

Where Do People Draw Lines?

Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros,
Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz

Princeton University

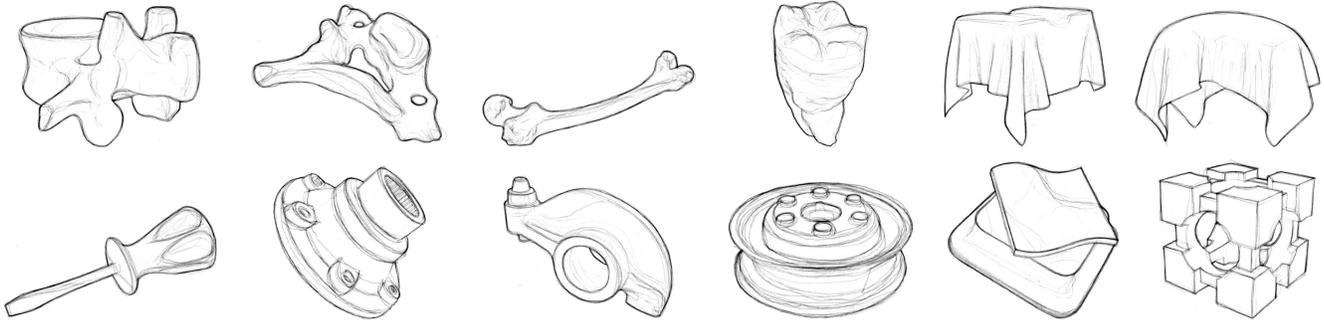


Figure 1: *Where people draw lines.* Average images composed of 107 drawings show where artists most commonly drew lines in our study.

Abstract

This paper presents the results of a study in which artists made line drawings intended to convey specific 3D shapes. The study was designed so that drawings could be registered with rendered images of 3D models, supporting an analysis of how well the locations of the artists' lines correlate with other artists', with current computer graphics line definitions, and with the underlying differential properties of the 3D surface. Lines drawn by artists in this study largely overlapped one another (75% are within 1mm of another line), particularly along the occluding contours of the object. Most lines that do not overlap contours overlap large gradients of the image intensity, and correlate strongly with predictions made by recent line drawing algorithms in computer graphics. 14% were not well described by any of the local properties considered in this study. The result of our work is a publicly available data set of aligned drawings, an analysis of where lines appear in that data set based on local properties of 3D models, and algorithms to predict where artists will draw lines for new scenes.

1 Introduction

The goal of our work is to characterize the mathematical properties of line drawings made by human artists. Specifically, we aim to draw relationships between the locations of lines drawn by artists and properties of the surface geometry, lighting, and viewing conditions at those locations. This type of analysis can both guide the future development of line drawing algorithms in computer graphics, and provide artists and observers with a precise vocabulary for characterizing and discussing where lines on a model are drawn.

This paper describes a study in which art students were asked to make line drawings that “convey the shape” of 3D models shown to them as rendered images. The study balances the competing concerns of allowing the artists to draw freely and of acquiring useful data. Specifically, the artists were asked to draw in two steps: first to draw in a blank area, then to register their drawing to a faint, photo-realistic image of the model. The registered drawings can then be used to study correlations between the locations of human-drawn and computer-generated lines (Section 4.2), characterize the differences between specific artists (Section 4.3), and provide training data for synthesis of new line drawings (Section 4.4).

We provide a statistical analysis of the locations where artists draw lines with the geometric, viewpoint, and lighting characteristics of the underlying 3D scene. The analysis supports several conclusions. First, human line drawings, made under our controlled conditions, are quite consistent with each other. Second, most of the areas where artists consistently drew lines can be described by well-known, simple mathematical properties, such as the locations of occluding contours and large gradients of image intensity. Third, current line drawing definitions can help explain many of the lines that do not lie in those areas, but cannot explain all the artists' lines.

We believe that this paper in no way exhausts the possible investigations that can be performed with this data. We therefore make our drawings and models freely available, in hopes that other researchers continue in this line of inquiry.

Overall, the paper makes the following contributions:

- A study methodology that supports registration of human line drawings with rendered images of 3D models.
- A dataset of 208 line drawings provided by 29 skilled artists covering a dozen 3D models, with two viewpoints and two lighting conditions for each model.
- Results of correlating local properties of 3D surfaces and rendered images with the locations of lines in artists drawings.
- Characterization of which pixels drawn by recent automatic line drawing algorithms are found in human line drawings.
- A method for predicting the probability of an artist drawing at a particular location in an image and using that image to generate new line drawings.

2 Background and Related Work

Principles of Drawing. Artists have for centuries studied the principles of how to make drawings. Books codifying these principles line the shelves of any major bookstore (e.g., [Guptill 1976; Meyer and Avilez 1985; Ruskin 1895]). Although these texts tend to emphasize the more artistically salient concerns of composition, motion, passion, and mystery, some also offer advice on using lines to convey texture, bulk, and shading, noting that even sparse line drawings are sufficient for the viewer to identify shape. Some explicitly identify known line types as candidates for drawing (e.g., contours and ridges [Smith 1997], or specific feature lines on a known shape such as the nose [Peck 1982]). Nevertheless, little more is said about *where* on a figure to place lines in order to best convey shape – this decision making process seems to be learned through trial and error over years of practice by individual artists.

Algorithmic Line Drawing. Inspired by the effectiveness and aesthetic appeal of human line drawings, scientists have investigated algorithms for generating line drawings. The mathematician Felix Klein reportedly [Hilbert and Cohn-Vossen 1999] asked his students to draw parabolic lines (lines of zero Gaussian curvature) over a bust of Apollo, believing they would expose some aspects of the aesthetics of the sculpture. Feature lines play a critical role in some of the early work in non-photorealistic rendering (NPR), for example the work on technical illustrations by Saito and Takahashi [1990] and on pen-and-ink illustration by Winkenbach and Salesin [1994; 1996]. These early examples of NPR, as well as many that follow, make drawings that also include hatching lines or stippling. These effects greatly facilitate the viewer’s understanding of shape by providing diffuse shading, and sometimes by following isoparametric lines or curvatures in the shape. This paper neglects such shading effects in order to focus on how more basic, sparse line drawings are used to convey shape.

More recently, interest has shifted to algorithms for drawing lines on smooth surfaces. Examples include smooth silhouettes [Hertzmann and Zorin 2000], suggestive contours and highlights [DeCarlo et al. 2003; DeCarlo and Rusinkiewicz 2007], geometric ridges and valleys [Ohtake et al. 2004], apparent ridges [Judd et al. 2007], and lines from diffuse shading [Lee et al. 2007]. Each algorithm has strengths and weaknesses. Silhouette lines (more precisely, occluding contours) are obviously important, but alone they tend to leave a drawing too sparse. Suggestive contours connect to occluding contours and often lie in naturally important areas, but do not exist at all on convex shapes. Geometric ridges help define convex shapes, but often seem to exaggerate curvature. Apparent ridges use a more sophisticated view dependent curvature metric, but still seem to exaggerate curvature in some cases and tend to be noisy. Lines from diffuse shading [Lee et al. 2007] are robust and smooth, and respond to changes in lighting, but are difficult to stylize. Finally, while not explicitly designed as a line drawing method, intensity edge detection in shaded images by algorithms like that of Canny [1986] is so straightforward that it is common in image editing software like Photoshop. These lines are surprisingly effective for many applications but require tuning intensity thresholds, are brittle in the face of image noise, and can break and bifurcate.

With the myriad line drawing options now available, it is natural to ask which are appropriate for a specific situation, or which more closely resembles what a human would draw. Recent efforts ([Judd et al. 2007; Lee et al. 2007; DeCarlo and Rusinkiewicz 2007]) include direct comparisons between their results and artists’ renderings. However, these comparisons are informal and generally intended to illustrate the inspiration for the work, not to evaluate the results of the algorithm. In contrast, the dataset presented here makes possible formal comparisons by directly associating 3D models, lighting, and camera angles with human drawings.

