More Optimal Strokes for NPR Sketching

J.P. Lewis*
Graphics Primitives

Nickson Fong
Egg Story Creative Productions

Xie XueXiang
Nanyang Technological University

Seah Hock Soon
Nanyang Technological University

Tian Feng
Nanyang Technological University

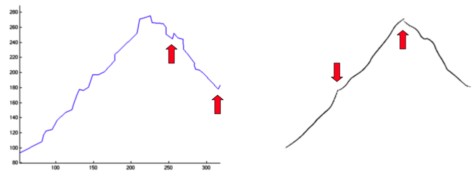

Figure 1: Test data set (left), rendered after clustering into three strokes (right). The contour is split at somewhat salient points rather than at curvature maxima, or into roughly equally sized strokes in the absence of distinguished features (left slope).

Abstract

Sketching is a drawing style where approximations and successive refinement in the drawing process are evident. The approximation of contours in sketching involves multiple overlapping strokes that are relatively long in regions of low curvature and shorter in high-curvature areas, yet unimportant high-curvature details are omitted in the initial stages of a sketch. Rendering contours with a single long stroke does not capture the feel of a sketch, and a simple strategy of breaking strokes at curvature maxima is easily confused by unimportant details and noise. We address the contour breaking problem for sketching by clustering samples of the contour based on proximity and orientation, making use of a global clustering algorithm (normalized cuts). The strokes generated by this approach qualitatively resemble those produced by real artists, and the successive approximation effect seen in sketching can be simulated by employing our approach at a succession of scales (increasing the number of clusters).

1 Introduction

"Sketching" describes a drawing that is not a final, perfect result. In a sketch the drawing process is somewhat evident, with approximation of contours and successive approximation in stroke placement often being visible. Although this approximation might be considered in abstract to be a sort of "rendering error", in fact the stroke character in a sketch can be quite beautiful, and may be a greater component of the art than the actual subject depiction.

In this short paper we seek to partially emulate the character of sketching strokes. Typically these strokes are made lightly and at relatively high speed, and a single contour is successively approximated with multiple strokes. Because rapid hand movements cannot curve sharply, strokes are often broken at points of curvature. Fig. 2 shows examples of characteristic sketching stroke effects.

These effects can be approximated with simple and local techniques, such as breaking the contour at curvature maxima, and perturbing strokes with noise. Although the resulting drawing is likely to be adequate for many purposes, it is unlikely to be an ideal example of a sketch: simply breaking the contour at curvature maxima is suspect, since not all maxima are equally important (Fig. 1). This is doubly true in common cases where the original contour has some artifactual curvature resulting from the discrete raster (e.g. vectorization algorithms that locally are limited to stepping in horizontal, vertical, and diagonal directions, as in Fig. 5), or from tracing the silhouette of a polygonal rather than spline model. While the algorithmic artifacts alone can be addressed by fitting an approximating spline to the contour, the spline may also smooth points of high curvature, obscuring the question of where to split the contour.

Our contribution is to note that the stroke segmentation problem can be considered as exactly that – an instance of a segmentation or clustering problem. The proximity and orientation of silhouette curve samples can be compared, with sufficiently similar samples being naturally grouped into strokes. This approach opens the problem to a variety of clustering and segmentation algorithms that have been developed in recent years. We employ the normalized cuts algorithm (section three), although this is not necessarily the only or best choice.

One may question whether a global approach striving for optimality (normalized cuts have an indirect connection to optimality) is really needed. Skilled artists are certainly not using only local information in drawing strokes: they can observe and remember the whole scene, and extensive prior practice provides knowledge of how the whole should be decomposed into pieces. In fact many practiced forms of human movement appear to optimize physical quantities (see [Bobrow et al. 2001] and references therein) and require training to do so (e.g. learning to walk, or to ice skate).

2 Related Work

The literature on NPR drawing algorithms is large and includes issues of stroke placement and orientation (e.g., [Salisbury et al. 1997; Hertzmann 1998]), silhouette tracing [Northrup and Markosian 2000; Isenberg et al. 2002], and simulations of particular media ([Winkenbach and Salesin 1994] and especially the detailed simulation in [Sousa and Buchannan 2000]). [Hertzmann 2003] provides a survey focusing on stroke placement issues (also see [Gooch and Gooch 2001; Strothotte and Schlechtweg 2002]). Examples of recent developments include [Sousa and Prusinkiewicz 2003], which selects a small number of salient contours that are then carefully rendered in an ink style, and [Li and Huang 2003], which uses image statistics and regional texture to tune pencil-like shading (hatching) to suit different regions.

