

Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution

Victoria Interrante
Institute for Computer Applications in Science and Engineering (ICASE)

Abstract

The three-dimensional shape and relative depth of a smoothly curving layered transparent surface may be communicated particularly effectively when the surface is artistically enhanced with sparsely distributed opaque detail.

This paper describes how the set of principal directions and principal curvatures specified by local geometric operators can be understood to define a natural “flow” over the surface of an object, and can be used to guide the placement of the lines of a stroke texture that seeks to represent 3D shape information in a perceptually intuitive way.

The driving application for this work is the visualization of layered isovalue surfaces in volume data, where the particular identity of an individual surface is not generally known *a priori* and observers will typically wish to view a variety of different level surfaces from the same distribution, superimposed over underlying opaque structures.

By advecting an evenly distributed set of tiny opaque particles, and the empty space between them, via 3D line integral convolution through the vector field defined by the principal directions and principal curvatures of the level surfaces passing through each gridpoint of a 3D volume, it is possible to generate a single scan-converted solid stroke texture that may intuitively represent the essential shape information of any level surface in the volume.

To generate longer strokes over more highly curved areas, where the directional information is both most stable and most relevant, and to simultaneously downplay the visual impact of directional information in the flatter regions, one may dynamically redefine the length of the filter kernel according to the magnitude of the maximum principal curvature of the level surface at the point around which it is applied.

Strokes are constrained in narrowness by the resolution of the volume within which the texture is represented, but may be variably widened, at the time of rendering, to reflect shading information or any other function defined over the volume data, by adaptively indexing into multiple pre-computed texture volumes obtained from advected particles of different sizes.

This research was supported by the National Aeronautics and Space Administration under NASA contract number NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681.

1 Introduction

The texturing method described in this paper is intended as a partial solution to the problem of effectively visualizing the complex spatial relationships between two or more overlapping surfaces. Applications requiring the simultaneous appreciation of multiple layers of information arise in a number of fields in scientific visualization, and particularly in situations where surfaces of interest are defined by a level set of intensities in a volume distribution. The specific application that motivated this research is radiation therapy treatment planning, in which physicians need to evaluate the extent to which a particular three-dimensional distribution of radiation dose might satisfy the twin objectives of maximizing the probability of tumor control and minimizing the probability of complications due to the excess irradiation of normal tissues.

Although transparent surface rendering offers the best possibility for enabling an integrated appreciation of the 3D spatial relationship between two superimposed structures, it can often be difficult, under ordinary conditions, to adequately perceive the full three-dimensional shape of a layered transparent surface or to accurately judge its depth distance from an underlying opaque object. To compensate for the lack of naturally-occurring shape and depth cues, one may artificially enhance the transparent surface with a small, stable set of appropriately defined, sparsely distributed, opaque surface markings.

It is universally recognized that shape and depth judgements can improve markedly when surfaces are covered with an appropriately-defined texture rather than left plain or inappropriately textured, and that shape and depth may be understood more accurately and more readily from some texture patterns rather than others [5, 6, 31, 32, 35, 37]. These results have been shown for actual objects viewed directly [9], as well as for photographs of actual objects [10] and for computer-generated images of objects viewed either monocularly [3] or in stereo [5], and have been deftly exploited by op artists such as Victor Vasarely [40].

What are the characteristic of texture that are most important for showing shape, and how can we define a texture pattern that conveys shape information both accurately and intuitively? Although research toward a definitive explanation of the role of texture in shape perception remains ongoing, some key observations help motivate the underlying philosophy behind the work described in the remainder of this paper.

Gradients of element compression, or the relative orientations of naturally elongated elements, appear to play a central role in the perception of surface curvature [6, 35, 37], and the perception of shape from texture may be inhibited when the texture pattern is non-homotropic [33] or when texture anisotropies mimic the effects of foreshortening [44]. Although we can estimate local surface orientation remarkably well from the projective deformation of a circle to an ellipse [30], recent research suggests that we do not understand shape as collection of mutually independent local estimates of the surface normal directions at scattered points but rather as an organization of space based on local depth order relationships [17, 36].

Densely spaced planar contours have historically been a popular device for representing shape with line; however recent work [14] inspired by empirical observations of the use of line by pen-and-ink

illustrators suggests that lines following the direction of greatest surface curvature may convey shape in a particularly effective and intuitive manner.