NPR by example. The use of human input to drive NPR algorithms has been previously explored by several research teams. Hamel and Strothotte [Hamel and Strothotte 1999] found the parameters of an NPR renderer to match an example rendering. The introduction of “image analogies” by Hertzmann et al. [2001] enabled a user to transfer certain stylistic properties of an image to another image, and the analogous approach was also developed for curves [Hertzmann et al. 2002]. The WYSIWYG NPR system of Kalnins et al. [2002] allowed an artist to draw directly onto a model and then transferred that stylization to other views of the model. Our synthesis approach described in Section 4.4 is most similar to that of Lum and Ma [2005], which also used a machine learning algorithm to classify lines based on a user’s input. However, Lum and Ma focus on an interactive system for specifying a subset of known lines, whereas we focus on synthesizing lines from learned relationships between surface properties and human line drawings.

Evaluation Studies. Experimental psychologists have long used computer generated imagery to study the human visual system, including the study of shape cues given by line drawings [Koenderink et al. 1996]. Objective evaluation studies are difficult to design for NPR imagery, where there is not an obvious ground truth. Nevertheless, there has been substantial work in perceptual studies of lines and NPR algorithms. For example, Girshick et al. [2000] showed that “line direction matters” when depicting shape. Gooch et al. [Gooch et al. 2004] evaluated people’s ability to recognize faces from caricature drawings versus photos. Several recent methods focus on directing the viewer’s attention, and used an eye-tracker to evaluate success [DeCarlo and Santella 2002; Cole et al. 2006; Santella and DeCarlo 2004]. When NPR is used to improve human comprehension, algorithms can be directly compared against each other [Winnemöller et al. 2007], but this approach can only evaluate relative effectiveness between algorithms, not how much such methods could improve.

Two recent studies used drawings by human artists alongside computer renderings of the same models. Isenberg et al. [2006] compared viewers’ perceptions of hand-drawn versus computer-generated pen-and-ink illustrations. Phillips et al. [2005] conducted a study similar to ours, in which artists were asked to draw synthetic, blobby shapes from a range of prompt types. Among other differences from that work, our study includes a separate tracing and registration step that allows greater accuracy in the analysis of artists’ lines.

3 Study Design

The study is designed to capture the relationships between the locations where human artists draw lines and the mathematical properties of the of the model’s surface and appearance at those locations. To achieve this goal in a way that supports detailed analysis, several important choices must be made: what drawing style to consider, what models, views, and lighting conditions to use as prompts, how to present these prompts to the artists, what instructions to give the artists, and how to scan and process the drawings. The following sections describe each of our design decisions in detail.

3.1 Artistic Style

The first challenge in designing the study is to decide on a style of drawing that is narrow enough that all artists have roughly similar intentions while drawing, yet flexible enough for each artist to exercise individual ingenuity.

We balance these goals by focusing on line drawing that includes only feature lines, with no hatching or shading (examples appear in Figure 4). This choice of style was made for two reasons. First,

it is a simple style that is familiar to most artists and yet expressive enough to depict shape. Second, it matches the style generated by several NPR rendering algorithms recently proposed in the computer graphics literature (e.g., [DeCarlo et al. 2003; Judd et al. 2007]). By asking the artists to draw in the same style as the computer algorithms, we can learn both about the human drawings (by using the vocabulary of the algorithms) and the computer drawings (by using statistical correlations with human tendencies).

We give each artist verbal and written instructions to make drawings with “lines that convey the shape” of an object. We do not provide instructions about whether lines should represent shape features, lighting features, or anything else. However, we specifically ask the artists to refrain from including lines that represent area shading or tone features, such as stippling or hatching.

3.2 Prompt Selection

A second design decision is to select 3D models and rendering parameters to use when producing prompts (images depicting a shape for the artists to draw). In making this choice, we use the following design criteria:

- **Comprehension:** our first concern is to provide images from which the artists can easily infer shape. This consideration rules out overly abstract 3D surfaces (i.e., shapes unlike anything in common experience), complicated concave shapes (e.g., with lots of occluded surfaces), and surfaces with spatially-varying BRDFs (e.g., textures). It also suggests that multiple views of the shape be provided as prompts, so that ambiguities in one view are resolved by another. Finally, prompt images should be photorealistic, to avoid confusing artists that are not familiar with classic CG rendering artifacts such as hard shadows and lack of indirect illumination.
- **Coverage:** the set of prompts presented to each artist should have pixels that cover a wide variety of mathematical properties (e.g., high image gradients, surface critical points, etc.). This consideration rules out objects containing only large, planar facets (few interesting surface features), convex objects (no concave surface features), and other surfaces with few inflections. Rather, it suggests blobby objects with many curved surfaces.
- **Separation:** the prompt images should have mathematical features of particular interest (e.g., suggestive contours, apparent ridges) in clearly distinguishable positions within the image. This consideration rules out using headlights (a point light centered at the viewer’s eye), since many interesting image features line up directly with object-space features in that case (e.g., suggestive contours and image intensity valleys).
- **Familiarity:** the objects shown in prompts must be familiar to the artist (so that he/she can understand it), but not so familiar to the that he/she applies domain-specific knowledge when drawing. This consideration rules out objects with strong semantic features (e.g., human faces) and ones commonly drawn in art classes (e.g., fruit).
- **Simplicity:** the objects must be relatively simple, without much fine scale detail. Otherwise, the artists may be tempted to abstract or simply omit important features.

Based on these criteria, we select 12 models of four object types for our study: (a) 4 bones, (b) 2 tablecloths, (c) 4 mechanical parts, and (d) 2 synthetic shapes (Figure 2). We synthesize four prompt images for each model, one for each combination of two different viewpoints and two lighting conditions. The two viewpoints are always 30° apart (so that large parts of each model can be seen from

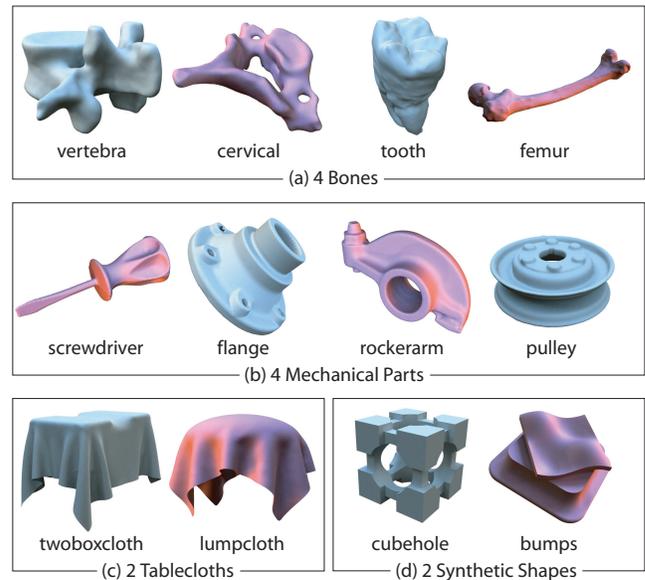


Figure 2: *Prompt models.* The twelve models from our study, shown with one of two views and one of two lighting conditions. Groups (a) and (b) are scanned meshes, (c) and (d) are synthetic.

both viewpoints) and are carefully chosen to distribute surface features across the image. By providing prompts with different lighting and different viewpoints for the same model, we can analyze image-space properties in isolation from object-space ones.

We generate our images using YafRay [2008], a free raytracing package capable of global illumination using monte carlo pathtracing. The models are rendered using a fully diffuse, gray material, and thus take on the color of the lighting environment. For lighting, we use the Eucalyptus Grove and Grace Cathedral high dynamic range environment maps captured by Debevec [1998].

3.3 Line Drawing Registration

The final and most difficult part of the study design is to engineer a system that is able to register line drawings made by artists to pixels of a prompt image with great accuracy.

