1 begin. The *stroke positioning system* initializes a set of particles on the surface and begins a loop which moves the particles into a certain orientation. For each step of this loop, stroke rendering takes place. Any particles found on the silhouette are used in the *silhouette stroke extraction system* to extract the silhouettes into a data structure called a chain. Once this is complete the *silhouette stylization system* performs Hidden Line Removal algorithm to check the stroke chains for visibility. Visible portions of the chains are evaluated with several shape measures and stroke are created and rendered, simulating different styles from pen-and-ink rendering. Concurrently, certain other particles are used to create small interior marks with the *interior stroke extraction system* (figure 4.2). The system determines which particles will be used to create a stroke using two Hidden Line Removal operations and user controlled shape measure restrictions. Once these are found the *interior stroke stylization system* creates short marks in certain directions and scales, both determined by shape measures or shape properties.

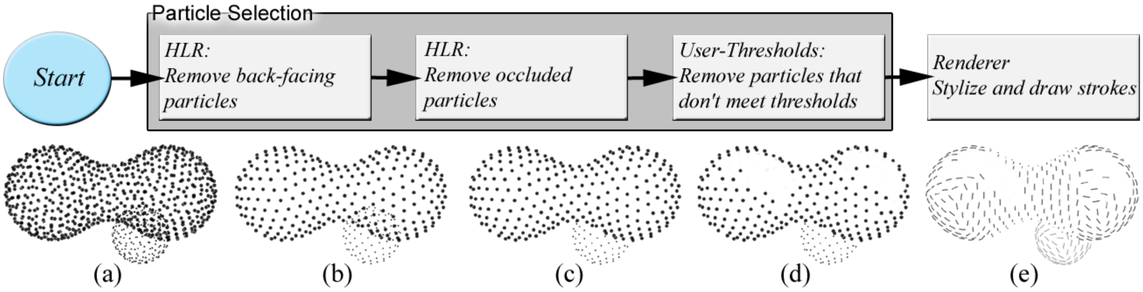

Figure 4.2: Top: The interior stroke extraction system evaluates particles for visibility (HLR) and for certain shape measures to determine which particles get strokes applied to them. Bottom: (a) All particles in a sample scene, (b) removing back facing particles, (c) removing occluded particles, (d) removing particles based on a user-selected measure, (e) stylizing small marks in the principal direction of curvature on the surface.

4.2 NPRBT Results

Some early results from system implementation are now provided. The system can extract silhouettes from implicit Blobtree surfaces and render them in several styles as is visible in Figure 4.3. In the current implementation of the system, the system can extract looping and non-looping silhouettes efficiently, although CSG curves cannot yet be extracted. A grid data-structure ensures that duplicate silhouettes are not extracted and so that silhouettes join properly. The silhouette extraction experiences a speed increase over Bremer and Hughes'[BH98] approach because, of the three steps that Bremer and Hughes' work requires for silhouette extraction, two steps are eliminated, and the third step is less computationally expensive.

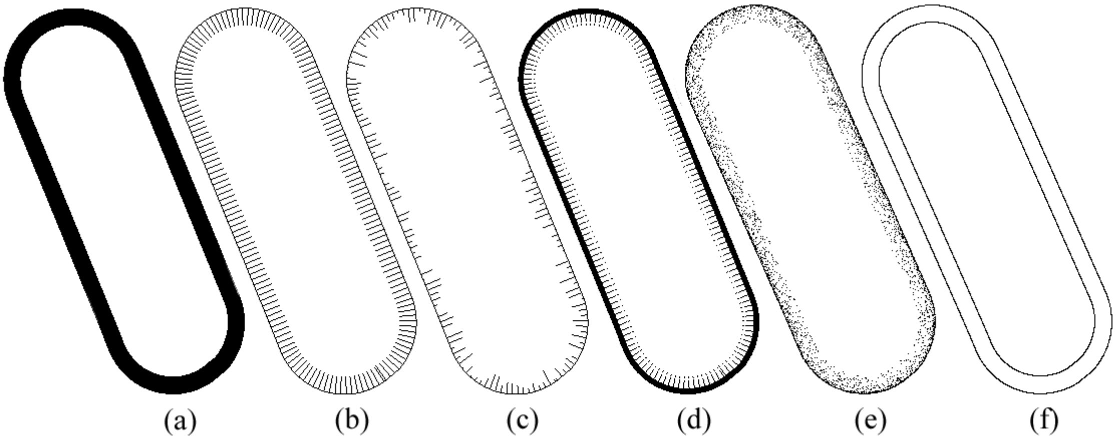

Figure 4.3: The stroke styles simulated in this system. From left to right: ink-filled, serrated-edge, serrated-edge random, serrated-edge with ink-filled, stippling and double line.