This paper advances the state of the art in surface shape representation by proposing that the set of principal directions and principal curvatures [16] can be understood to define the intrinsic geometrical and perceptual “flow” of the surface of an object, and can be used as such to automatically define a continuous stroke texture that “follows the shape” in a perceptually intuitive and geometrically meaningful way. Specifically, this paper describes a technique for automatically generating a single solid texture [24, 25] of scan-converted strokes that can be simply and efficiently applied during rendering to more effectively convey the essential shape features of every level surface in the volume distribution.

2 Previous And Related Work

Dooley and Cohen [7] suggested using screen-space opacity-masking texture patterns to help disambiguate the depth order of overlapping transparent surfaces; such patterns, however, may give a false impression of flatness when applied to curved surfaces. To more clearly represent the shapes of transparent surfaces in volume data, Levoy et al. [20] proposed using a solid grid texture, comprised of planes of voxels evenly spaced along the two orthogonal axes of the volume most nearly aligned with the image plane, to increase the opacity of selected planar cross-sections. Interrante et al. [13] suggested selectively opacifying valley and sharp ridge regions on transparent skin surfaces to emphasize their distinctive shape features in the style of a viewpoint-independent “3D sketch”. Rheingans [26] described how surface retriangulation could be used in combination with a procedurally-defined 2D opacity-masking texture of small circles to accurately portray fine-grained information about the orientation of a smoothly curving layered transparent surface, and Interrante et al. [14] proposed a method for covering a transparent surface with individually-defined short opaque strokes locally aligned with the direction of maximum surface curvature. Although the results presented in [14] are encouraging, the stroke definition proposed there is cumbersome, the lines do not bend to follow the principal directions along the length of their extent, and the texture definition is inherently tied to a specific surface definition, and would have to be completely reiterated in order to be applied to multiple level surfaces from the same 3D distribution.

In terms of more general inspiration, Saito and Takahashi [27] showed how the comprehensibility of 3D shaded surface renderings could be improved via highlighting the first- and second-order depth discontinuities, and suggested defining a hatching pattern based on either the latitude and longitude lines of a sphere or the parametric representation of a surface that could be applied according to the values in an illumination map to evoke the impression of a pen-and-ink illustration. Winkenbach and Salesin defined intricately detailed resolution-independent fine stroke textures [42] and showed how they could be applied in accordance with the directions of the surface parameterization to represent a class of curved surfaces in the style of a pen-and-ink illustration [43]. Other textures that “follow the surface” in some sense include the reaction-diffusion textures proposed by Turk [38] and Witkin and Kass [45]. Most

recently, Turk and Banks [39] described a method for evenly distributing streamlines over a 2D vector field to represent the flow in a visually pleasing manner, akin to a hand-drawn illustration.

Slightly farther afield, researchers in computer-aided design [2, 12, 23] have developed suites of methods for illustrating various geometrical properties on analytically-defined surfaces, for purposes such as facilitating NC milling and evaluating surface “fairness”.

The direction taken in this paper was most fundamentally inspired by the elegant vector field visualization work that began with van Wijk’s introduction of spot noise [41] and Cabral and Leedom’s line integral convolution method [4], and advanced by Stalling and Hege [29] and others [1, 8, 15, 18, 28]. Line integral convolution is particularly attractive as a device for generating strokes through a volume because by advecting the empty (black) space in a texture volume of scan-converted points along with the full it is possible, by and large, to finesse the problem of appropriate streamline placement, at least as far as the æsthetic requirements of this particular application are concerned.

3 Defining The Texture

In many of the applications that call for the visualization of superimposed surfaces, it is necessary to view not just one but multiple level surfaces through a volume distribution. Sophisticated methods for improving the comprehensibility of a transparent surface via texture are of greatest practical utility in these cases when the texture used to convey surface shape is applicable throughout the volume and does not have to be derived separately for each level surface examined.

3.1 Distributing the particles

The first step in the process of defining the volume texture of principal direction strokes is the task of defining the evenly-distributed set of points that will be advected to form them. I try to approximate a minimum distance Poisson disk sampling distribution by applying a random jitter of 0-1 times the inter-element spacing of points on a uniform grid and throwing away and recomputing any sample that falls within a specified minimum distance of a previously computed neighbor. Because the points are processed in a predetermined order, there are only 13 possible points, out of all of the points already derived, that could conceivably be too close to any new candidate, so only 13 comparisons are needed to decide whether to accept or reject a particular random amount of jitter for each new point. As long as the minimum allowable distance between points is reasonably less than the inter-element spacing before jittering, this procedure turns out to be very efficient and has the advantage of producing a point distribution in which the number of samples contained within any plane of neighboring voxels is more or less equivalent, and at the same time avoids instances in which samples bunch up too closely.