Designing such a system is challenging because there is a trade-off between allowing the artist to draw in a natural manner (e.g., with pencil on a blank sheet of paper) versus including constraints that facilitate accurate registration between prompts and line drawings. On one hand, the drawing process surely must not bias the locations of lines made by the artist, and thus it is not a good idea to have the artist *compose* a drawing directly over the image prompt. On the other hand, the process must provide enough registration accuracy to distinguish between important mathematical properties at nearby pixels in the prompt. This problem is particularly difficult since free-hand drawings can be geometrically imprecise, and the intended location of every line is only known by the artist.

Our design balances these trade-offs with a simple two step process. In the first step, the artist is given a pencil and a blank sheet of paper and then asked to make a free-hand line drawing that “conveys the shape” of the surface in the prompt. In the second step, the artist is asked to re-create the same line drawing by tracing over a faint copy of the prompt, being careful to redraw every line of the free-hand drawing at the position corresponding to its originally intended location.

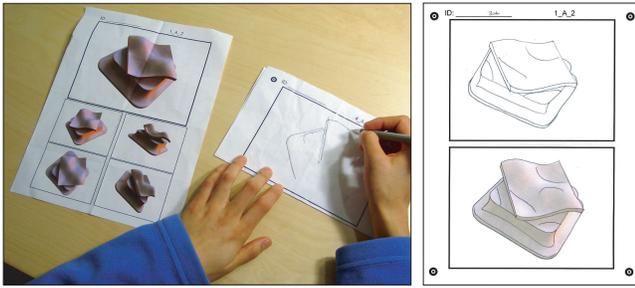


Figure 3: *Making a drawing.* With the drawing page folded in half, the artist makes a free-hand drawing while referring to the prompt page (left). The completed drawing page (right) contains a free-hand drawing and a registered drawing.

Specifically, the artist is given two sheets of paper for each line drawing (Figure 3). The *prompt page* (shown on the left) contains multiple full color views of the prompt shape, one of which is large across the top and is called the *main view*. The *drawing page* (shown on the right) contains two boxes, each the same size as the main view. The top box is initially blank, while the bottom box contains a faint version of the main view.

The artist is asked to complete the drawing page by first folding the page vertically in half so that only the blank space at the top is visible (left of Figure 3). Using the viewing page for reference, the artist draws the prompt shape in the blank space, just as if they were making a normal sketch. When finished, the artist unfolds the drawing page and copies their freehand drawing onto the faint image on the bottom of the same page. During the copying step, the artist is asked to change the shape of their lines to match the target rendering, but not to change the number or relative position of the lines. In effect, the artist is asked to perform a non-linear warp of their original drawing onto the target shape. A typical result is shown on the right side of Figure 3.

We scan the drawing page with a flat-bed scanner, locate fiducials included in the corners of the page, and then use the fiducials to register the traced lines with the 3D model rendered from the main viewpoint. An adaptive thresholding method is used to convert the scanned gray-scale image into a binary image so that all the artist’s lines, regardless of strength, are included in the binary image. We then use a thinning operator to narrow the lines in the binary image down to the width of one pixel. The final result is a 1024×768 pixel binary image containing a single pixel wide approximation of the human artist’s lines.

While this procedure takes up to twice as long as a single drawing (e.g., it requires the artist to draw every line twice), it achieves a nice balance between the design trade-offs: the line drawings are composed in a free-hand manner familiar to artists, while the intended locations of every line on the 3D surface can be inferred with great accuracy.

3.4 Data Collection

This line drawing and registration procedure was repeated for 29 artists, most of whom were enrolled in one of four art classes (two composed of middle and high school students, one of adult evening students, and another of college students). Two of the participants were professional artists. Each artist completed up to 12 prompts.

Every participant completed a questionnaire listing his/her gender, age, and number of years of art training. In all, there were 22 females and 7 males. The ages ranged from 10 to 54 years, with an

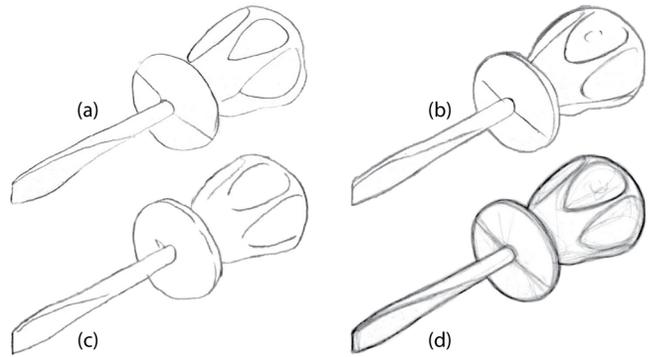


Figure 4: *Example drawings.* Three drawings of the screwdriver model from the same view (a,b,c), and the average of 14 drawings of the same view (d).

average of 22; and the participants reported an average of 6 years of art training (this number should be taken with a grain of salt, as some participants reported only training at the college level, while others reported all art classes).

Every artist was provided a folder with one page of instructions, twelve prompt pages, and twelve corresponding drawing pages (one for each model). The folders were arranged such that no artist could draw the same model more than once, and prompts for models, viewpoints, and lighting conditions were arranged in shuffled order to reduce effects of training on our analysis.

The artists were given brief verbal instructions (“draw lines that convey shape” and “be sure to copy every line from your free-hand drawing over the faded image below”) and then told to complete line drawings at their own pace for as long as they had time. Most of the art classes were scheduled for a two hour block, and each line drawing took 10-15 minutes, on average (with time split around 2/3 for drawing free-hand and 1/3 for tracing lines over the faded image). Each participant completed an average of 7.5 drawings – only one participant (a professional artist) completed all twelve available in his folder.

In all, 208 line drawing images were collected. Generally speaking, the artists followed the directions well, produced line drawings that convey shape effectively, and were careful when tracing lines over the faded image (some example line drawings are shown in Figure 4). However, in some cases, the artists clearly were not careful in the registration step, failing to follow even the exterior outline of the shape. Since accurate registration of lines to image features is essential for meaningful results, we cull these tracings from our analysis. To do this in an unbiased way, we assume that inclusion of the exterior outline is common to all human line drawings, and eliminate from our data set any drawings where less than 90% of the exterior is within 1mm of a human-drawn line. The remaining 170 line drawings form the basis for our analysis.

4 Results

We can investigate a number of questions by comparing how our captured line drawings overlap with the synthetic images provided as prompts to the artists.

We ask not only how artists’ drawings overlap with one another, but also how they overlap with lines generated by computer graphics algorithms, and how they can be predicted from local properties of the underlying surface and rendered image. The latter two topics are of particular interest for computer graphics, as they pro-

vide a characterization of artist line drawings in terms of line definitions (e.g., this drawing is $X\%$ occluding contours, $Y\%$ suggestive contours, $Z\%$ apparent ridges, and so on) and differential properties commonly used in the field (e.g., there is a high propensity for lines when the view dependent curvature is large and its derivative is zero). We believe that characterizing the relationships between artists’ drawings and these terms is the most interesting aspect of our study.

All comparisons between drawings are based on overlaps of pixels, rather than strokes or lines. This approximation is made for two practical reasons. First, since artists’ drawings are scanned after they are complete, we have no robust method to tell where line strokes begin and end (an artist may make several small strokes that merge together into a single line). Second, since it is difficult to establish correspondences between lines robustly, there is no obviously good measure of the overlap between sets of lines. Rather, we compare line drawings based on proximity of pixels, an approximation that is both simple and robust.

4.1 How similar are the artists’ drawings?

The first and most basic analysis we perform is to measure the similarity between artists’ drawings of the same prompts.

We can show consistency between artists qualitatively by superposing drawings on top of each other and visualizing how much they overlap (Figure 1). For example, Figure 5a shows each artist’s drawing in a separate color. In this example, the artists agree very closely with each other in most areas, especially along obvious features such as boundaries and occluding contours, but differ in exactly where they place lines in the right part of the rockerarm.

In order to quantify consistency, we compute a histogram of pairwise distances between artists’ drawings (Figure 5b). For every pixel in every drawing, we record the distance to the closest pixel in every other drawing of the same prompt, and then observe how often these distances lie within the tolerance of the tracing procedure (1mm). Across all prompts, approximately 75% of human drawing pixels are within 1mm of a drawn pixel in all other drawings for that prompt.

