Isophote Distance: A Shading Approach to Artistic Stroke Thickness

Todd Goodwin Ian Vollick Aaron Hertzmann
University of Toronto

Abstract

This paper presents an approach for determining stroke thickness in computer-generated illustrations of smooth surfaces. We assume that dark strokes are drawn to approximate the dark regions of the shaded surface. This assumption leads to a simple formula for thickness of contours and suggestive contours; this formula depends on depth, radial curvature, and light direction in a manner that reproduces aspects of thickness observed in hand-made drawings. These strokes convey local shape and depth relationships, and produce appealing imagery. Our method is simple to implement, provides temporally-coherent strokes, and runs at interactive rates.

1 Introduction

Stroke width and style are key features of art, comics, and technical illustration. Line style adds life and clarity to drawings [McCloud 2006], and provides cues to 3D shape [Guptill 1997; Hodges 2003]. The effects of stroke thickness are subtle but important: although viewers may not pay much attention to line quality, a drawing with a plain or inappropriate stroke style can be lifeless or confusing. Current methods in Non-Photorealistic Rendering (NPR) typically use constant-thickness strokes or simple tapering rules that do not capture the liveliness of hand-made drawings and illustration. Hence, determining artistic stroke thickness remains an open problem. How do artists choose stroke thickness, and how can this be modeled in a computational framework?

In this paper, we propose simple, effective rules for determining stroke thicknesses for computer-generated paintings and illustrations. We introduce a quantity called *isophote distance* that represents the thickness of a rim shadow at a contour or suggestive contour. Our basic observation is that *hand-drawn stroke thicknesses often have the qualitative properties of isophote distance.* This leads to a rule for determining stroke thickness: namely, thickness can be determined as a function of how quickly the shading "falls off" from the curve. Based on these observations, we describe algorithms for rendering thick strokes.

Several other approaches to stroke thickness appear in the literature, but previous methods have provided little or no concrete justification in terms of hand-made artwork. In general, the distinctions are subtle, and evaluating stroke thickness is quite tricky; how should we choose among different thickness formulae? In this paper, we identify several qualitative properties of drawings shared by hand-made artwork and our algorithm; no previous method captures these properties. This provides support for our method as a theory of how artists determine stroke thickness. The theory is incomplete — it does not precisely reproduce any particular artwork — but it does provide a parsimonious explanation for many qualitative properties of hand-made artworks, and also points the way to better theories in the future. Because our framework is relatively simple, there are very few "tuning" parameters. Our method is currently designed to work only for smooth surfaces without creases or boundaries, although we believe that it could be generalized to these cases in the future.

1.1 Relation to Previous Work

Most NPR systems employ strokes of some kind to convey shape. Commonly-used strokes include contours, creases, hatching, and suggestive contours [DeCarlo et al. 2003; DeCarlo et al. 2004; Elber and Cohen 1990; Gooch et al. 1999; Grabli et al. 2004; Hertzmann and Zorin 2000; Markosian et al. 1997; Saito and Takahashi 1990; Sousa and Prusinkiewicz 2003; Winkenbach and Salesin 1994]. While these strokes provide effective choices of lines to draw, less is known about how to determine stroke thickness. One approach is to employ constant thickness, which yields a somewhat mechanical appearance. Simple 2D functions, such as tapering at endpoints, can be very pleasing visually; Grabli et al. [2004] demonstrate several such functions. However, 2D functions do not convey 3D shape information.

Existing methods provide several options for determining stroke thickness based on 3D shape. Gooch et al. [1999] suggest making thickness inversely proportional to object depth. Depth-based thickness conveys overall depth relationships between parts of a scene, but not detailed local shape. Sousa and Prusinkiewicz [2003] make contour line thickness *proportional* to mean curvature or maximum curvature. On the other hand, Bremer and Hughes [1998] suggest placing strokes near contours with spacing *inversely proportional* to radial curvature (they do not discuss stroke thickness directly). A few authors suggest using thicker strokes in shadowed regions [Grabli et al. 2004; Schlechtweg et al. 1998; Sousa and Prusinkiewicz 2003], although few details are given. There is little existing evaluation or basis for choosing among these methods. These methods each provide some desirable properties and not others; to what extent do these choices correspond to hand-made styles? As we will show, isophote distance is related to each of these methods, and captures a superset of the properties of real artwork present in previous methods. Our framework also determines thicknesses for suggestive contours, where simple curvature-based methods would be unreliable.

Methods for hatching on surfaces normally use shading to determine hatch density (e.g., [Hertzmann and Zorin 2000; Saito and Takahashi 1990; Winkenbach and Salesin 1994]). Our method bears some similarity to the isophote hatching directions employed by Markosian et al. [1997]. The environment mapping approach to rendering contours produces renderings very similar to our approach [Gooch et al. 1999], which draws in black all surface points for which the dot product of the surface normal and the view vector is below a threshold. However, varying thickness has generally been viewed as a defect [Kindlmann et al. 2003]; this may be because the pure thresholding approach does not produce identifiable strokes (e.g., stroke thickness cannot easily be thresholded to a range of valid widths).

2 Isophote Distance

Our work begins with the following basic intuition for drawing strokes to represent surfaces. Consider a smooth, white, Lambertian

Lambertian shading | Thresholded shading | Isophote distance strokes | Isophotes

Figure 1: *The idea of our work is to use isophote distance for stroke thickness, in order for the strokes to approximate the dark regions of a surface. An isophote is a curve consisting of all surface points with a specific intensity.*

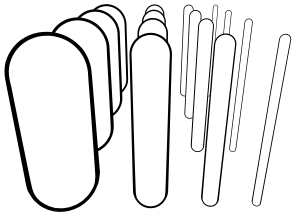

Figure 2: *Cylinders rendered using isophote distance strokes. The stroke thickness depends both on curvature and depth.*

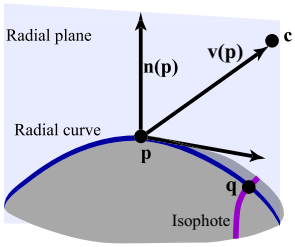

Figure 3: *The radial plane at a surface point* $\mathbf{p}$ *is the plane containing the normal direction* $\mathbf{n}(\mathbf{p})$ *and the camera* $\mathbf{c}$*. The radial curve is the intersection of the surface with the radial plane. The radial curvature* $\kappa(\mathbf{p})$ *is the curvature of the radial curve at* $\mathbf{p}$*. The isophote is the set of points with reflected intensity* r_0*. The isophote distance is the image-space distance between* $\mathbf{p}$ *and* $\mathbf{q}$*.*