Most silhouette rendering efforts have sought to link segments into as long a contour as possible. The full contour can then be rendered

*zilla@computer.org

Figure 2: Examples of real sketches, and details (right).

in the style of a careful ink drawing or painting, but long contours are inappropriate for sketching (on the other hand linked contours are a convenient input to our process). [Mignotte 2003] describes a stochastic optimization process for sketching in which a truncated Fourier basis provides a prior of admissible deformations on stroke shape, and edge gradients provide the data. Less probable strokes are rendered with less pressure and width. While [Mignotte 2003] chooses only one stroke per non-overlapping window on the image, a minor modification of their process to repeatedly sample strokes at a single location could generate sketching effects similar to ours. On the other hand, their process is somewhat expensive, requiring on the order of tens of minutes even without repeated sampling.

Silhouette tracing of 3D models is peripheral to our main concern; our stroke sketching can work with both 3D silhouettes and contours obtained from other (perhaps 2D) sources. Silhouette algorithms differ in whether they produce an explicit silhouette curve representation or produce the sihouette by independently coloring appropriate pixels (vector or raster representations), and in whether they make direct use of the 3D model or begin with a rendered image (object or image-based algorithms). Silhouette algorithms are surveyed in [Isenberg et al. 2003].

Image segmentation is often approached by clustering based on location, pixel color, and other attributes. In recent years several new families of clustering algorithms have been introduced: normalized cuts [Shi and Malik 2000] and related approaches [Ng et al. 2001], min-cut/max-flow based graph cuts [Kolmogorov and Zabih 2002], mean shift [Cheng 1995], and others [Gdalyahu et al. 2001]. In many cases these approaches offer clearly improved performance relative to the classic k-means (Lloyd's) algorithm and Expectation Maximization based clustering, but there is not yet a widespread understanding of how the methods compare to each other.

3 Method

Our general approach to producing an NPR sketch consists of the following steps: 1) silhouette tracing, 2) segmentation, 3) spline approximation, and 4) rendering. The contribution of this paper is in step two (segmentation), so the remaining areas will be described only briefly.

Silhouette Tracing. Our silhouette estimation is based on [Raskar and Cohen 1999], with an additional post process that links adjacent silhouette samples into chains. In general linking can be done before or after clustering for segmentation. By doing it before clustering, the approximate orientation of samples (used in the clustering) is available as the direction from each sample to one of its neighbors.

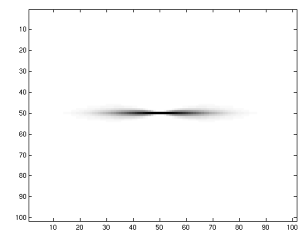

Figure 3: Shape of the pairwise affinity, decaying from a horizontally oriented sillhouette segment at center (black indicates stronger). The affinity falls off with distance, but also allows more orientation discrepancy with increasing distance.

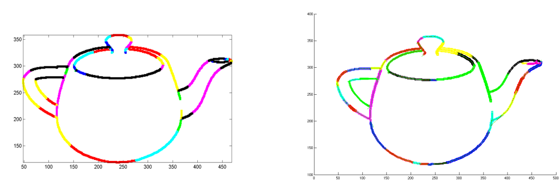

Figure 4: (Figure best viewed in color): (left) Segmentation resulting from affinity defined as a weighted sum of distance and orientation discrepancies. Parallel but separate contours on the handle and spout are grouped together. (right) Segmentation resulting from affinity as shown in Fig. 3. (A few distinct clusters are colored similarly due to reuse of the color labels).

Segmentation. Normalized cuts and other graph cut segmentations require a pairwise affinity $A_{i,j}$ giving the strength of the estimated similarity between any two contour samples i, j. The choice of affinity is important – using a simple weighted sum of proximity and orientation results in a segmentation that incorrectly groups samples on similarly oriented but separate silhouettes (Fig. 4). We use an affinity inspired by the voting field in tensor voting [Medioni et al. 2002] and perceptual grouping methods. This characteristic "figure-8" shape (Fig. 3) generally falls off with distance and orientation discrepancy, but it also allows relatively more orientation discrepancy at increasing distance, to allow for smooth curvature of the contour. This function is computed by projecting the location of sample j onto the line defined by the orientation of sample i, and then forming the following measure involving the distance along the orientation line d_o for sample i, the projection distance d_p from sample j to that line, and the dot product of the two orientations $o_i \cdot o_j$:

$$o_i \cdot o_j \ \exp\left(-\frac{(d_o + \alpha \, d_p / d_o)^2}{\sigma^2} \right)$$

where α, σ are constants that tune the orientation selectivity and

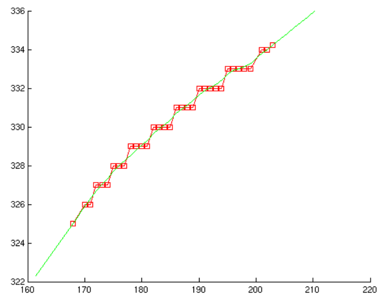

Figure 5: Approximating spline fit with intentional overshoot.

overall scale. The d_p/d_o term causes the orientation selectivity to reduce with increasing distance.