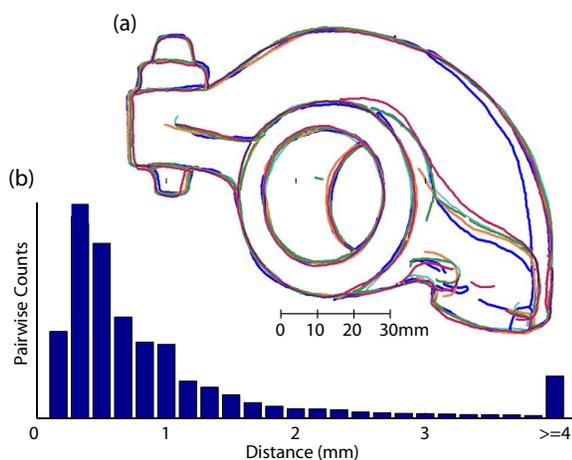


Figure 5: *Consistency of artists’ lines.* a) Five superimposed drawings by different artists, each in a different color, showing that artists’ lines tend to lie near each other. b) a histogram of pairwise closest distances between pixels for all 48 prompts. Note that approximately 75% of the distances are less than 1mm.

4.2 Do known CG lines describe artists’ lines?

A natural question to ask is how well currently known line drawing algorithms can describe the human artists’ lines. In our analysis, we consider the following line drawing algorithms: image intensity edges [Canny 1986], geometric ridges and valleys (as defined by [Ohtake et al. 2004]), suggestive contours [DeCarlo et al. 2003], and apparent ridges [Judd et al. 2007]. For the object space methods (ridges and valleys, suggestive contours, and apparent ridges), we always include the exterior boundary and interior occluding contours in the generated drawing. For Canny edge detection we always include the exterior, but not the interior contours, since they are not necessarily image intensity edges.

Quantifying comparisons between drawings

In order to compare an artist’s drawing and a computer generated drawing quantitatively, we use the standard information retrieval statistics of *precision* and *recall* (PR). Here, precision is defined as the fraction of pixels in the CG drawing that are near any pixel of the human drawing. Recall is defined as the fraction of pixels in the human drawing that are near any line of the CG drawing. We define “near” by choosing a distance threshold – we use 1mm.

As an example, consider comparing the set of five human drawings shown in Figure 5a with the lines generated by the apparent ridges algorithm (Figure 6). The output of the apparent ridges algorithm is not only a set of lines, but also a “strength” value at each line point. In general, we expect stronger lines to be more important and thus more likely to match the artists’ lines. We thus generate a series of binary apparent ridges images, each consisting of all points with strength above a given threshold. The PR of each drawing compared with this set of images is shown as a dotted pink line in Figure 6. As the strength threshold is lowered more lines are produced, typically causing recall to increase and precision to go down, yielding a sloping line in the PR graph. For completeness, we allow the PR plot to extend to $P = 1.0, R = 0.0$ (defined as a blank image), and directly downward to $P = 0.0$ from the highest recall obtained by the algorithm. Since each PR curve is defined for $P = [0, 1]$, we can compute an average curve by combining points along lines of fixed precision. The PR values for occluding contours alone are plotted as black dots, and are not averaged.

While computing precision and recall for the other object space definitions is performed similarly, computing PR for the Canny algorithm is slightly more complicated. Canny also has a natural “strength” value (the intensity of the filter response), but the algorithm has three free parameters: the size of the image filter, and the low and high thresholds (l and h). For our analysis, we fix the fil-

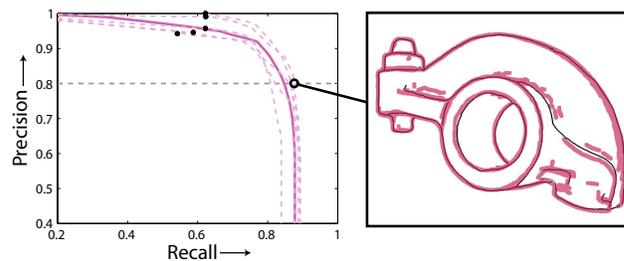


Figure 6: *Precision and recall example.* Left: apparent ridges are compared with five artist drawings. Solid line (highlighted) is the average PR for the set of drawings. Black dots indicate contours only. Right: an example drawing with overlapping apparent ridges (widened by 1mm on each side). PR of the example is circled.

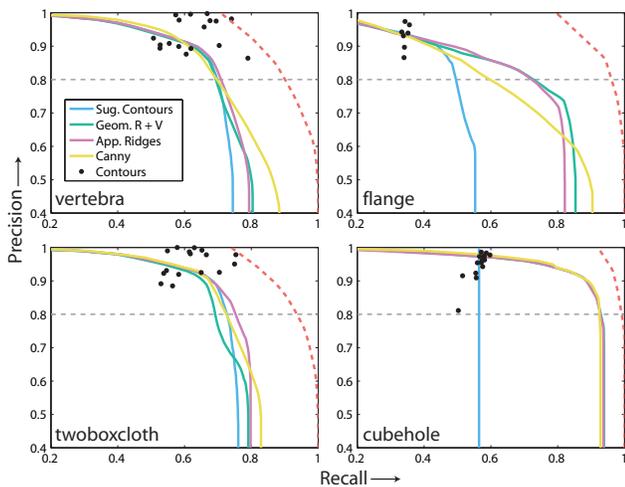
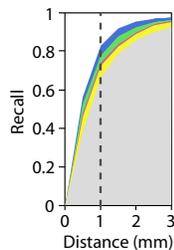


Figure 7: Average precision and recall. Lines further to the upper right represent better matches between the artists’ and CG lines. The black dots represent occluding contours only, and are not averaged. The slope of the curve generally falls off rapidly after 80% precision. Red dotted line indicates theoretical maximum recall. Note: axes begin at 20% recall and 40% precision.

ter σ at 2 pixels (a value we find produces reasonable results) and set $l = 0.4h$. We then vary h to control the number of line pixels produced by the edge detector.

In choosing the distance threshold, there is tension between achieving high recall and causing nearby, but distinct, line definitions to overlap. The figure inset right shows the cumulative recall over all drawings of lines explained by distinct (colored, as in Figure 7) versus overlapping (gray) definitions, as a function of threshold distance. We find that the threshold of 1mm provides a good balance between high recall and low overlap. Importantly, we find that while the exact recall numbers change somewhat with the distance threshold, the qualitative behavior and relative rankings of the CG lines do not.



Comparing CG lines individually

Figure 7 shows the average precision and recall for four representative models. The lines drawn on mechanical models such as the flange are classified readily into ridge-like features (green and pink), and are also explained well by Canny edges (yellow). The lines drawn on the cloth and bone models are largely occluding contours (black dots), though suggestive contours (blue) explain relatively more of the lines on these smooth models than on the other sets. The cubehole is almost completely explained by the CG methods. The tightness of the group of occluding contour dots gives a measure of how similar each drawing’s PR is to the others, since the unaveraged PR lines (with the exception of Canny) pass through the contour dots.

We find that overall, Canny edges, apparent ridges, and geometric ridges and valleys best match the human lines when taken individually. No single definition matches the artists’ drawings perfectly. Even in theory, however, no single CG algorithm with a single free parameter could match all the drawings, because the drawings are different from one another.

To gauge how far the CG algorithms could possibly improve, we imagine a computer algorithm to create an *optimal CG drawing* for a set of artists’ drawings. The optimal drawing has the highest recall for a given precision of any possible CG drawing (dotted red in Figure 7). Thus, it both describes how closely the original drawings match each other, and puts a conservative ceiling on how well *any* CG algorithm can match the human drawings.