object viewed by a perspective camera, and lit by a headlight, i.e., a point light source at the focal center. As can be seen in Figure 1, contours represent global minima of the reflected intensity (i.e., the contours are the set of points with intensity zero), and the suggestive contours represent local minima of intensity [DeCarlo et al. 2003]. Thresholding the image creates dark *rim shadows* near the contours and suggestive contours. (These rim shadows are essentially the inverse of *rim lighting*, often used by photographers and cinematographers.) (This shading model is roughly equivalent to rim lighting with a ring of distant light sources perpendicular to the camera axis, but with inverted intensities. With enough light sources, points near the contours and suggestive contours will be dramatically brighter than the rest of the surface, leading to a natural thresholding effect when the image is rendered to a normal dynamic range.) Suppose we wish to illustrate this thresholded rendering by painting a few black brush strokes on a white background. A natural solution is to draw strokes along the contours of the object (since they are black curves in the rendering), and along suggestive contours (since they are curves that lie in dark regions as well). Moreover, the thicknesses of the brush strokes should match the shading of the object: we do not want the black paint to cover any surface point with intensity less than some threshold r_0. Hence, the stroke should fill the space between the contour (or suggestive contour) and the nearest isophote of r_0. (An isophote is a curve on the surface with constant intensity). In other words, the stroke thickness at a point on the surface should be equal to the image-space distance to the nearest isophote in image space. These thicknesses depend both on the depth of the object and the local shape of the object. For example, a large cylinder will have relatively thick rim shadows, and large distances between the contours and the isophotes, and, as a result, will be drawn with thick strokes (Figure 2). A small cylinder, or a cylinder far in the distance, will have thin rim shadows in image space, and be drawn with thin strokes. The stroke thickness will be clamped to the limited range of thicknesses possible with the brush or pen.

This is essentially a theory of how line drawing works: line drawings simulate the appearance of an object under specific shading conditions. We will not attempt to prove or disprove the theory here; we only note that it provides a unified (though incomplete) explanation for strokes and their thicknesses.

2.1 Definition and Approximations

We now formalize the intuition above. We denote the camera focal center $\mathbf{c}$, and the light source position $\mathbf{s}$. The reflected intensity of a point $\mathbf{p}$ on the surface is given by $r(\mathbf{p}) = \mathbf{n}(\mathbf{p}) \cdot \ell(\mathbf{p})$, where $\mathbf{n}(\mathbf{p})$ is the outward-facing surface normal at point $\mathbf{p}$, and $\ell(\mathbf{p})$ is the unit vector to the light source: $\ell(\mathbf{p}) = (\mathbf{s}-\mathbf{p})/||\mathbf{s}-\mathbf{p}||$. For each point $\mathbf{p}$ on a contour or suggestive contour, we define a quantity called *isophote distance*, denoted $d(\mathbf{p})$. The normalized view vector is $\mathbf{v}(\mathbf{p}) = (\mathbf{c}-\mathbf{p})/||\mathbf{c}-\mathbf{p}||$. Following DeCarlo et al. [2003], we define the *radial plane* of a point $\mathbf{p}$ as the plane that contains $\mathbf{p}$ and has normal $\mathbf{n}(\mathbf{p}) \times \mathbf{v}(\mathbf{p})$. The *radial curve* is the intersection of the surface with the radial plane (Figure 3).

The isophote distance at a point $\mathbf{p}$ is defined as the image-space distance from $\mathbf{p}$ to the nearest surface point $\mathbf{q}$ that satisfies the following three conditions. First, $\mathbf{q}$ must satisfy $r(\mathbf{q}) = r_0$, where r_0 is a user-defined constant. Second, $\mathbf{q}$ must lie in the radial plane of $\mathbf{p}$. This requirement is made because our stroke rendering algorithm uses the projected surface normal as the stroke's spine). In other words, these first two requirements state that $\mathbf{q}$ lies at the intersection of the radial curve and an isophote. Third, $\mathbf{q}$ must lie on the "near part" of the radial curve with respect to $\mathbf{p}$. Specifically, sup-

pose we find $\mathbf{q}$ by traversing the mesh along the radial curve. There are two possible directions we could traverse along this curve, one towards the camera and one away from it; we require that $\mathbf{q}$ can be found by tracing toward the camera. If we were to trace away from the camera from a contour point $\mathbf{p}$, then we would be tracing over a hidden region of the surface. However, for simplicity, we do not require that $\mathbf{q}$ be visible, or that the entire curve from $\mathbf{p}$ to $\mathbf{q}$ be visible.

Given these definitions, our basic rule for stroke thickness is simple: *stroke thickness is equal to isophote distance,* clamped to the range $[T_{min}, T_{max}]$, where T_{min} and T_{max} are user-defined limits.

In order to understand the properties of isophote distance, we can derive an approximation that uses local surface properties. Let the *radial curvature* $\kappa(\mathbf{p})$ be the normal curvature in the radial plane at $\mathbf{p}$, and f be the camera focal length. Then, at any contour point $\mathbf{p}$ with $\kappa(\mathbf{p}) > 0$ and $\mathbf{c} = \mathbf{s}$, the isophote distance can be approximated by:

$$d_C(\mathbf{p}) = \frac{f}{\kappa(\mathbf{p})||\mathbf{p}-\mathbf{c}||}\left(\frac{r_0^2}{2(1-r_0^2)}\right) \tag{1}$$

This approximation makes explicit the relationship of isophote distance to depth, focal length, and radial curvature.

In the more general case of an arbitrary light position $\mathbf{s}$, we have

$$d_C(\mathbf{p}) = \frac{f}{\kappa(\mathbf{p})||\mathbf{p}-\mathbf{c}||}\left(\frac{a^2}{2}\right) \tag{2}$$

where:

$$\begin{aligned} c &= \ell(\mathbf{p})\cdot\mathbf{n}(\mathbf{p}) \\ b &= \ell(\mathbf{p})\cdot\mathbf{v}(\mathbf{p}) \\ a &= \frac{r_0\sqrt{b^2+c^2-r_0^2}-bc}{b^2-r_0^2} \end{aligned}$$

These approximations are derived in Appendix A. The cases $b^2 + c^2 < r_0^2$ and $a < 0$ are handled by setting $d_C(\mathbf{p}) = T_{max}$ and $d_C(\mathbf{p}) = T_{min}$, respectively.

2.2 Computing Thick Strokes

In this section, we describe an algorithm to render strokes using isophote distance, given a polygonal mesh as input. The algorithm has the following steps:

1. Compute visible contours and suggestive contours. Piecewise-linear approximations to these curves are computed as in Hertzmann and Zorin [2000] and DeCarlo et al. [2003; 2004]

2. Compute isophote distances. Isophote distance is computed for each point on each curve, providing a target thickness for each point.

The most general procedure for determining the isophote distance is to traverse the radial curve from $\mathbf{p}$ until a point $\mathbf{q}$ on the isophote is found. This traversal begins in the direction along the radial curve toward the viewer, ensuring that a visible portion of the curve is used. As with contours and suggestive contours, the radial curve and isophote are represented as piecewise-linear approximations to zero sets. The point $\mathbf{q}$ is the intersection of these zero sets. The image-space distance between $\mathbf{q}$ and $\mathbf{p}$ is denoted $d_T(\mathbf{p})$.