The final set of points chosen defines the voxels of the input to LIC that will be turned “on” (set to 255). The remainder of the voxels are left “off” (set to zero). Introducing varying levels of grey into the input texture, unless for the purpose of representing larger particles, only complicates matters

unnecessarily. The resolution of the volume data will usually be coarse enough, relative to the resolution of the final image, that unit-width input points will produce strokes of ample thickness. However, wider strokes may sometimes be desirable for representing shading or some other variable, as will be discussed in section 6.3.

3.2 Defining the principal directions

In addition to defining a suitable set of particles to advect, it is necessary to define the vector field of principal directions along which the particles will be made to flow. Principal directions and principal curvatures are classical geometric measures that can be used to describe the local shape of a surface around any given point. Although they are amply described in almost any text on differential geometry, and various algorithms for defining them have been explained in great detail elsewhere in the literature [12, 16, 22], for the sake of completeness and to help make these concepts perhaps somewhat more easily accessible I will briefly restate the basic process and definitions given by Koenderink [16] and used in [12].

At any point on a smoothly curving surface, there will in general be one single direction in which the curvature of the surface is greatest. This direction is the first principal direction, and the curvature of the surface in this direction is the first principal curvature. The second principal direction is mutually orthogonal to both the first principal direction and to the surface normal, and represents the direction in which the surface curvature is most nearly flat. Starting from an orthogonal frame $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ at a point $P_{x,y,z}$ in which \bar{e}_1 and \bar{e}_2 are arbitrary orthogonal vectors lying in the tangent plane to the surface and \bar{e}_3 points in the surface normal direction, it is possible to determine the principal directions by diagonalizing the Second Fundamental Form, a matrix of partial derivatives

$$A = \begin{bmatrix} \tilde{\omega}_1^{13} & \tilde{\omega}_1^{23} \\ \tilde{\omega}_2^{13} & \tilde{\omega}_2^{23} \end{bmatrix},$$

where the elements $\tilde{\omega}_j^{i3}$ can be computed as the dot product of \bar{e}_i and the first derivative of the gradient in the \bar{e}_j direction. Specifically, diagonalizing A means computing the matrices

$$D = \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \text{ and } P = \begin{bmatrix} v_{1u} & v_{2u} \\ v_{1v} & v_{2v} \end{bmatrix},$$

where $A = PDP^{-1}$ and $|\kappa_1| > |\kappa_2|$. The principal curvatures are the eigenvalues κ_1 and κ_2 , and the principal directions are the corresponding eigenvectors, expressed in 3D object space coordinates as $\bar{e}'_i = v_{iu}\bar{e}_1 + v_{iv}\bar{e}_2$.

To ensure the best possible results it is useful to represent the gradients at full floating point precision, and use a Gaussian-weighted derivative operator over a 3x3x3 neighborhood rather than central differences when computing the values of $\tilde{\omega}_j^{i3}$. As an extra precaution, I enforce the expected equality of

$\tilde{\omega}_2^{13}$ and $\tilde{\omega}_1^{23}$ by replacing each of these twist terms by the average of the two of them before performing the diagonalization.

Where the surface is locally spherical (at points called umbilics) or locally planar, the principal directions will be undefined. These “non-generic” points arise relatively infrequently in nature, but of course are found everywhere over manmade surfaces. A texturing technique based on principal directions could potentially run into a lot of trouble if it were applied to an object made up of spheres and rectangular slabs. For this application, however, a few zeros are no problem, and the LIC program has to check for such points in any case.

A more significant issue is the fact that the orientation indicated by the first principal direction is actually an axis that can point either way. There is no way to guarantee the consistency of the particular chosen direction *a priori*; recognition of this must be built into the LIC program and taken into consideration during streamline tracing. In this implementation, I use unit length principal direction vectors for streamline tracing, and save the principal curvature values in a companion volume so that they may be accessed independently.

3.3 Advecting the particles via 3D LIC

The implementation of LIC that I use to obtain the scan-converted strokes is basically a straightforward 3D extension of the “fast-LIC” method described by Stalling and Hege [29]. Voxels are processed in block-sequential order, and streamlines are traced in both directions through each voxel using 4th-order Runge Kutta with maximum-limited adaptive step size control, and resampled at equally spaced points ($h_t=0.5$) via a cubic spline interpolation that preserves C^1 continuity.