The optimal CG drawing for precision P is created by the following procedure: for each of n binary, thinned drawing images, make an image that contains all the pixels within 1mm of any drawn pixel. Add these images to create an overlap image with values in the range $[0, n]$. Sort the pixels of this overlap image, and choose pixels with the highest value until precision falls below P . The blank image is defined to have $P = 1.0$, so this procedure always produces a drawing. Since the value of a pixel in the overlap image is exactly the precision of that pixel if added to the optimal drawing (times n), this procedure will choose a drawing with the largest possible number of pixels for a given precision.

Comparing CG lines in combination

Even with the theoretical ceiling imposed by the optimal CG drawing, the individual CG algorithms do not match the artists’ lines particularly closely. In almost all human drawings, however, there are examples of multiple classes of lines. For example, an artist might draw mostly along image intensity edges, but still draw other semantically important features. It is thus interesting to consider how different line definitions might explain the human drawings in combination.

In order to combine line definitions fairly, we use computer generated drawings with a fixed 80% precision. We then classify each pixel in each human drawing by the nearby CG lines. Pixels that lie near a single line definition are considered to be explained only by that definition, while pixels that lie near multiple definitions are considered explained by all the nearby definitions.

To visualize the results we create bar charts that partition the lines into object space definitions (blue), image intensity edges (green), or both (brown). Looking at the results in Figure 8a, we find that the large majority of lines are described by both image intensity edges (Canny edges) and an object space definition. Of the remainder, slightly more lines are explained by the combined object space approaches than by image edges alone.

Lines that are explained only by image edges account for at most 5% of all classified lines at 80% precision. We can therefore learn a good amount by examining the object space lines alone. Analyzing object space lines is also more informative than analyzing image edges, since the different definitions correspond to familiar geometric concepts. For example, we can break down the human

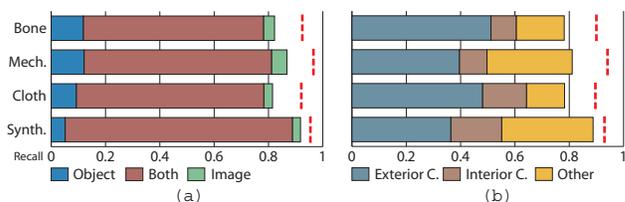


Figure 8: Categorizing artists’ lines. a) fraction of all lines explained by image based lines only, object based lines only, and both. b) fraction of all lines explained by the exterior contours, interior occluding contours, and all other object space lines. Dotted red indicates theoretical maximum recall.

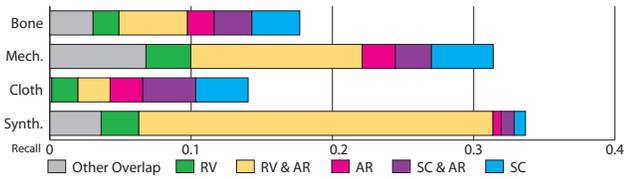


Figure 9: *Non-contour lines*. Categorization of artists’ lines that are not exterior or interior occluding contours: geometric ridges and valleys (RV), apparent ridges (AR), suggestive contours (SC), and combinations.

lines by intuitive categories, such as exterior and interior occluding contours, and everything else (Figure 8b). Across all model groups, the exterior boundary alone accounts for between 35-50% of all classified pixels. Occluding contours account for between 10-20% of all classified pixels, while all other definitions make up 20-35%.

If we look at the object space lines that are not exterior or interior contours, we see that in the mechanical and synthetic models, ridge and valley like features dominate the remaining lines (green, yellow, and pink in Figure 9). Both apparent ridges and geometric ridges and valleys contribute alone, but the majority of ridge like features are classified both as apparent and geometric.

For the bone and cloth models, ridges and valleys are less important, though the overall total of non-contour lines is approximately half that of the mechanical and synthetic models. One particularly interesting combination of object space lines is suggestive contours and apparent ridges (purple in Figure 9). Both suggestive contours and apparent ridges extend contours, but are largely disjoint elsewhere. Lines that are classified as both suggestive contours and apparent ridges, therefore, are likely to be extensions of occluding contours. As might be expected, the folds in the synthetic cloth lead to a disproportionately high amount of this combination.

4.3 Can CG lines characterize artists’ tendencies?

Given a way of describing an artist’s drawing in terms of CG line types, it is possible to investigate whether those descriptions can characterize the similarities and difference between artists’ styles or tendencies. For example, it may be possible to characterize whether certain artists tend to draw certain geometric features (e.g., ridges) more than other artists do. In such cases, the CG line definitions provide a vocabulary to discuss features of human line drawings.

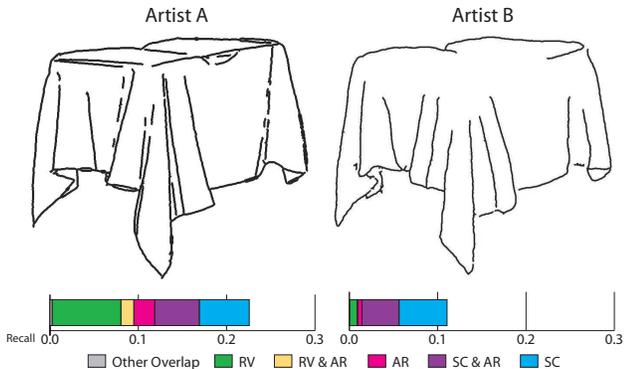


Figure 10: *Comparison of two drawings by different artists*. Two drawings of the same prompt show significant visual differences. These differences are reflected in the statistics, especially in the use of ridge-like lines (green).

Figure 10 shows a simple example of this type. Two drawings of the same prompt (twoboxcloth with Grace Cathedral lighting) are compared by the composition of CG line types. As in Figure 9, the colored bars indicate the fraction of the drawing made up by each line type. In this case, however, each set of bars represents a single drawing. One immediate difference between the drawings is that artist A drew more lines besides the contours. Non-contour lines account for 26% of artist A’s drawing, and only 13% of artist B’s drawing. The bulk of the difference between the artists is in the use of ridge-like lines (green, yellow, and pink bars). Artist A drew ridge-like lines along the top of the shape, while artist B did not. This visual difference is evident from the statistics, which show a large fraction of geometric ridges and apparent ridges in artist A’s drawing, and almost none in artist B’s drawing.

While this analysis is instructive in some cases, we find that some individual artists appear to have consistent tendencies that are not well explained by the CG lines examined here. For example, artist C made seven drawings, in which 16% of the lines are unexplained by the tested CG definitions at 80% precision. Over the same seven prompts, all other artists averaged only 8% unexplained lines. As shown by the examples in Figure 11, artist C made consistently distinct drawings. Artists with different styles, such as artist C, may provide valuable data for future research on line definitions and shape perception.

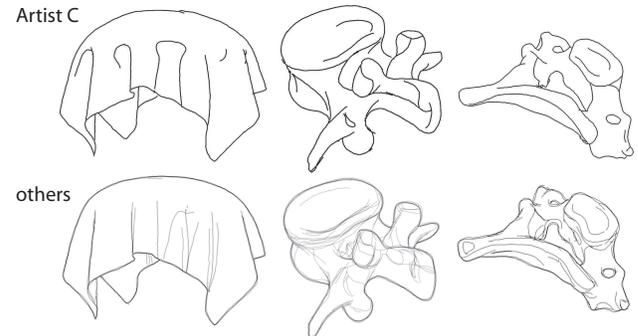


Figure 11: *Unusual drawings by an individual artist*. CG definitions explain fewer lines in artist C’s drawings than other artists’ drawings of the same prompts. However, artist C’s drawings are careful and consistent.

4.4 Can combined local properties explain lines?

While it is interesting to investigate the relationship between artists’ lines and the lines commonly used in computer graphics, a more fundamental question is how artists’ lines relate to differential properties of images and surfaces. The analysis above addresses this question indirectly, since each CG definition is based on a set of local properties, but it is restricted to the relationships suggested by the known line drawing algorithms.

To address this question, we take a classic data mining approach. For every pixel of every prompt, we compute: 1) a feature vector (x) of properties derived from the 3D surface and 2D rendered image, and 2) an estimated probability that a line will be included at the corresponding location in an artist’s line drawing. Our goals are to learn a function $f(x)$ that estimates the probability p of an artist drawing at a point (regression) and to understand which combinations of properties are most useful for building such a function (feature importance).