Given the affinity matrix A, segmentation proceeds by considering a graph with $A_{i,j}$ being the weight between nodes i and j. The overall goal is to cut the graph such that the weights of the cut edges are small while the interior weights of the resulting subgraphs are simultaneously large. This general formulation describes a variety of clustering algorithms. The intuition behind spectral methods is as follows [Forsyth and Ponce 2003]: A good cluster will have strong weights in the affinity matrix, and elements of the cluster will be strongly associated with the cluster, so the objective $w^T Aw$ (with w a vector of weights giving the association of each element with a proposed cluster) will be large for a good cluster. Maximizing this subject to $\|w\|$ remaining constant gives

$$w^T Aw + \lambda(w^T w - 1)$$

with the result that w is the leading eigenvector of A. Values of w larger than a threshold indicate membership in the strongest cluster. This procedure can be iterated on the remaining (unchosen) samples to find additional clusters.

It is known that the basic spectral clustering scheme just described has a tendency to produce overly small clusters, since the sum of weights to a small group of nodes will also tend to be small. The normalized cuts algorithm [Shi and Malik 2000] addresses this by normalizing the cost of the cut weights by the total weight mass from that cluster, thereby taking cluster size into account.[1] This discrete problem is then approximated by allowing continuous values, leading to an eigenvector system only slightly more complicated than the one described above. Rather than recursively compute a series of segmentations we use the n-way variant of normalized cuts [Yu and Shi 2003].

Approximating Spline. Silhouette samples often contain artifacts of polygonal models or of raster grid tracing (Fig. 5). We therefore fit strokes with an approximating spline before rendering. The spline y is formulated as

$$\min_y \; \|S(y - x)\|^2 \; + \; \lambda \, y^T C^T C y$$

where S is a "selection matrix" selecting only the elements of y corresponding to the locations of the data x, and C is a matrix containing the finite-difference approximation to the second derivative. This can be solved for y as a linear system. This formulation also allows a natural formulation of the characteristic stroke "overshoot"

[1] As a result, when there is no evidence guiding it towards other choices, normalized cuts will cluster a straight line into roughly equally sized strokes (Fig. 1).

Figure 6: Test pattern (top) and a similar shape produced by an artist (bottom), with enlargements (right). Please enlarge to see further line details.

often seen in sketching, by having the fit spline extend beyond the data. In the past the overshoot effect has been obtained simply by extending the line using the slope of the last segment. This sudden transition to an extended straight segment may look unnatural, however, if the the rest of the stroke is significantly curving. In our formulation the curvature in the overshoot region decays more smoothly (Fig. 5).

Continuing the curvature into the overshoot region can be accommodated easily, by using $\|Cy - c\|^2$ in place of $\|Cy\|^2$, with c being a desired curvature vector (extrapolated from an average of Cx). We believe however that the overshoots in real sketches tend to approach straight lines and so curvature continuation was not employed here.

Stroke Rendering. Although we have experimented with more textural stroke rendering styles, the figures in the paper all use simple antialiased lines from the Java language graphics library with a constant but adjustable opacity. Design parameters consist of the opacity, the maximum overshoot (overshoot on each stroke is chosen at random up to the maximum), and the usual NPR small random perterbation of stroke vertices (as in [Northrup and Markosian 2000] for example), added to increase the hand-drawn look.

4 Results and Discussion

Figs. 6-7 show examples of our stroking approach. Note that the stroking effect is rather subtle but can be seen clearly when the figures are enlarged.

In sketching artists often lightly outline major structures first and then overlay additional structures and details. This successive refinement strategy can be quite effectively simulated with our approach, by varying the number of clusters and overlaying the results. The result figures make use of this technique, for example, Fig. 7 (left) overlays the segmentations obtained with 60, 70, 80, and 90 clusters.

The method we have outlined has several potential limitations. Performance is a possible concern. In our implementation the eigenvector solve takes from a few seconds to a few tens of seconds for the drawings shown here (several thousand silhouette samples). There is opportunity for acceleration, however, by using the Nystrom approximation, or by simply subsampling the silhouette. A