The system also successfully extracts short interior strokes. This is done by selecting groups of particles and are stylizing them in various directions. Furthermore, these strokes can be curved to follow the surface based on curvature. Figure 4.4 illustrates a simple case of stroke extraction and stylization. The top-left pear in this image displays raw particles from the NPRBT with convex particles in orange and concave particles in blue. The rest of the images display a subset of these particles based on the particle's angle from the silhouette using various stroke directions. Figure 4.5 illustrates the effect of using more numerous, shorter strokes. Figure 4.6 displays the results of using shorter, thinner strokes selected based on lighting information for a peanut shape.

The system experiences a large speed improvement over standard polygonization and raytracing approaches. The stalagmite model in Figure 4.7 takes 5 seconds to initialize 15000 particles and continuously extracts strokes at 2 FPS afterwards. In comparison, the ray-traced image the same model requires about 30 minutes, a good polygonization takes about 3 minutes and rough polygonization requires about 30 seconds to generate.

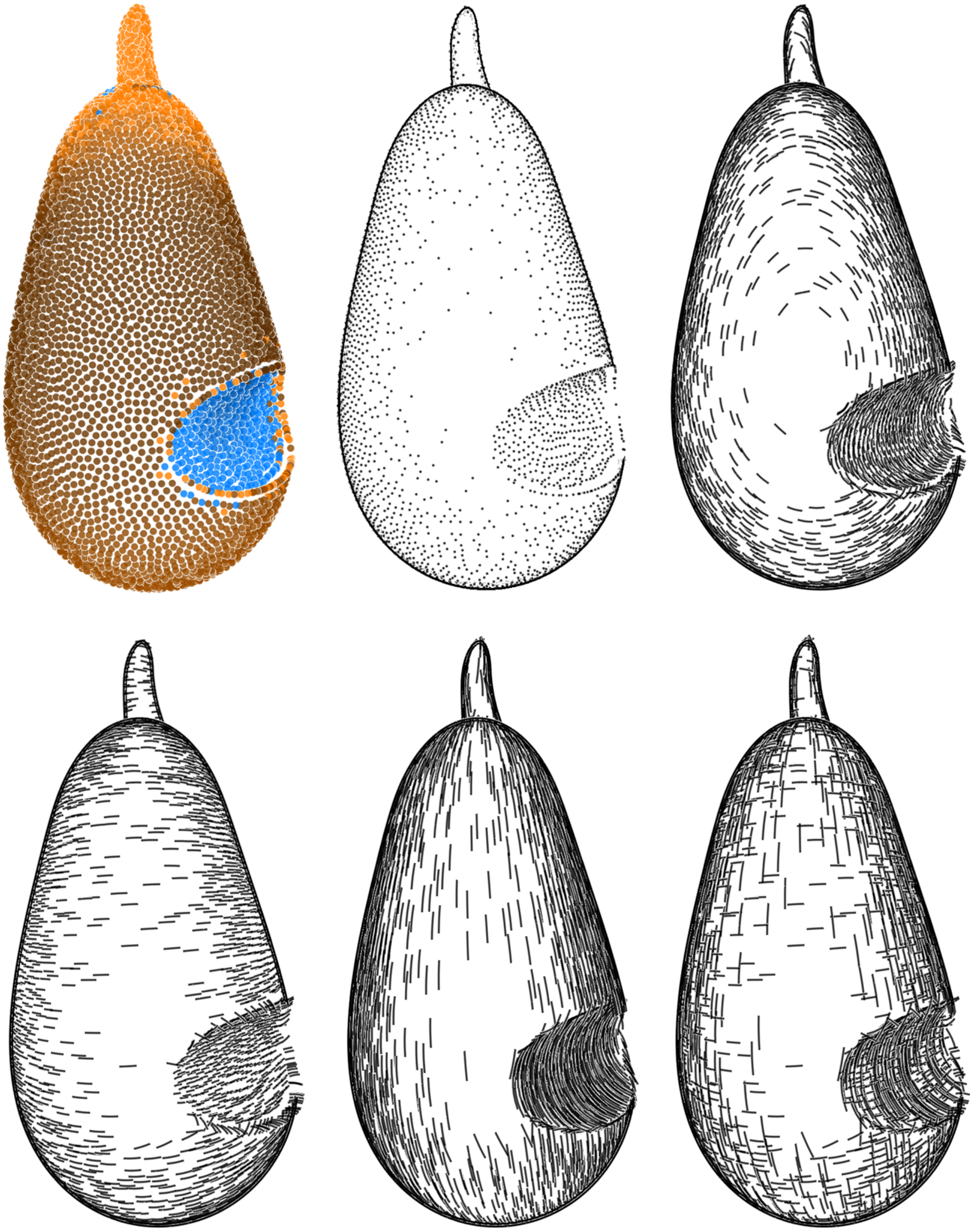

Figure 4.4: Examples of creating different types of interior strokes for a pear.

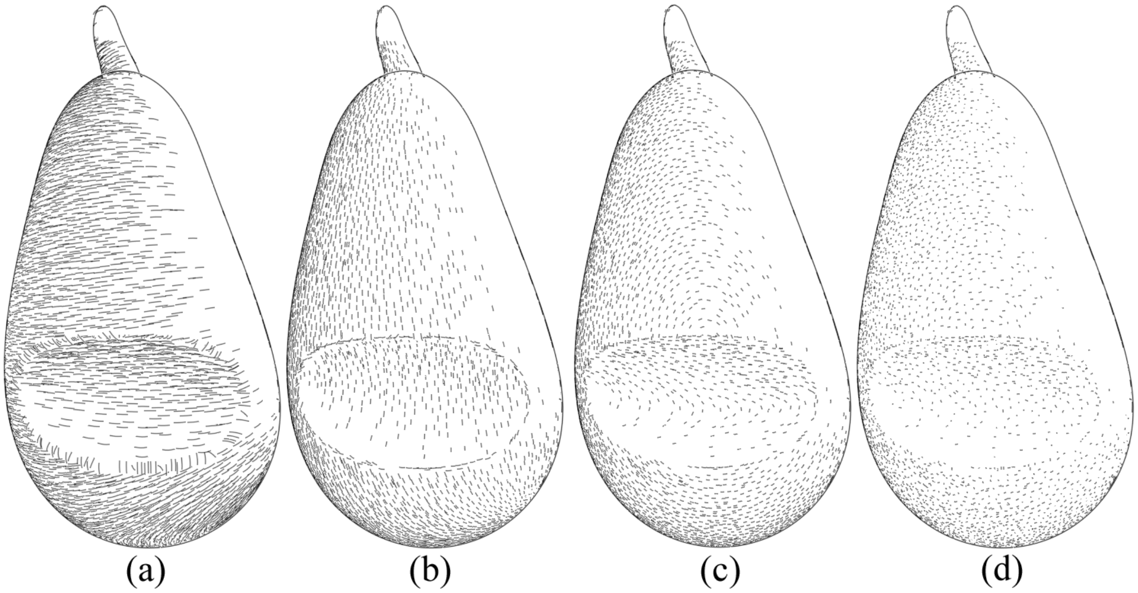

Figure 4.5: Short strokes in various directions for the pear shape.

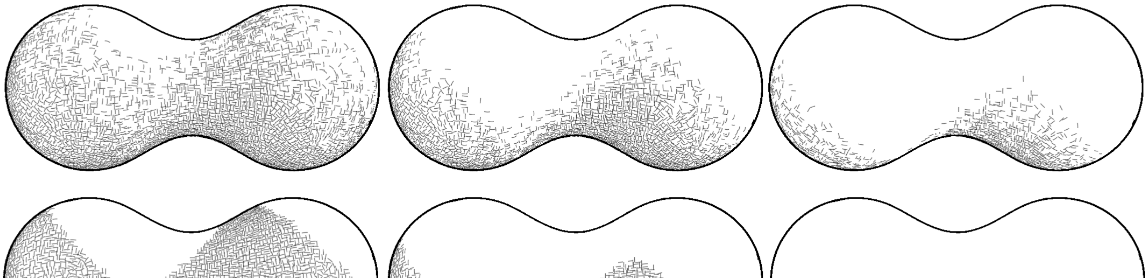

Figure 4.6: Varying distribution of interior strokes based on a light shining from the top right of the peanut shape.