Choosing local properties

To build the feature vector for each pixel, we compute 15 local properties of three types commonly used in image processing, computer graphics, and differential geometry. First, we consider four image-space properties of the photorealistically-rendered image prompt: the luminance, the gradient magnitude after a Gaussian blur with $\sigma=2$ pixels (ImgGradMag), and the minimum and maximum eigenvalues of the image Hessian (corresponding to the minimum and maximum directional second derivative of luminance (ImgMinCurv and ImgMaxCurv, respectively)). In general, we expect that lines are more likely near image edges (ImgGradMag is large) and at ridges and valleys of luminance (where ImgMinCurv and ImgMaxCurv are large).

Second, we consider view-independent, differential properties of the visible point on the 3D surface, including the maximum (κ_1), minimum (κ_2), mean $((\kappa_1 + \kappa_2)/2)$, and Gaussian ($\kappa_1 \kappa_2$) curvatures (SurfMaxCurv, SurfMinCurv, SurfMeanCurv, and SurfGaussianCurv, respectively). In most cases, we expect lines to occur in areas where these expressions are large, though it has also been observed that lines are drawn near parabolic lines ($\kappa_1 \kappa_2 = 0$).

Third, we consider view-dependent properties that correspond to specific definitions for computer-generated lines. Corresponding to the definition of ridges and valleys, we take the derivative of the largest principal curvature in the corresponding principal direction (SurfMaxCurvDeriv), which is zero at ridges and valleys. Corresponding to occluding contours, we compute the dot product between normal and view vectors ($N \cdot V$). Corresponding to apparent ridges and valleys, we compute the largest view-dependent principal curvature (ViewDepCurv) and its derivative in the corresponding apparent principal direction (ViewDepCurvDeriv), which are large and zero, respectively, at apparent ridges. Corresponding to suggestive contours, we compute the radial curvature (RadialCurv) and its derivative in the radial direction (RadialCurvDeriv), which are zero and large, respectively, at suggestive contours. Finally, corresponding to principal highlights, we compute radial torsion, which is zero at principal highlights.

Finally, we estimate the probability, p , of an artist drawing at a pixel by averaging the registered drawings of all artists for the same prompt and blurring with a Gaussian filter to account for tracing errors ($\sigma = 0.5\text{mm}$).

Predicting lines by regression

While several of the computed properties clearly can be used to distinguish pixels where artists draw from where they do not (Figure 12), the interesting question is whether combinations of those properties can be used to predict where artists will draw more accurately than any of them alone. To investigate this question we have experimented with several regression models, including linear regression, radial basis functions, regression trees, and several others.

As an example, Figure 13a shows a regression tree built with the MSP package in Weka [Witten and Frank 2005] to predict the set of line drawings for one view of the twoboxcloth model shown in Figure 2. In this visualization, branches of the tree are shown as conditionals proceeding down the tree (indentation indicates level of the tree). If the conditional at a branch is TRUE, then its descendant on the next line is evaluated; otherwise, the one lower in the tree and connected by vertical bars is visited. Leaves are drawn with two text strings, the first one (colored) indicates a predicted value of p , while the second provides the count of pixels mapping to that leaf during training. The two images to the right of the tree show how the tree is used for regression – every pixel is sorted into

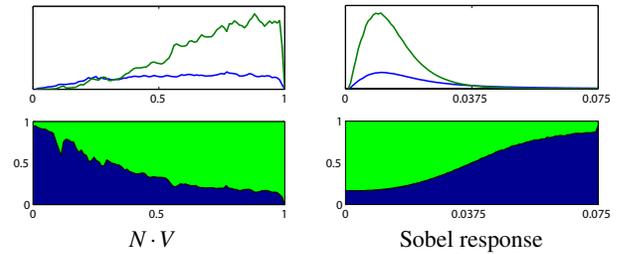


Figure 12: *Example local surface features.* Top: the relative frequencies of pixels near artists' lines (blue), and away from artists' lines (green). Bottom: the same data, but shown as probabilities.

the tree based on its properties and assigned the value stored at the leaf. Figure 13a shows the prediction of p resulting from this simple tree, while Figure 13b provides a visualization of which pixels sort into which leaves of the tree (pixels in the image are colored to match the text of the leaf).

In this example, several properties are combined by the decision tree to predict p , including ImgGradMag, RadialCurvDeriv, ViewDepCurvDeriv, SurfGaussianCurv, ViewDepCurv, $N \cdot V$, SurfMaxCurvDeriv, and SurfMinCurv. The set of properties chosen is instructive, as it suggests that they provide the highest incremental value in predicting p (at the start of tree building). Of course, many properties are correlated, and the decision tree may be non-optimal, so an alternative tree may have produced similar or better predictions. None the less, it is interesting to see how non-trivial combinations of local properties can be used to make predictions – even though the tree was purposely kept small in this example, it still is able to provide a plausible (albeit coarse) prediction for where artists draw lines (Figure 13a). If we consider deeper trees or other regression models, we are able to predict p from x more accurately.

It is also possible to use image processing tools to find ridges in the predicted probability distribution to produce a new line drawing. In the case shown in Figure 14a, we predict the image p (top left) using linear regression. We observe that this synthesis qualitatively resembles the composite of artists' drawings (Figure 14b). We then extract lines with a ridge finding algorithm to produce the line drawing in Figure 14d. This line drawing is comparable to a sample artist's drawing from the data set (Figure 14e) and con-

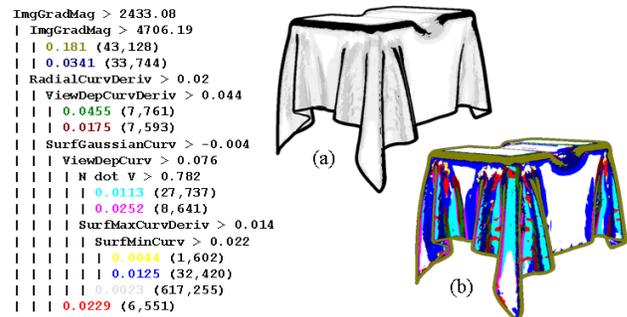


Figure 13: *Decision tree for predicting where artists will draw.* (a) decision tree learned from prompts of bones, (b) predicted probabilities of where artists will draw for this view (black is high probability), and (c) a visualization of which pixels fall into which leaves of the tree. Note that this tree was purposely kept small for didactic purposes, which causes the prediction to be coarse.

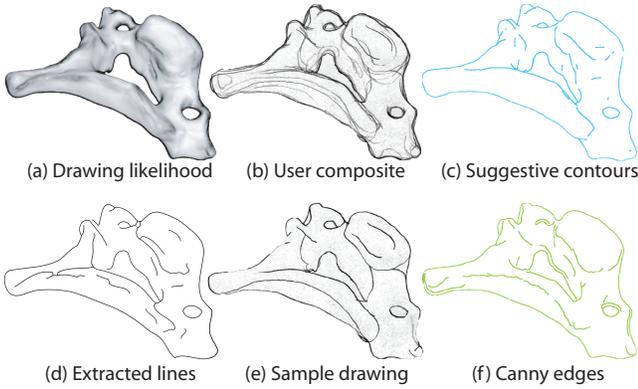


Figure 14: *Synthesis*. Linear regression on the cervical model, trained on drawings of other bone models, estimates the likelihood of an artist’s line appearing at each pixel (a). The likelihood image resembles the overlaid artists’ drawings of the same prompt (b). To synthesize a line drawing, ridges are extracted from the likelihood image (d). This synthesis qualitatively compares with a sample human drawing (e). Note that the synthesis could not have emerged from independently analyzing the CG lines shown in (c),(f)

tains elements from multiple CG line definitions. For comparison, the right two images (Figure 14c,f) show suggestive contours and Canny edges for the same prompt with parameter settings tuned to produce approximately the same line density as the artist’s drawing. Since the probability predicted by our model combines the differential properties that lay the foundations of these other algorithms, the fact that it exhibits features from more than one of them is not surprising. However, it provides a practical way to combine features from many automatic line drawing algorithms into a single framework where thresholds are learned automatically.