During streamline tracing, I use a reference vector to keep track of the direction of the most recently obtained sample from the vector field of principal directions, and use comparisons with this vector to determine which of the two possible orientations of the first principal direction to select at each gridpoint before performing the trilinear interpolation to retrieve the next sample. When using a constant-length box filter kernel, I take advantage of the method suggested in [29] for incrementally computing the convolution integral, but go back to computing the convolution separately for each point when the filter length is allowed to vary.

Since most of the space in the input texture is empty, the average intensity at each voxel after LIC will be quite low. To avoid loss of precision and maintain a reasonable dynamic range in the grey levels of the output texture (which is necessary to avoid aliasing artifacts), one may either

- estimate an appropriate scaling factor, based on the length of the filter used for the convolution, to apply during the normalization step of the LIC (where final voxel intensity is set to the accumulated intensity divided by the number of streamlines contributing to this accumulation), or

- output the results as a floating point volume and use standard image processing utilities to window and rescale the results into an appropriate range.

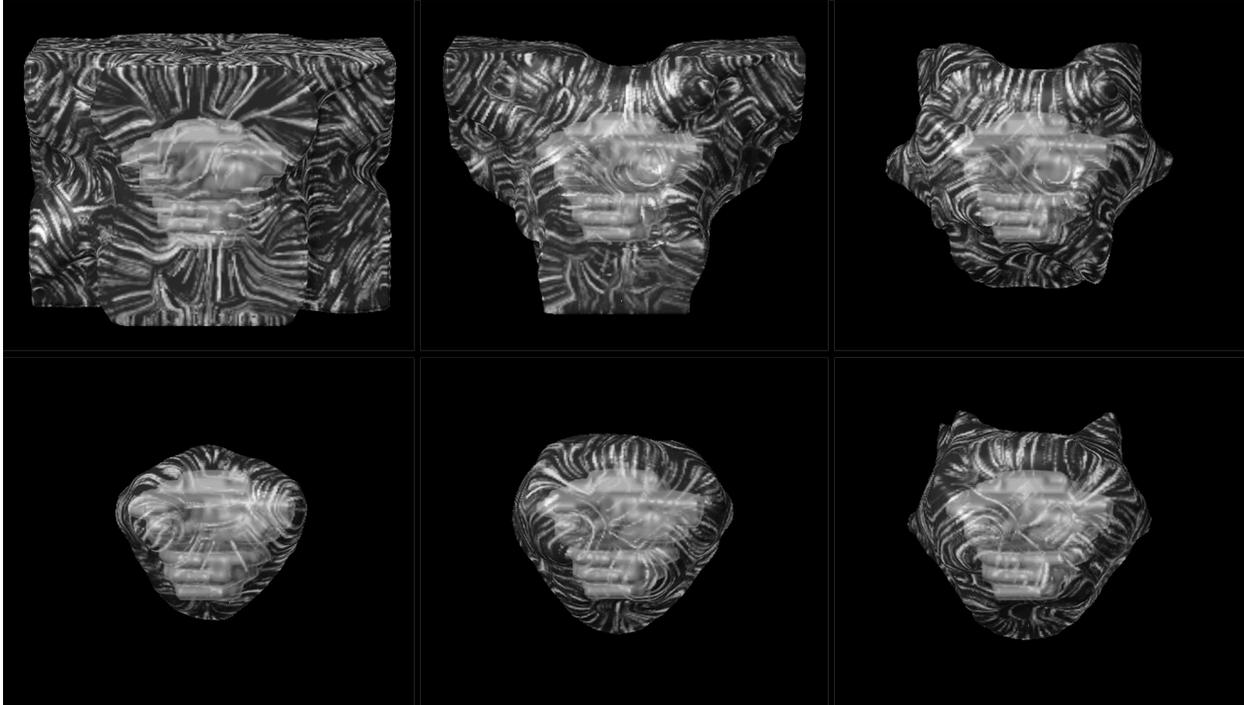


Figure 1: A single solid texture is applied to the volume data shown in each of these pictures. Nevertheless, the image of the texture on each isosurface conveys shape information specific to that surface. These images depict a series of level surfaces of radiation dose enclosing an opaque treatment region. Clockwise from the upper left, dose concentrations, relative to the prescribed level, are: 4%, 28%, 36%, 47%, 55%, and 82%.

4 Applying The Texture

Once a 3D scan-converted stroke texture is obtained, it can be used during rendering to selectively increase the opacity of the corresponding points on the transparent surface being displayed.