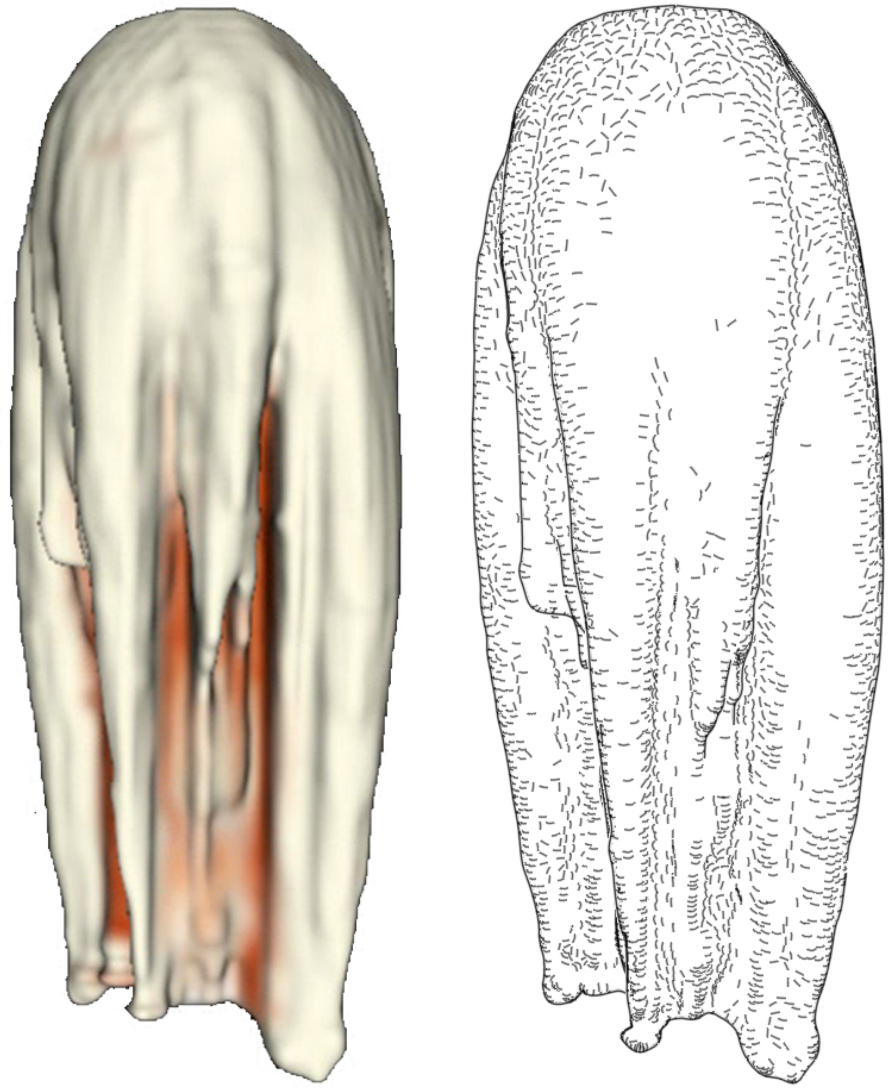

Figure 4.7: A ray-traced image and a pen-and-ink image generated for a stalagmite model, which is created with over 100,000 primitives.

Bibliography

[Akl98] E. Akleman. Implicit surface painting. In *Implicit Surfaces 98*, 1998.

[BH98] D. Bremer and J.F. Hughes. Rapid approximate silhouette rendering of implicit surfaces. pages 1–10, 1998.

[Boi95] E. Boix. *Approximation lineaire des surfaces de* $\Re^3$ *et applications*. PhD thesis, Ecole Polytechnique, France, 1995.

[BS00] J.W. Buchanan and M.C. Sousa. The edge buffer: A data structure for easy silhouette rendering. In *Proc. of NPAR '00*, pages 39–42, 2000.

[Cra00] W. Crane. *Line Form*. George Bell Sons, London, 1900.

[DHvOS00] O. Deussen, S. Hiller, C. van Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings. In *Proc. of Eurographics '00*, pages 40–51, 2000.

[DS00] O. Deussen and T. Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proc. of SIGGRAPH '00*, pages 13–18, 2000.

[Elb98] G. Elber. Line art illustrations of parametrix and implicit forms. In *IEEE Transactions on Visualization and Computer Graphics, Vol. 4, No. 1*, pages 71–81, 1998.

[EP00] H. Hoppe E. Praun, A. Finkelstein. Lapped textures. In *Proc. of SIGGRAPH '00*, 2000.