4.5 Which local properties are most important?

In our data mining framework, it is not only possible to predict where artists will draw, but also to examine which local features are most important when building such a regression model. For example, Random Forests [Breiman 2001] estimate the importance of every feature to its model by building a large number of decision trees trained on different subsets of the data [Breiman 2001]. For each feature m of each built tree, the error observed in predictions for the “out of bag” data (the part held out of training) is computed and compared to the error that is observed when values of feature m are permuted. The difference between these errors, averaged and normalized, is reported as the “importance” of feature m . Of course, the importance only measures an average over many trees, and so it does not capture the importance of any single feature at any single branch in the tree. Yet, it is interesting to use importance estimates to study how much low-level features contribute on average to predictions of line drawing locations. For this analysis, we make the assumption that almost all occluding contours ($N \cdot V = 0$) are drawn by artists (Figure 12), and so exclude any pixel within 1mm of a contour from the training set.

Table 1 shows the relative feature importance as computed with the Random Forest implementation of Breiman and Cutler in R [R 2005] for the remaining pixels of all drawings in our study. The first four columns report the importance of features (rows) estimated when training on models from each of one type (bones, cloth, mechanical, and synthetic), while the rightmost column reports the average over the whole dataset.

The results indicate that image-space intensity gradient magnitude is the feature amongst the tested set that is most useful in predicting the probability that an artist will draw at a particular location in our study (e.g., the average prediction error is largest if values of the image-space gradient magnitude are randomized). While image-space discontinuities often appear at the same place as boundary contours and occluding contours ($N \cdot V = 0$), the locations where those contours appear have been excluded from this study. So, this result suggests that image-space intensity gradients away from the contours are also highly correlated with artist line locations. Of course, this is not surprising, as ridges, valleys, and shadow boundaries are commonly drawn by artists. However, it is a bit surprising how all the simple image-space features (which do not require a 3D model to compute) are so important relative to the other more complex properties that have been the focus of recent research in computer graphics.

	Feature	Bone	Cloth	Mech	Synth	Avg
Image-Space	ImgGradMag	31.3	36.0	73.8	147.8	72.2
	ImgMaxCurv	38.0	15.8	55.5	64.4	43.4
	ImgMinCurv	15.1	15.3	23.4	56.6	27.6
	ImgLuminance	20.2	19.8	33.9	33.6	26.9
View-Dependent	$N \cdot V$	23.6	13.9	31.3	36.9	26.4
	ViewDepCurv	21.5	17.2	49.8	10.1	24.7
	ViewDepCurvDeriv	22.8	14.4	31.9	9.5	19.7
	RadialCurvDeriv	19.2	15.0	29.8	8.0	18.0
	RadialTorsion	14.6	10.3	27.8	7.2	15.0
View-Independent	RadialCurv	14.8	10.3	26.2	7.2	14.6
	SurfMaxCurvDeriv	16.9	11.0	27.3	8.9	16.0
	SurfMaxCurv	13.9	8.8	25.1	7.6	13.9
	SurfMinCurv	13.9	8.1	27.0	5.1	13.5
	SurfMeanCurv	14.1	8.9	22.5	7.0	13.1
	SurfGaussianCurv	13.1	8.5	25.7	4.9	13.1

Table 1: *Property “importance.”* This table shows the relative importance of local properties in predicting the probability of an artist drawing at a particular location. Columns show results for Random Forests trained on different subsets of the data.

4.6 Which CG lines are most important?

We can also use Random Forests to compute importance of the computer graphics line definitions studied in Section 4.2 for predicting where artists draw lines. For this analysis, we compute a new feature vector for every pixel storing the strength for every CG line definition. Note that strength is only defined at pixels where the algorithm would draw a line (e.g., zeros of maximum curvature derivative for ridges). At all other pixels, strength is always zero. We then recompute the Random Forests with the new feature vectors.

From the results in Table 2, we see that strong image-space gradients in illumination (Canny edges) still provide the strongest cues for artists to draw lines, even in relation to other computer graphics line definitions.

Feature	Bone	Cloth	Mech	Synth	Avg
Canny edges	18.2	37.2	50.9	145.0	53.4
Apparent Ridges	8.7	11.2	21.2	77.9	24.8
Ridges & valleys	6.8	7.4	24.4	77.1	24.5
Suggestive Contours	9.8	11.9	17.4	1.6	11.3

Table 2: *CG line definition “importance.”* This table shows results similar to those in Table 1, but for features derived directly from CG line definitions.

5 Conclusion

Overall, we make the following conclusions from this study, some of which are obvious and others of which are not.

First, we observe that artists in our study draw nearly 75% of their lines at a location that is within 1mm of *all* other artists drawing from the same prompt. The overlaps appear mainly at exterior and interior occluding contours, which comprise 57% of all lines drawn.

Amongst the other lines, large gradients in image intensity (as measured by image-space gradient magnitude) provide the best single predictor for where artists will draw under the conditions of our study. Lines generated by Canny edge detection on a prompt image cover 76% of artists' lines with 80% precision. These lines are almost entirely overlapped (95%) by lines predicted by object-space line definitions commonly found in computer graphics. The three object space definitions together cover 81% of the artists' lines at the same precision. We find that each of the four CG line definitions we considered explains some artists' lines that the others do not.

The cumulative output of the four line drawing algorithms considered in this paper cover only 86% of artists lines. We believe that some of the remaining lines could be explained by other line definitions based on local properties (e.g., lines from diffuse shading [Lee et al. 2007], or a new definition not yet described), or by clever combinations of current line definitions.

In some cases, however, it appears that artists select lines using criteria beyond the local features we examine here. For example, Figure 15 shows two cases where artists chose to draw lines on locally weak ridge and valley features, while omitting lines along locally stronger ridges and valleys. This choice is consistent between several artists.

While it may be that local surface features that we have not examined (or a combination of them) could explain the artists' choices in these cases, we believe it is more likely that the artists used non-local criteria for selecting these lines. Indeed, several artists mentioned that, for the flange drawing (Figure 15a), they omitted lines that were "implied." Since implied features depend on context, they are not describable with local properties alone.

Limitations and Future Work

We identify the following limitations of our study, which suggest topics for further work:

Potential bias. The possibility exists that our results carry some bias due to the way we collected data. We believe that artists generally followed the given instructions and faithfully copied their drawings from the drawing area to the tracing area, but it is possible that they altered their lines when tracing over a faint version of the prompt. Such alteration could contribute to the relatively high importance of image-space features noted in our analysis (Sections 4.5 and 4.6). Note, however, that the same image-space features in the faint images also appear in the prompt images.

Limited data. This paper draws a number of conclusions from the limited data set we have acquired to date. Of course, those conclusions are limited to the conditions of our study and the artists participating. In the future, we hope to expand this study to provide more drawings per prompt and cover a greater range of subjects. More data would reduce noise in the analysis, support a greater range of analyses, and offer greater predictive power in machine-learning approaches to synthesis.