If the isosurface is defined by a marching cubes [21] triangulation, one may determine the amount of additional opacity to be added at any surface point by trilinearly interpolating from the values in the texture volume. If the isosurface is defined by a volume region of finite thickness, as described in [19], one must be careful to add the additional opacity indicated by the texture only to those voxels occupied or partially occupied by the isovalue contour surface.

One advantage of the polygonal representation produced by marching cubes is that it facilitates discounting all but the first occurrence of the transparent surface in depth along the viewing direction. I usually take advantage of this option during rendering to help simplify the images and avoid a “gauze curtain” look where the transparent surface overlaps itself multiple times in the projection. This technique was used to produce the images in figure 1. The “strokes” on each of the level surfaces shown in this figure are obtained from a single solid texture, defined in a 241x199x181 volume, equal in resolution to the dose data.

It is of course possible to define the resolution of the stroke texture to be several times finer than the resolution of the volume data defining the isosurface to which it is applied, and when this is done,

somewhat better-looking results may be achieved by using Levoy's [19] volume definition for the isosurface and displaying all strokes that fall within the isosurface region. This approach is illustrated by the image in figure 2. The opaque treatment region is left out of this particular image so that it may be easier to appreciate the detail of the strokes. The 3D nature of the irregularities in the positions of the tiny strokes in this kind of representation may also more aptly evoke a "hand-drawn" look, when that is the aim.



Figure 2: Narrower strokes may be represented via higher resolution textures. The resolution of the texture volume in this image is 482x398x362, twice as great as the resolution of the texture used to compute the images in figures 1 and 3-6.

5 Some Empirical Comparisons

The potential effectiveness of the proposed principal direction stroke texturing method may perhaps be most easily appreciated through comparison with alternatively rendered images. Figure 3 shows three different representations of the same pair of overlapping surfaces. On the left, the external transparent surface is left plain. The only cues to its shape are given by the subtle intensity variations in the diffuse surface shading and by the shapes and locations of the reflected specular highlights. There are no cues to the distance between the overlapping surface in this image, and the addition of stereo and/or motion can do little in this situation to improve the perceptibility of this depth information [14].

In the image on the right, a "solid grid" texture has been applied to increase the opacity of the external transparent surface along selected planar cross-sections. When this image is viewed in motion or in stereo, the relative depth distances between the opacified points on the external surface and points on the inner object become immediately apparent. However, it is not easy to obtain an intuitive

understanding of the overall surface shape from this representation; the grid lines demand our attention, and the directions they indicate are only indirectly related to features of the surface shape.

The image in the center shows a principal direction-driven 3D LIC stroke texture. At every point, the lines are oriented in the direction of maximum surface curvature. Important shape information is readily available even in this static image, and with the addition of stereo and motion cues, the shapes and depths of the two surfaces may be yet more easily and accurately perceived.



Figure 3: A comparison of methods for displaying overlapping surfaces. Left: a plain transparent isointensity surface of radiation dose surrounding an opaque treatment region. Center: the same dataset, with principal direction-driven 3D LIC texture added to the outer surface. Stroke lengths and widths in this image have each been adaptively defined according to the magnitude of the curvature in the stroke direction. Right: the same dataset, with a solid grid texture used to highlight selected contour curves.

6 Refining The Texture Definition

The quality of the shape description provided by the principal direction texturing approach described in this paper may be improved somewhat when curvature magnitude information is incorporated into the texture definition process, and other interesting effects may be achieved when stroke color and/or width are allowed to vary in accordance with the values of a second function over the volume data.

6.1 Stroke length

Stroke length is controlled by the length of the filter kernel used for convolving the intensities at successive points along each streamline. Figure 4, after figure 5 in [4], illustrates the effect on the stroke texture of using different constant values for this parameter.

Because the visual prominence of the indication of a specific direction should ideally reflect the significance of that particular direction, it can be advantageous to adaptively modify the length of the filter kernel applied at each point in correspondence with the magnitude of the first principal curvature there. The principal directions and principal curvatures have already been precomputed and stored for

every gridpoint, and it is relatively straightforward to define a mapping from relative curvature (κ_1/κ_{\max}) to filter kernel length that can be locally applied during the LIC. The effect of adaptively controlling stroke length in this fashion can be seen in the central image of figure 3 above.