An alternative method for computing isophote distance is to employ the approximations in Equations 1 or 2; this approach, as it requires only local computations, is potentially much faster. However, since radial curvature appears in the denominator, the approximations cannot be used directly for suggestive contours (where $\kappa(\mathbf{p}) = 0$), or for contour points with small $\kappa(\mathbf{p})$. In these regions, the radial curve is nearly flat, and both approximations give poor estimates of isophote distance; stroke thickness will frequently saturate at T_{max}. We can alleviate this effect by replacing the radial curvature $\kappa(\mathbf{p})$ with $\kappa(\mathbf{p}) + \varepsilon$ (for some small constant ε), and employing mesh traversal for very small $\kappa(\mathbf{p})$. In order for the thickness to vary smoothly, we linearly interpolate between $d_T(\mathbf{p})$ and the approximations $d_C(\mathbf{p})$ when κ is between two small thresholds κ_{min} and κ_{max}.

These computation strategies are compared in Figure 8. All other figures in the paper employ mesh traversal.

3. Generate thick strokes. Thick strokes are generated for each contour and suggestive contour. Stroke thickness is determined as a function of isophote distance for each point. The isophote distance at a point directly specifies a desired thickness for that point. However, typical drawings restrict strokes to a specific range (e.g., as limited by the thickness of the brush or pen); shading beyond this range will typically be conveyed some other way, such as with hatching or washes. The stroke thickness $T(\mathbf{p})$ is computed by clamping the isophote distance to a user-specified range $[T_{min}, T_{max}]$, and then scaling it by a tapering factor w. The tapering factor is used to taper suggestive contours in lighter regions where their thickness would otherwise become unstable. We compute w as defined by DeCarlo et al. [2004]. For all contour points, we define $w = 1$.

A stroke can be defined in terms of a spine and a thickness at each point, and rendered by a sequence of quadrilaterals [Strassmann 1986]. Given a curve (contour or suggestive contour) and specified stroke thicknesses, an obvious choice is to use the curve itself as the spine of the stroke. This method is effective when strokes are very thin. However, for thicker strokes, this method gives a poor approximation to the shading of the surface: when the thickness is large, bulges appear that distort the apparent silhouette of the surface. We could imagine simply filling the region bounded by the curve on one side, and an isophote on the other. However, this is essentially equivalent to toon shading, and does not simulate brush strokes. For example, the stroke thickness may vary wildly, sometimes surpassing the maximum thickness. Moreover, the two boundaries of the stroke will usually have very different shapes: typically, the contour is relatively smooth, whereas the isophote curve may be very "wiggly," a mismatch that would not normally occur with a brush stroke drawn between the two curves.

Instead, we create brush strokes as follows (Figure 4). Let $\mathbf{f}(\mathbf{p})$ be the camera projection of the inward-facing surface normal, normalized to unit length, and let $\hat{\mathbf{p}}$ be the camera projection of $\mathbf{p}$. Then, a spine point is created by extending a vector from $\hat{\mathbf{p}}$ in the direction $\mathbf{f}(\mathbf{p})$ (Figure 4):

$$\mathbf{s}(\mathbf{p}) = \hat{\mathbf{p}} + \alpha\, T(\mathbf{p})\, \mathbf{f}(\mathbf{p}) \tag{3}$$

where α is a constant. Setting $\alpha = .5$ places the stroke to fill the region between the contour and the isophote. However, we notice that doing so slightly "shrinks" the appearance of the surface; in practice, we use $\alpha = .4$ instead (although the difference is extremely subtle). The stroke is then drawn with spine points $\mathbf{s}(\mathbf{p})$, and with thicknesses $T(\mathbf{p})$. This produces a smooth, "brush-like" curve that follows the contour or suggestive contour, with thickness determined by the isophote distance. Additional stylization can be added by smoothing the spines and/or the thicknesses, or by adding procedural noise.

A key feature of our approach is the use of a consistent definition

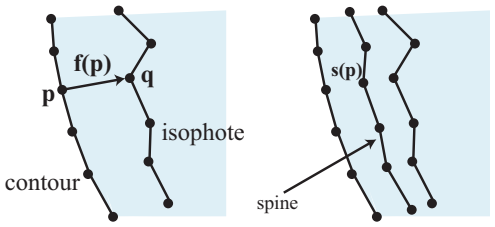

Figure 4: *Left: Each point on a contour and suggestive contour has a corresponding isophote point* $\mathbf{q}$*. The distance between them is* $d(\mathbf{p})$*. Right: A stroke spine is created between the contour and the isophote. This ensures that the stroke lies between the two curves, rather than bulging out of the contour.*

of isophote distance and stroke shape for both contours and suggestive contours. Doing so ensures that contours seamlessly blend into suggestive contours at cusps. For example, one alternative would be to compute isophote distance for suggestive contours by traversing the surface in both directions from the curve, and summing the distances to determine stroke thickness. However, stroke thickness would then double between contours and suggestive at cusps. Similarly, when the surface rotates and a suggestive contour becomes a contour, a temporal discontinuity would occur in the curve thickness; our scheme ensures temporal coherence in stroke thickness.

3 Qualitative Properties

In this section, we examine some of the qualitative properties of our approach, and compare them with images from a variety of sources. One goal of our work is to understand and explain the choices that artists make with regard to stroke thickness. However, artists draw strokes in many different ways [McCloud 2006], and may often be inconsistent — or even deliberately random — in their style. Hence, isophote distance does not provide a full explanation for any existing artwork (nor could any single theory), but, rather, provides a very simple explanation for many qualitative properties observed in some artworks. Nonetheless, the fact that all of the properties listed below arise from the simple concept of isophote distance suggest its value in generating imagery. This suggests that isophote distance could be a useful ingredient in designing many rendering styles (e.g., as a primitive provided by a procedural NPR shader [Grabli et al. 2004]). We will also consider some instruction from art texts and websites. In general, such sources provide coarse, qualitative descriptions and do not tell us how to build algorithms — this is not surprising, as it is often quite difficult to articulate precisely how these choices are made.

Our method for determinining stroke thickness from isophote distance has the following features. (Unfortunately, we are not able to reproduce most of the source artwork to which we compared these observations, due to copyright restrictions.)

1. An object has thicker strokes when it is near the viewer than when it is distant. This follows from the inverse dependence on depth in Equations 1 and 2. This is a widely-observed trait of drawings; Gooch et al. [1999] propose determining stroke thickness solely based on depth. Many illustrations and cartoons with significant depth variations shows examples of this property. This effect may be seen in Figure 2, and between the elephant's front and back legs in Figure 6.

2. Dependence on radial curvature. The following properties follow from the inverse dependence of isophote distance on radial curvature in Equations 1 and 2.

(a) **Large cylindrical objects have thicker strokes than small objects.** This property often manifests as fingers having thinner strokes than arms, which have thinner strokes than legs, which have thinner strokes than torsos, etc., for example, in Figure 11.

(b) **Strokes are thicker at "bulges."** This effect sometimes manifests as biceps being thicker than wrists. Another commonly-seen example is that the upper-cheekbone of a face is typically thinner than the cheek.

(c) **Foreshortened objects are thicker than non-foreshortened objects.** It is difficult to find examples of this property that factor out the confounding effect of depth on thickness.