This paper does not exhaust the possible analyses that could be performed with the data presented here. Other possibilities include:

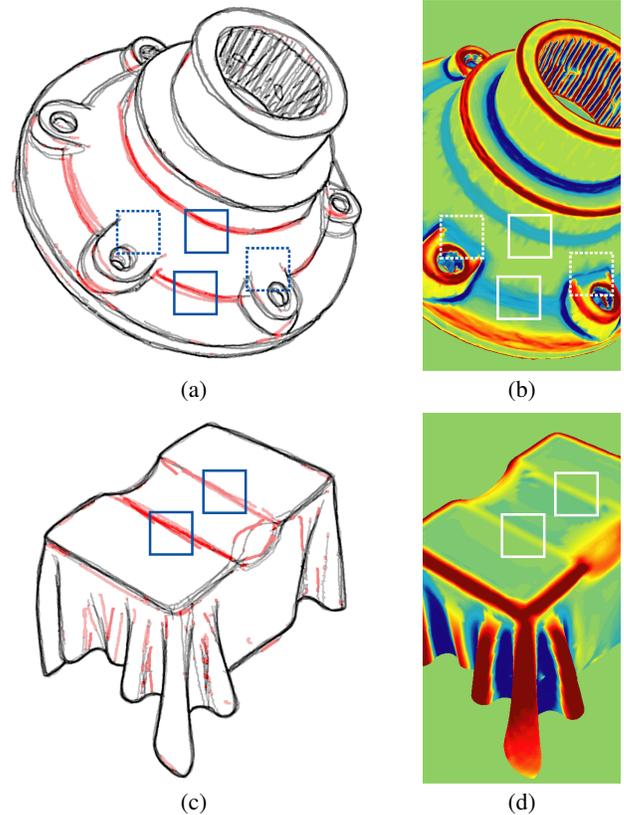


Figure 15: *Artists' sophisticated line selection.* The red lines (solid boxes) in this composite (a) are unexplained at 80% precision, but can be characterized as geometric valleys. However, several artists have omitted locally stronger valley features (dotted boxes), as shown by the maximum curvature of the model (b). The red lines in (c) are also ridge like features, but the curvature strength in the area is very low relative to the rest of the model (d).

Local per-pixel analysis. Our analysis has to date only studied per pixel properties of the strokes. We believe that studying such properties along the lengths of strokes will be both challenging and fruitful. Moreover, it is known that humans generally need a *global* view of a line drawing in order to fully understand it [Waltz 1975]. Likewise, Figure 15 suggests a global analysis might yield a better model for the human lines.

View-dependence of lines. The two viewpoints selected for our prompts are close enough to allow analysis of how lines move with changing views. One possible way to approach this question would be to reproject lines drawn from one view to compare with lines drawn in the other view.

In addition, our results suggest two areas for future studies:

Perception. As mentioned in Section 2, perception of drawings is an area of active research. This study would be greatly complemented by a companion study that showed what shape percepts are induced by the drawings presented herein.

Shading and Stylization. Thus far, we have considered only sparse, simple line drawings. Artists use a variety of other effects such as shading and varying line qualities to achieve greater visual impact. We believe that it would be fruitful to use the framework described in this paper to acquire and study such data.

Acknowledgments

First and foremost, we would like to thank the artists that participated in our study, including those from Art Collaborations and Princeton University. Many people provided helpful suggestions, feedback, and encouragement for this project, particularly Doug DeCarlo, Maneesh Agrawala, Fredo Durand, Eitan Grinspun, and the members of the Princeton Graphics Group. We also thank the NSF (CNS-0406415, IIS-0612231, CCF-0702672, CCF-0702672, IIS-0511965, and CCF-0347427) and Google for funding to support this project and the Aim@Shape, VAKHUN, and Cyberware repositories for the models used in the study.

References

- BREIMAN, L. 2001. Random forests. *Machine Learning* 45, 1, 5–32.
- CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6, 679–698.
- COLE, F., DECARLO, D., FINKELSTEIN, A., KIN, K., MORLEY, K., AND SANTELLA, A. 2006. Directing gaze in 3D models with stylized focus. *EGRS 2006* (June), 377–387.
- DEBEVEC, P. 1998. Rendering synthetic objects into real scenes. In *Proceedings of SIGGRAPH 1998*, 189–198.
- DECARLO, D., AND RUSINKIEWICZ, S. 2007. Highlight lines for conveying shape. In *Proceedings of NPAR 2007*.
- DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. *ACM Trans. Graph.* 21, 3 (July), 769–776.
- DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph.* 22, 3, 848–855.
- GIRSHICK, A., INTERRANTE, V., HAKER, S., AND LEMOINE, T. 2000. Line direction matters: an argument for the use of principal directions in 3d line drawings. In *Proceedings of NPAR 2000*, 43–52.
- GOOCH, B., REINHARD, E., AND GOOCH, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph.* 23, 1, 27–44.
- GUPTILL, A. L. 1976. *Rendering in Pen and Ink*. Watson-Guption Publications, New York.
- HAMEL, J., AND STROTHOTTE, T. 1999. Capturing and re-using rendition styles for non-photorealistic rendering. In *Computer Graphics Forum (Eurographics '99)*, vol. 18(3), 173–182.
- HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. In *Proceedings of SIGGRAPH 2000*, 517–526.
- HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of SIGGRAPH 2001*, 327–340.
- HERTZMANN, A., OLIVER, N., CURLESS, B., AND SEITZ, S. 2002. Curve analogies. In *EGRW 2002*, 233–245.
- HILBERT, D., AND COHN-VOSSEN, S. 1999. *Geometry and the Imagination*. American Mathematical Society.
- ISENBERG, T., NEUMANN, P., CARPENDALE, S., SOUSA, M. C., AND JORGE, J. A. 2006. Non-photorealistic rendering in context: an observational study. In *Proceedings of NPAR 2006*, 115–126.
- JUDD, T., DURAND, F., AND ADELSON, E. H. 2007. Apparent ridges for line drawing. *ACM Trans. Graph.* 26, 3, 19.
- KALNINS, R. D., MARKOSIAN, L., MEIER, B. J., KOWALSKI, M. A., LEE, J. C., DAVIDSON, P. L., WEBB, M., HUGHES, J. F., AND FINKELSTEIN, A. 2002. WYSIWYG NPR: drawing strokes directly on 3D models. In *Proceedings of SIGGRAPH 2002*, 755–762.
- KOENDERINK, J. J., VAN DOORN, A., CHRISTOU, C., AND LAPPIN, J. 1996. Shape constancy in pictorial relief. *Perception* 25, 155–164.
- LEE, Y., MARKOSIAN, L., LEE, S., AND HUGHES, J. F. 2007. Line drawings via abstracted shading. *ACM Trans. Graph.* 26, 3, 18.
- LUM, E. B., AND MA, K.-L. 2005. Expressive Line Selection by Example. *The Visual Computer Special Issues of Pacific Graphics 2005 21*, 8–10 (Sept.), 811–820.
- MEYER, S. E., AND AVILLEZ, M. 1985. *How to Draw in Pen and Ink*. Roundtable Press.
- OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.* 23, 3.
- PECK, S. R. 1982. *Atlas of Human Anatomy for the Artist*. Oxford University Press.
- PHILLIPS, F., CASELLA, M. W., AND GAUDINO, B. M. 2005. What can drawing tell us about our mental representation of shape? *Journal of Vision* 5, 8 (9), 522–522.
- R. 2005. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.
- RUSKIN, J. 1895. *The Elements of Drawing*. Elibron Classics.
- SAITO, T., AND TAKAHASHI, T. 1990. Comprehensible rendering of 3-d shapes. In *Proceedings of SIGGRAPH 1990*, 197–206.
- SANTELLA, A., AND DECARLO, D. 2004. Visual interest and NPR: an evaluation and manifesto. In *Proceedings of NPAR 2004*, 71–78.
- SMITH, S. 1997. Looking at line. In *Complete Guide to Drawing and Painting*, R. D. Editors, Ed. Readers Digest, 13.
- WALTZ, D. 1975. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, P. H. Winston, Ed. McGraw-Hill, New York.
- WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proceedings of SIGGRAPH 1994*, 91–100.
- WINKENBACH, G., AND SALESIN, D. H. 1996. Rendering parametric surfaces in pen and ink. In *Proceedings of SIGGRAPH 1996*, 469–476.
- WINNEMÖLLER, H., FENG, D., GOOCH, B., AND SUZUKI, S. 2007. Using npr to evaluate perceptual shape cues in dynamic environments. In *Proceedings of NPAR 2007*, 85–92.
- WITTEN, I. H., AND FRANK, E. 2005. *Data mining: Practical machine learning tools and techniques*, 2nd edition.
- YAFRAY. 2008. *YafRay 0.0.9: Yet another free raytracer*. www.yafRay.org.