[Eva72] I.S. Evans. General geomorphometry, derivatives of altitude and descriptive statistics. *Spatial analysis in geomorphology*, pages 17–90, 1972.

[GGS$^+$98] A. Gooch, B. Gooch, P.J. Sloan, P. Shirley, and R. Riesenfeld. A non-photorealistic lighting model for automatic technical illustration. In *Proc. of SIGGRAPH'98*, pages 447–452, 1998.

[GIHL00] A. Girshick, V. Interrante, S. Haker, and T. Lemoine. Line direction matters: An argument for the use of principal directions in 3d line drawings. In *Proc. of NPAR 2000*, 2000.

[GSG$^+$99] B. Gooch, P.J. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. Interactive technical illustration. In *1999 ACM Symposium on Interactive 3D Graphics*, pages 31–38, 1999.

[Hod89] E.R.S. Hodges. *The Guild Handbook of Scientific Illustration*. Van Nostrand Reinhold, 1989.

[HS99] J. Hamel and T. Strothotte. Capturing and re-using rendition styles for non-photorealistic rendering. In *Proc. of Eurographics '99*, pages 173–182, 1999.

[HZ] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *Proc. of Siggraph 2000*, pages 517–526.

[IHS02] T. Isenberg, N. Halper, and T. Strothotte. Stylizing silhouettes at interactive rates: From silhouette edges to silhouette strokes. In *Proc. of Eurographics 2002*, 2002.

[LME$^+$02] A. Lu, C. Morris, D. Ebert, P. Rheingans, and C. Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proc. of IEEE Visualization '02*, 2002.

[MH93] H. Mitasova and J. Hofierka. Interpolation by regularized spline with tension: II. application to terrain modeling and surface geometry analysis. *Mathematical Geology*, 25:657–669, 1993.

[MKT$^+$97] L. Markosian, M.A. Kowalski, S.J. Trychin, L.D. Bourdev, D.Goldstein, and J.F. Hughes. Real-time nonmphotorealistic rendering. In *Proc. of SIGGRAPH '97*, pages 415–420, 1997.

[NM00] J.D. Northrup and L. Markosian. Artistic silhouettes: A hybrid approach. In *Proc. of NPAR 2000*, 2000.

[OEMP03] T. Strotthote O. E. M. Pastor, B. Freundenberg. Animated, real-time stippling. In *IEEE Compter Graphics and Applcations (Special Issue on Non-Photorealistic Rendering)*, 2003.

[PHWF01] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-time hatching. In *Proc. of SIGGRAPH '01*, pages 581–586, 2001.

[Raw87] P. Rawson. *Drawing*. University of Pennsylvania Press, 1987.

[RKS00] C. Rossl, L. Kobbelt, and H. Seidel. Line art rendering of triangulated surfaces using discrete lines of curvature. In *Proc. of WSCG 2000*, pages 168–175, 2000.

[Sec02] A. Secord. Random marks on paper: Non-photorealistic rendering with small primitives. Master’s thesis, Department of Computer Science, University of British Columbia, October 2002.

[SGG$^+$00] P.V. Sander, X. Gu, S.J. Gortler, H. Hoppe, and J. Snyder. Silhouette clipping. In *Proc. of SIGGRAPH 00*, pages 327–334, 2000.

[Sim92] G. Simmons. *The Technical Pen.* Watson-Guptill Publications, 1992.

[SP] M.C. Sousa and P. Prusinkiewicz. A few good lines: Suggestive drawing of 3d models. *Computer Graphics Forum (Proc. of EuroGraphics ’03)*, 22(3).

[SSM02] P.A. Shary, L.S. Sharaya, and A.V. Mitusov. Fundamental quantitative methods of land surface analysis. *Geoderma*, 107(1-2):1–32, 2002.

[Whi94] S. Whitaker. *The Encyclopedia of Cartooning Techniques*. Running Press, Philadelphia, 1994.

[Woo96] J. Wood. *The Geomorphological Characterisation of Digital Elevation Models*. PhD thesis, University of Leicester, UK, 1996.

[WS] G. Winkenbach and D.H. Salesin. Computer-generated pen-and-ink illustration. In *Proc. of SIGGRAPH ’94*.

[YKM99] P. Yuen, N. Khalili, and F. Mokhtarian. Curvature estimation on smoothed 3-d meshes. In *Proc. of British Machine Vision Conference ’99*, pages 133–142, 1999.