Note that these examples support the use of an inverse dependence of thickness on curvature, rather than the proportional dependence proposed by Sousa and Prusinkiewicz [2003]; in fact, their hand-drawn French horn example (Figure 2 in their paper) uses thicknesses suggestive of inverse curvature. Rawson [1969] (pp. 109–113) also describes contour shading in medieval European drawing in a manner suggestive of radial curvature: *"... the dark of the contour is accepted symbolically as the ultimate degree of the form's recession from the eye, the limit of its turning away..."*

3. Selective stroke tapering. Current NPR algorithms that taper strokes at the ends do so in 2D (e.g., [Grabli et al. 2004; Hertzmann and Zorin 2000]), without regard to 3D geometry. In our method, tapering occurs only at the ends of suggestive contours; contours are not tapered. This is important when contours become occluded, and is often observed; in both Figures 11 and 12, the hand-drawn contours do not taper when they become occluded.

4. Strokes are limited in size. Artists draw strokes with a limited range of thicknesses. This range is often determined by the media, e.g., pens provide a very limited range of stroke thickness; a brush cannot make strokes thicker than the width of the brush. For example, despite the significant range of curvatures in Figure 12, the technical illustration uses a very narrow range of thickness.

5. Lighting modulates stroke thickness: highlight regions have thinner strokes and shadowed regions have thicker strokes. Contours towards the bottom of objects are frequently thicker, suggesting shadows. This is noticeable in Figure 12, on the right side of the beetle's horn; Hodges [2003] uses this example (and the shadowed regions beneath the body) to illustrate the use of thicker strokes in shadows. Examples of thicker strokes in shadows in NPR algorithms are shown by Grabli et al. [2004].

6. Interior vs. exterior curves. Suggestive contours fall into two classes, anticipation and extension [DeCarlo et al. 2003]. The extension contours extend real contours, whereas anticipation contours live entirely "within" the surface. We often observe that anticipation contours have thinner strokes in drawings, and Scott Ruggels, in his online "Inking Tutorial," seems to suggest this[1]. This arises in our algorithm because isophote distance is typically small in the interior of the surface, when the light source is at the camera position and $\mathbf{v} \cdot \mathbf{n}$ is nonzero. An example is in the cow's eye in Figure 6.

[1] http://www.rdwarf.com/users/ruggels/inks/inks01d.gif; see also http://www.rdwarf.com/users/ruggels/inks/inks01.html.

3.1 Features Not Captured

In this section, we list a few features of thickness that we observe in hand-made drawings that our not handled by our method.

1. Compressed dynamic range. Renderings purely based on isophote distance often reach maximum or minimum thickness in many regions, thereby reaching a constant thickness in these regions. However, hand-made drawings often show thickness variation even in the thickness and thinnest regions.

2. Boundaries and creases. It may be possible to generalize our approach to surface boundaries and creases (e.g., the sides of a building), which we do not currently handle. We do observe that, for example, more distant buildings typically have thinner strokes, and windows have thinner strokes than the silhouette of a building.

4 Styles

In order to demonstrate the value of our approach, we demonstrate its use in several different rendering styles in Figures 6. One style is a basic black-paint style, as outlined in the previous section. We also define a *pen overdraw* style, simulating the appearance of an illustration with many strokes overdrawn over each edge. In this style, strokes follow the contours and suggestive contours, but a procedural offset is added to the curve (based on a sine function with random phase) that varies the shape of the strokes. The amplitude of this offset is proportional to $T(\mathbf{p})$, so that the overdrawn strokes will produce wider regions in areas with larger $T(\mathbf{p})$. A third style simulates a simple *brushed ink* style. Each stroke is composed of many partially overlapping substrokes; the number of which is proportional to $T(\mathbf{p})$. The sub-strokes are also offset by procedural noise to achieve a more hand-painted appearance.

Our system renders the cow model (92864 polygons) at 13 fps on a 3.4 GHz dual-core Intel Xeon; the system is written in C++. However, most of the computation time is spent on line intersection and visibility testing, and we believe that better performance could be achieved with acceleration of these steps.

5 Discussion and Future Work

We have presented an interpretation of artistic strokes based on shading, and demonstrated how it can be used to create artistic imagery that conveys 3D surface shape through stroke thickness. These types of variations can be employed in various artistic styles that make use of strokes. Our main focus has been to show the value of isophote distance in artistic renderings. We have not attempted to fully explore the space of possible styles using isophote distance. For example, our approach could be combined with the stroke stylizations of Grabli et al. [2004] and Sousa and Prusinkiewicz [2003] to produce a much broader space of appealing styles.

We do not currently address thickness for surface boundaries and creases. One possible approach would be to consider to rim shadows observed under finite (area) light sources, rather than infinitesimal point light sources. In some situations, artists also use line thickness to convey shading for multiple light sources, as well as cast shadows [Guptill 1997; Hodges 2003]. It is possible that our method could be generalize to account for both.

Computing isophote distance is potentially expensive, particularly when traversal is required. A variety of possible approaches suggest themself. (We have experimented with a higher-order local surface approximation based on a cubic approximation to third derivatives of the surface, but found it to be too unstable.) Image-space and GPU-based approaches may offer significant speed-ups, but at a possible penalty in accuracy and stylistic control.

Acknowledgments

Our thanks to authors of `rtsc` for making their code available online. We are grateful to Jeff Smith for permission to reproduce his artwork, and to the University of Michigan Museum of Zoology for reproduction permission. This research was supported in part by the Alfred P. Sloan Foundation, the Canada Foundation for Innovation, Microsoft Research, the National Sciences and Engineering Research Council of Canada, the Ontario Ministry of Research and Innovation, and the Ontario Graduate Scholarship Program.

References

BREMER, D. J., AND HUGHES, J. F. 1998. Rapid Approximate Silhouette Rendering of Implicit Surfaces. In *Proc. Implicit Surfaces*.

DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive Contours for Conveying Shape. *ACM Transactions on Graphics 22*, 3 (July), 848–855.

DECARLO, D., FINKELSTEIN, A., AND RUSINKIEWICZ, S. 2004. Interactive rendering of suggestive contours with temporal coherence. In *NPAR 2004*, 15–24.

ELBER, G., AND COHEN, E. 1990. Hidden Curve Removal for Free Form Surfaces. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, vol. 24, 95–104.

GOOCH, B., SLOAN, P.-P. J., GOOCH, A., SHIRLEY, P., AND RIESENFELD, R. 1999. Interactive Technical Illustration. In *Proc. 1999 ACM Symposium on Interactive 3D Graphics*.

GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. 2004. Programmable Style for NPR Line Drawing. In *Rendering Techniques (Proc. EGSR)*.

GUPTILL, A. L. 1997. *Rendering in Pen and Ink*. Watson-Guptill.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. *Proceedings of SIGGRAPH 2000* (July), 517–526.

HODGES, E. 2003. *The Guild Handbook of Scientific Illustration*, 2nd ed. Wiley.

KINDLMANN, G., WHITAKER, R., TASDIZEN, T., AND MLLER, T. 2003. Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proc. Visualization*.

MARKOSIAN, L., KOWALSKI, M. A., TRYCHIN, S. J., BOURDEV, L. D., GOLDSTEIN, D., AND HUGHES, J. F. 1997. Real-Time Nonphotorealistic Rendering. In *SIGGRAPH 97 Conference Proceedings*, 415–420.