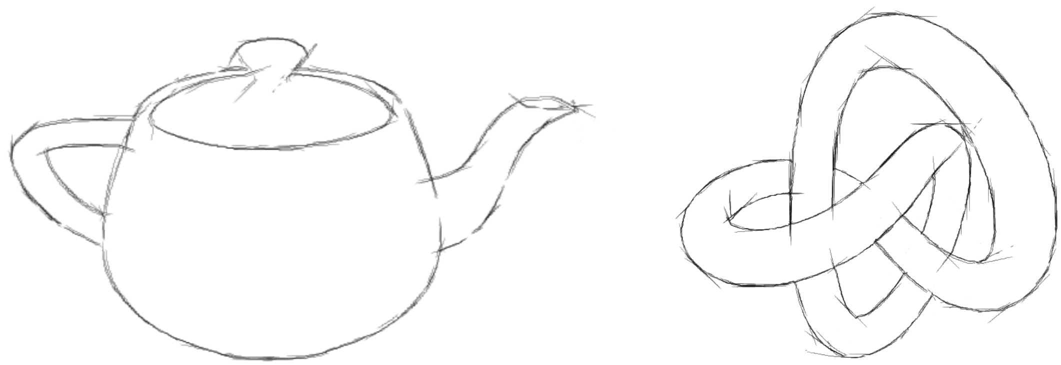

Figure 7: Teapot and knot each rendered with four different segmentation levels, and increased overshoot in the knot figure. Enlarge to see line quality. Several slight direction changes in the overshoots at T-junctions are visible in the knot figure; these are due to a few samples in the vicinity of the junction being incorrectly clustered. This could be avoided by tuning the clustering affinity, or by processing each sillhouette edge individually (if the silhouette tracing provides this information).

more serious concern is that the number of clusters and affinity distance require tuning. In our experience these parameters have some range of values over which the results are reasonable and vary intuitively with the parameters, however, outside this range the results (with few clusters in particular) sometimes surprised us. Some of the variety of recently proposed spectral and graph-based clustering methods (for example [Ng et al. 2001]) may address this problem in the future.

References

BOBROW, J. E., MARTIN, B., SOHL, G., WANG, E. C., KIM, J., AND PARK, F. C. 2001. Optimal robot motions for physical criteria. *J. Robotic Systems 18*, 12 (Dec.), 785–795.

CHENG, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8, 790–799.

FORSYTH, D., AND PONCE, J. 2003. *Computer Vision: A Modern Approach*. Prentice Hall.

GDALYAHU, Y., WEINSHALL, D., AND WERMAN, M. 2001. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 10, 1053–1074.

GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A. K. Peters, Ltd., Natick, MA, USA.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. 453–460.

HERTZMANN, A. 2003. A survey of stroke based rendering. *IEEE Computer Graphics and Applications 23*, 4, 70–81.

ISENBERG, T., HALPER, N., AND STROTHOTTE, T. 2002. Stylizing silhouettes at interactive rates: From silhouette edges to silhouette strokes. *Computer Graphics Forum 21*, 3 (Sept.).

ISENBERG, T., FREUDENBERG, B., HALPER, N., SCHLECHTWEG, S., AND STROTHOTTE, T. 2003. A developer's guide to silhouette algorithms for polygonal models. *IEEE Comput. Graph. Appl. 23*, 4, 28–37.

KOLMOGOROV, V., AND ZABIH, R. 2002. What energy functions can be minimized via graph cuts? In *ECCV (3)*, 65–81.

LI, N., AND HUANG, Z. 2003. A feature based pencil drawing method. In *Proc. GRAPHITE 03*, 135–140.

MEDIONI, G., TANG, C. K., AND LEE, M. S. 2002. Tensor voting: Theory and applications. In *European Conference on Computer Vision*.

MIGNOTTE, M. 2003. Unsupervised statistical sketching for non-photorealistic rendering models. In *10th IEEE International Conference on Image Processing, ICIP'03, volume III*, 573–577.

NG, A., JORDAN, M., AND WEISS, Y. 2001. On spectral clustering: Analysis and an algorithm. In *In Advances in Neural Information Processing Systems 14*.

NORTHRUP, J. D., AND MARKOSIAN, L. 2000. Artistic silhouettes: a hybrid approach. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, ACM Press, New York, NY, USA, 31–37.

RASKAR, R., AND COHEN, M. 1999. Image precision silhouette edges. In *SI3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 135–140.

SALISBURY, M. P., WONG, M. T., HUGHES, J. F., AND SALESIN, D. H. 1997. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 401–406.

SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8, 888–905.

SOUSA, M. C., AND BUCHANNAN, J. W. 2000. Observational models of graphite pencil materials. *Comput. Graph. Forum 19*, 1, 27–49.

SOUSA, M. C., AND PRUSINKIEWICZ, P. 2003. A few good lines: Suggestive drawing of 3d models. *Computer Graphics Forum 22*, 3 (September).

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-Photorealistic Computer Graphics*. Morgan Kaufmann, San Francisco, CA, USA.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, 91–100.

YU, S. X., AND SHI, J. 2003. Multiclass spectral clustering. In *Int. Conf. Computer Vision (ICCV)*.