MCCLOUD, S. 2006. *Making Comics: Storytelling Secrets of Comics, Manga and Graphics Novels*. Harper.

RAWSON, P. 1969. *The Appreciation of The Art of Drawing*. Oxford University Press.

SAITO, T., AND TAKAHASHI, T. 1990. Comprehensible Rendering of 3-D Shapes. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, vol. 24, 197–206.

SCHLECHTWEG, S., SCHÖNWÄLDER, B., SCHUMANN, L., AND STROTHOTTE, T. 1998. Surfaces to Lines: Rendering Rich Line Drawings. In *Proc. WSCG*.

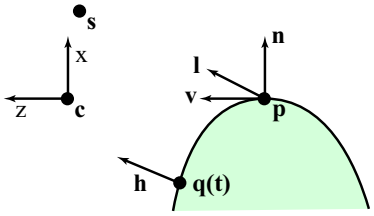

Figure 5: *Parabola used for deriving isophote distance approximation. The figure lies in the radial plane. (The figure is not to scale; the local surface region is assumed to be small relative to the distance to the camera center* **c***.)*

SMITH, J. 1998. *Bone, Volume 5: Rock Jaw: Master of the Easter Border*. Cartoon Books.

SOUSA, M. C., AND PRUSINKIEWICZ, P. 2003. A Few Good Lines: Suggestive Drawing of 3D Models. *Computer Graphics Forum 22*, 3 (Sept.), 381–390.

STRASSMANN, S. 1986. Hairy Brushes. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, vol. 20, 225–232.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer–Generated Pen–And–Ink Illustration. In *Proceedings of SIGGRAPH '94*, 91–100.

A Isophote distance approximation

We now derive an approximation to isophote distance at a point $\mathbf{p}$. We locally approximate the radial curve with a parabola $\mathbf{p}(t)$. We establish a coordinate system for the radial plane, with origin at $\mathbf{p}$, Z-axis pointing toward the camera center $\mathbf{c}$ (that is, coinciding with $\mathbf{v}(\mathbf{p})$, and X-axis aligned with $\mathbf{n}(\mathbf{p})$ (Figure 5). The radial curve is parameterized as $\mathbf{q}(t) = (x,z) = (g(t),t)$ for some quadratic $g(t)$. The unit normal to the parabola is

$$\mathbf{h}(t) = \left(\frac{1}{\sqrt{1-g'^2(t)}}, -\frac{g'(t)}{\sqrt{1-g'^2(t)}} \right) \tag{4}$$

This parabola passes through $\mathbf{p}$ at $t = 0$:

$$g(0) = 0 \tag{5}$$

The parabola has normal $\mathbf{n}(\mathbf{p})$ at $t = 0$:

$$g'(0) = 0 \tag{6}$$

The curvature of the parabola must be $\kappa(\mathbf{p})$ at $t = 0$:

$$g''(0) = -\kappa(\mathbf{p}) \tag{7}$$

These constraints are sufficient to define the quadratic:

$$g(t) = \frac{1}{2} g''(0) t^2 \tag{8}$$

The isophote distance is the image-space distance to the point $\mathbf{q}(t)$ that satisfies:

$$\mathbf{h}(t) \cdot \mathbf{l} = r_0 \tag{9}$$

where

$$\mathbf{l} = (\ell(\mathbf{p}) \cdot \mathbf{n}(\mathbf{p}), \ell(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p})) \tag{10}$$

is the projection of the normalized light vector $\ell(\mathbf{p})$ to the radial plane. We will assume that the light is sufficiently distant that $\mathbf{l}$ is constant over the neighborhood of $\mathbf{p}$. Solving Equation 9 for t gives:

$$t = \left(\frac{\mathbf{l}_z \mathbf{l}_x - r_0 \sqrt{\mathbf{l}_z^2 + \mathbf{l}_x^2 - r_0^2}}{\mathbf{l}_z^2 - r_0^2} \right) / g''(0) \tag{11}$$

Given t, we can compute the x-coordinate of $\mathbf{q}(t)$:

$$q_x = \frac{1}{2} g''(0) t^2 \tag{12}$$

The projected distance between $\mathbf{q}$ and $\mathbf{p}$ is then:

$$d(\mathbf{p}) \approx \frac{f}{p_z} p_x - \frac{f}{p_z} q_x \tag{13}$$

where we have assumed that $q_z \approx p_z$. Moreover, we have $p_z = ||\mathbf{p} - \mathbf{c}||$ and $p_x = 0$. Simplifying the above expression yields Equation 2. Substituting in $\ell(\mathbf{p}) \cdot \mathbf{n}(\mathbf{p}) = 0$ yields Equation 1.

Non-contour points. The same derivation can be applied to non-contour points, yielding:

$$d_C(\mathbf{p}) = \frac{f}{\kappa(\mathbf{p}) ||\mathbf{p} - \mathbf{c}||} \left(\frac{a^2 - e^2}{2\left(1 + e^2\right)^{3/2}} \right) \tag{14}$$

where:

$$\begin{aligned} e &= \frac{\mathbf{n}(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p})}{\sqrt{1 - (\mathbf{n}(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p}))^2}} \\ c &= \frac{\ell(\mathbf{p}) \cdot \mathbf{n}(\mathbf{p}) - (\mathbf{n}(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p}))\,(\ell(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p}))}{\sqrt{1 - (\mathbf{n}(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p}))^2}} \\ b &= \ell(\mathbf{p}) \cdot \mathbf{v}(\mathbf{p}) \\ a &= \frac{r_0 \sqrt{b^2 + c^2 - r_0^2} - bc}{b^2 - r_0^2} \end{aligned}$$

The cases $b^2 + c^2 < r_0^2$ and $a < e$ are handled by setting $d_C(\mathbf{p}) = T_{max}$ and $d_C(\mathbf{p}) = T_{min}$, respectively.

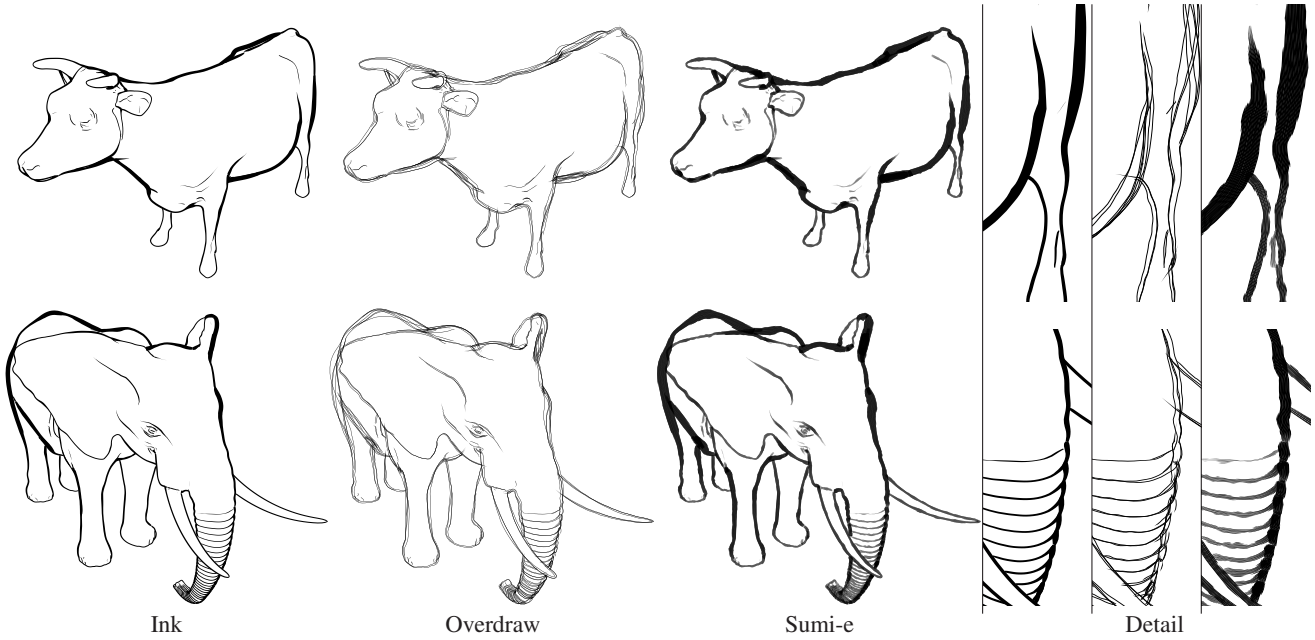

Figure 6: *Left: Three rendering styles using isophote distance strokes. Right: Corresponding stroke details.*

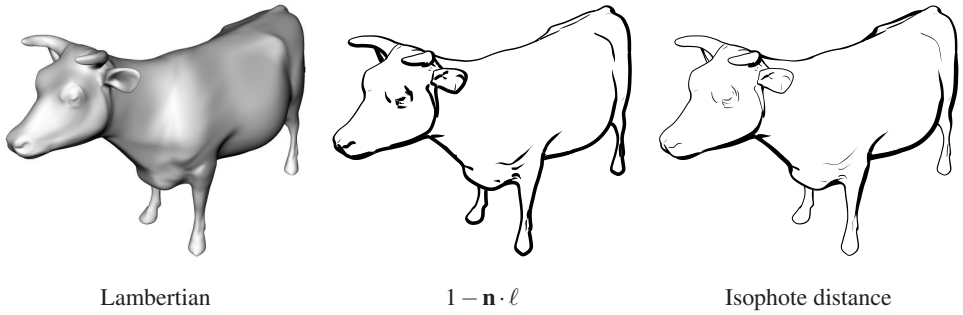

Figure 7: *Moving the light source. The middle column shows stroke thickness set to* $1-\mathbf{n}\cdot\ell$ *and clamped; note that the surface has an "embossed" look, e.g., at the horns. The right column shows our method. The isophote distance thickness conveys more of the local variation of shape, similar to rim shadows.*

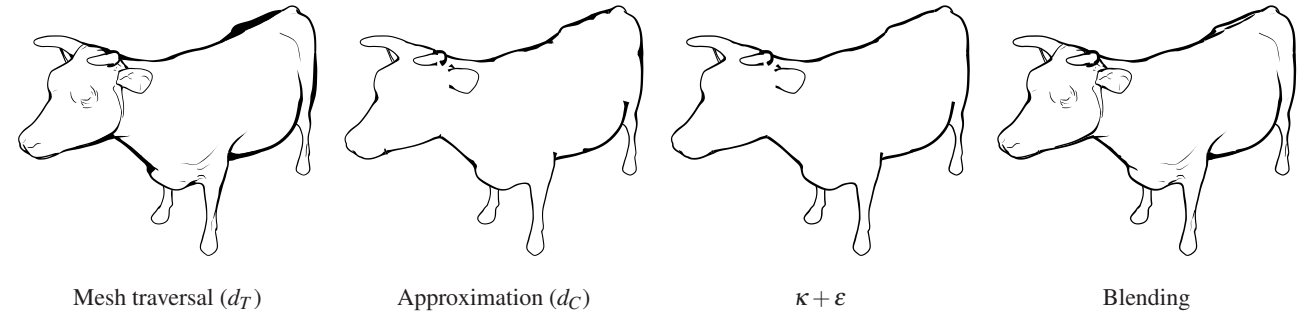

Figure 8: *Different algorithms for computing isophote distance. Mesh traversal is nearly exact, but may be slow. The radial curvature approximation* d_C *is potentially much faster, but inaccurate for small values of* κ*. Replacing* κ *with* $\kappa+\varepsilon$ *yields a smoother result. Using* d_T *for points with small* κ*,* d_C *(with* $\kappa+\varepsilon$*) for large* κ*, and linearly blending in-between yields both accuracy and speed.*

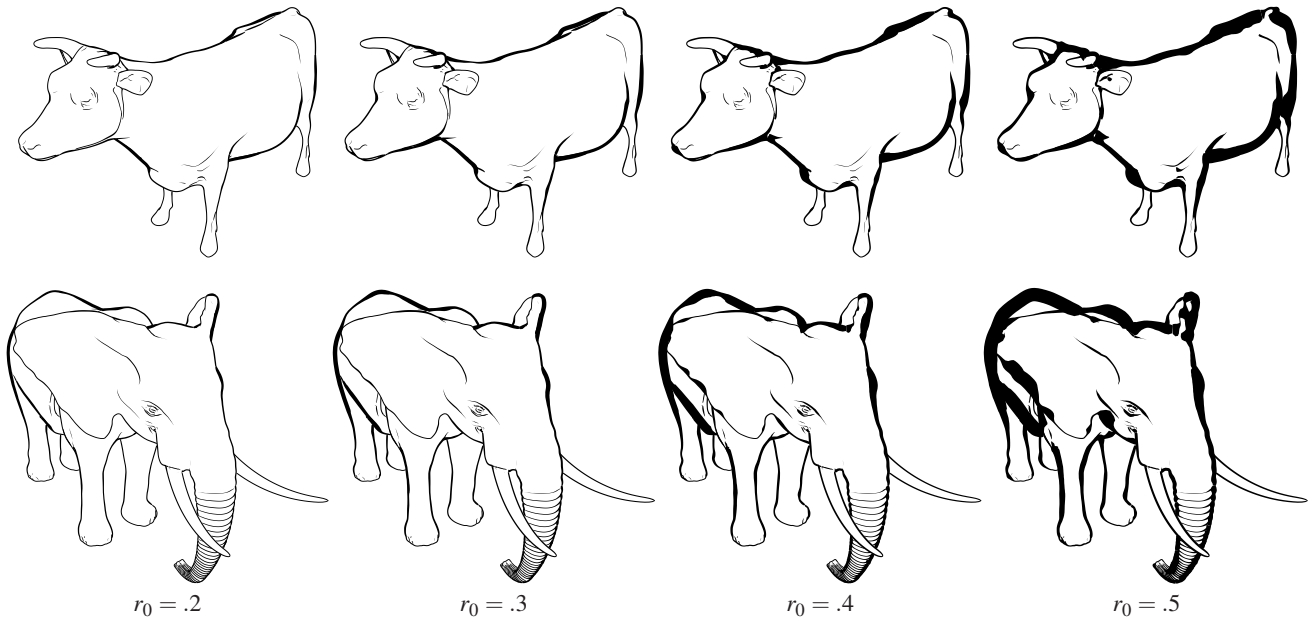

Figure 9: *Effect of varying the r_0 parameter, which selects the isophote.*

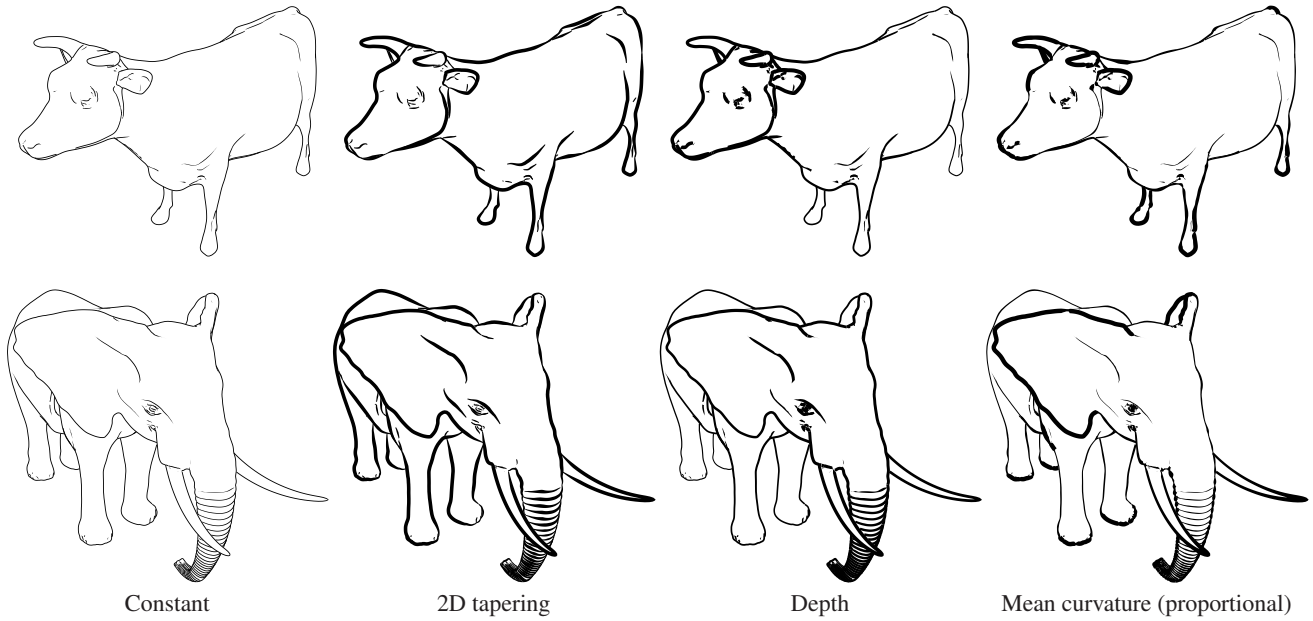

Figure 10: *Comparison with existing methods for stroke thickness. Constant thickness strokes appear plain. 2D tapering ignores local surface shape, and tapers without regard to stroke occlusion. Thickness proportional to mean curvature yields fat strokes for thin regions. (For these surfaces, thickness proportional to max curvature appears nearly identical.)*

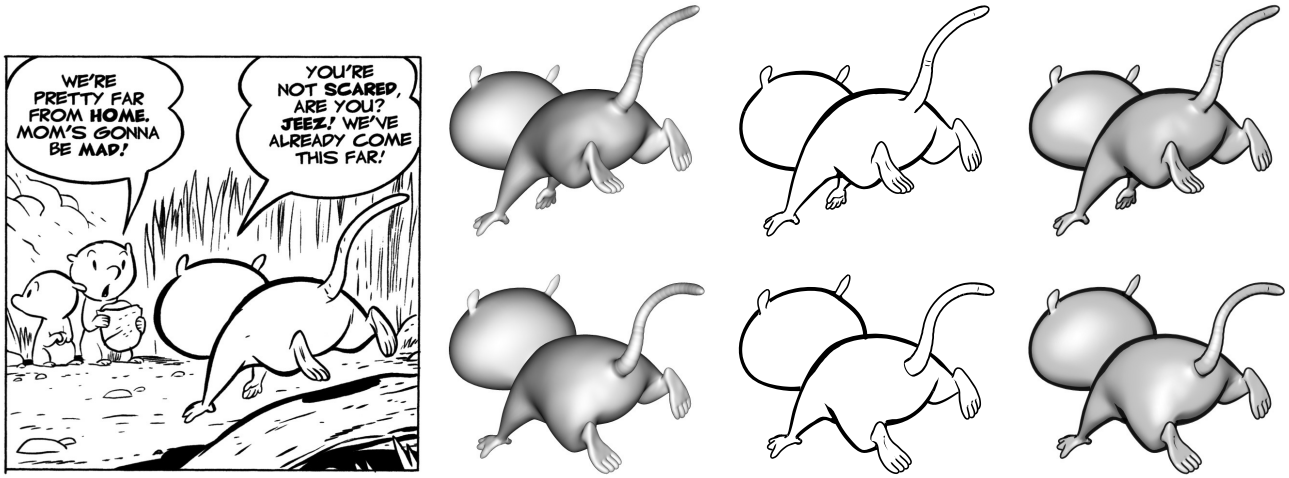

Figure 11: *Rendering based on a comic illustration from "Bone" [Smith 1998]. Based on the drawing, we constructed a 3D model by hand, and then rendered the model using our algorithm. Our result is qualitatively similar to the original drawing. We also show a rendering of the surface rotated slightly. (BONE® is Copyright © 2007 Jeff Smith; image used with permission.)*

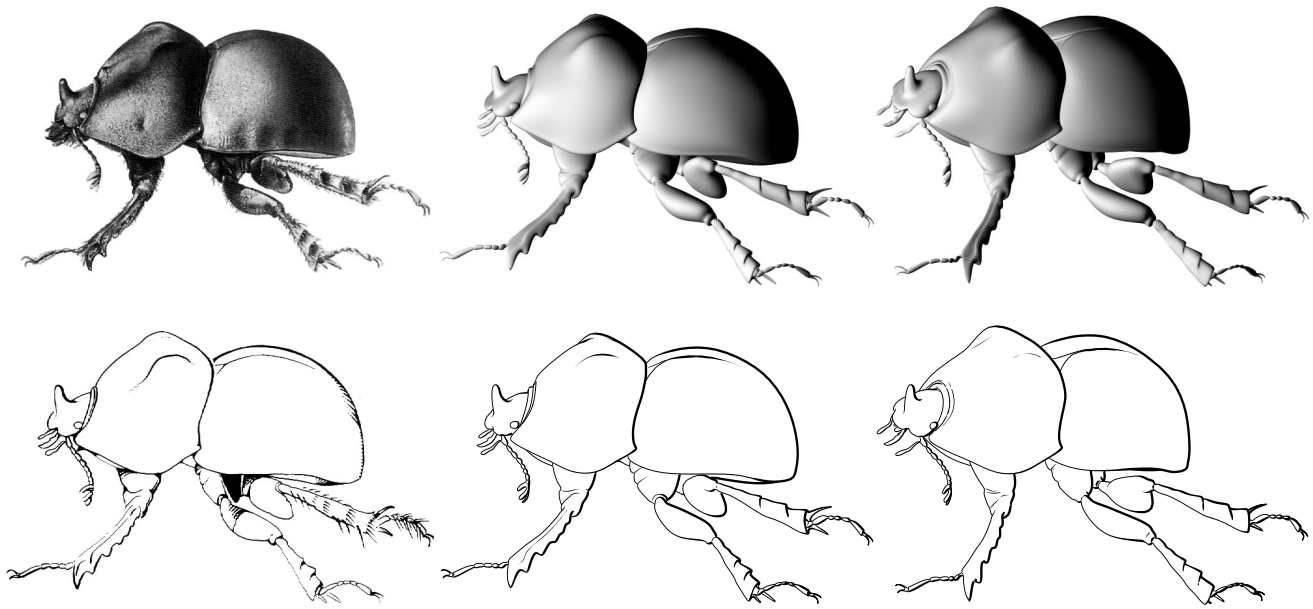

Figure 12: *Left: A tonal drawing of a dung beetle, and a hand-drawn technical illustration based on the same image. The illustration shows a strong dependence of line thickness on shading. Center: Based on the drawings, we constructed a 3D model by hand, and then rendered the model using our algorithm. Our result is qualitatively similar to the original drawing. Right: We also show a rendering of the surface rotated slightly. (Drawings by William L. Brudon, from University of Michigan Museum of Zoology, Misc. Publication No. 54; images used by permission).*

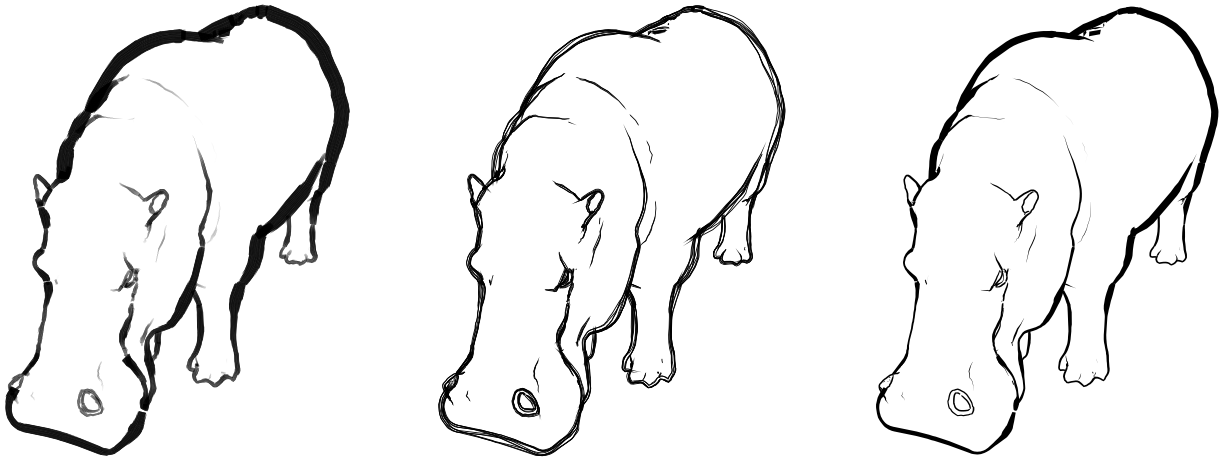

Figure 13: *Hippos rendered in brush, overdraw, and ink styles.*

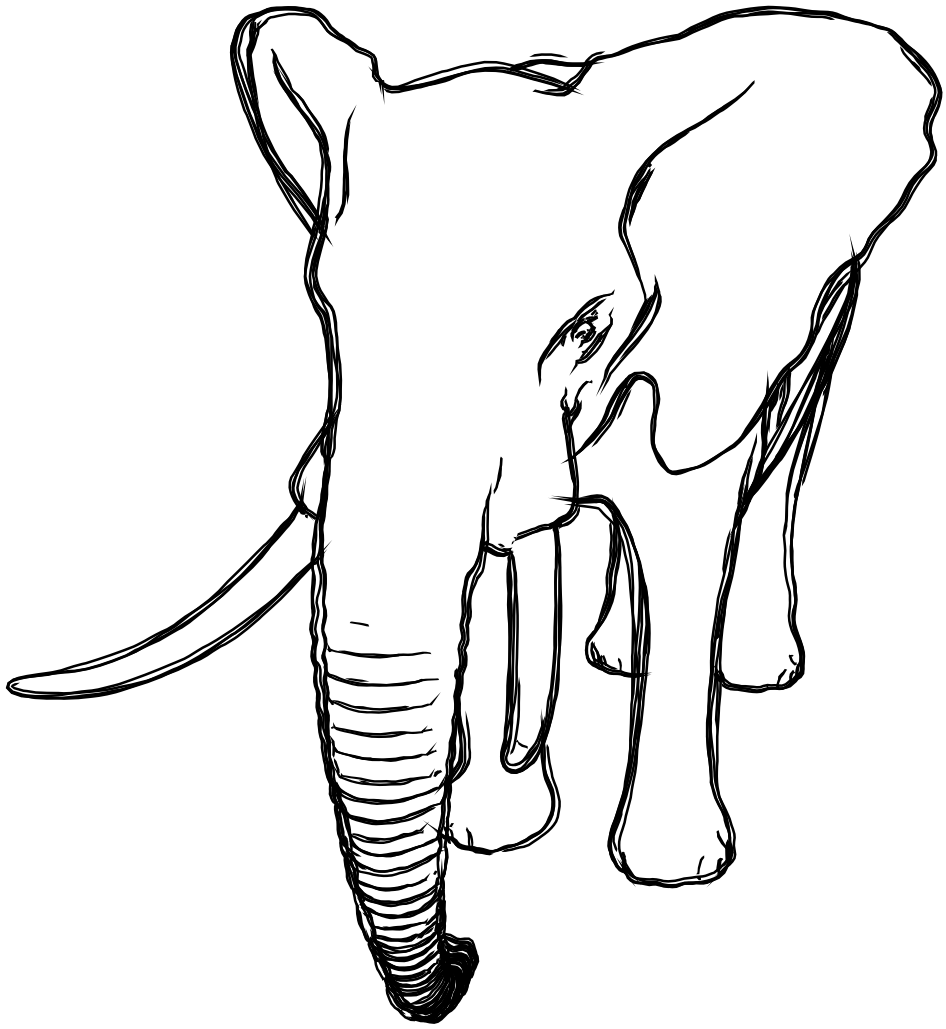

Figure 14: *Elephant rendered in an overdraw style.*