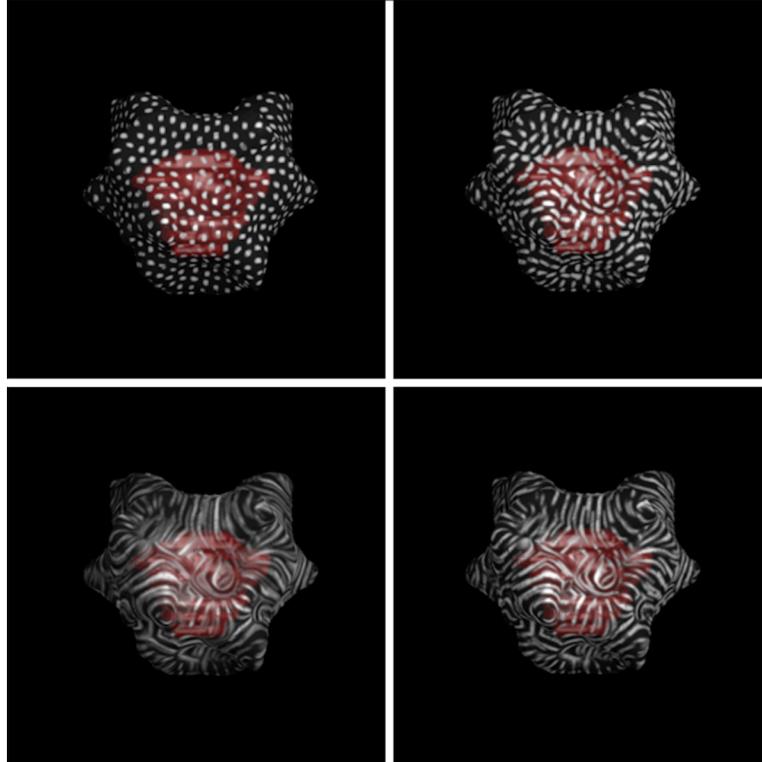


Figure 4: An illustration of the effect of filter kernel length on the lengths of the strokes in the texture. Clockwise from the upper left, these images were computed using filter kernel lengths of 2, 6, 20 and 40. The initial spots were defined by a point-spread function approximately four voxels in diameter, applied at evenly-distributed surface points in the 241x199x181 voxel volume.

6.2 Stroke width

Because the strokes are scan-converted, the resolution of the stroke texture volume fundamentally limits the narrowness with which any stroke may be represented. For example, if the texture volume is only 100 voxels wide, the thinnest stroke will occupy 1% of the total width of the image. To apply a finer stroke texture to surfaces obtained from more coarsely sampled data, it is necessary to compute a texture volume that has higher resolution than the data. To achieve a stroke texture of wider strokes, it is necessary to run the LIC on an input texture containing larger spots.

Stroke thickness in the scan-converted texture is directly related to the size of the spots advected by LIC. Supersampling during traditional LIC to get a higher resolution output won't result in thinner strokes unless the values interpolated from the input texture are windowed or ramped before being used. The best way to obtain a texture of very thin strokes is to either supersample the directional data and

index into a higher resolution input texture, which is the approach I have taken, or, as Battke et al. [1] suggest, to define the input texture procedurally, in which case resolution is not a limiting factor.

There are several different ways in which stroke width can be used to convey additional information about the volume data. One possibility is to vary stroke width according to a static variable such as the magnitude of the principal curvature at a point, as has been done in figure 3. Such an approach can be used to emphasize the stroke texture in the specific areas where the directional information that the texture indicates is most perceptually relevant and to play down the visual impact of texture direction in flatter regions, where it is less helpful for shape understanding. By using stroke width rather than stroke opacity to modulate the texture prominence, it is possible to maintain the impression of a continuous and coherent surface and to avoid imparting an ephemeral or “moth-eaten” look to the outer object.

It may alternatively be desirable to allow the stroke width to be locally determined by the value of a dynamically changing variable such as surface shading. While adaptive stroke width might be approximated in the former case by defining a single input texture containing spots of different sizes, as in the “multi-frequency” LIC texturing approach suggested by Kiu and Banks [15], to efficiently reflect the value of a dynamically changing function it is far preferable to precompute a short series of LIC texture volumes from inputs containing a succession of spot sizes at identical points, and then adaptively index into any particular one of these during rendering, depending on the value of the dynamically-defined function. The right-hand image in figure 5 was computed from a combination of five different LIC textures, and the particular texture applied at each point was determined by the value of the local illumination, illustrated in the picture to its left. An additional advantage this approach is that it allows quick and easy experimentation with different function value to stroke width mappings. While computing the 3D LIC can be a fairly slow process, recombining the precomputed texture volumes is really fast.

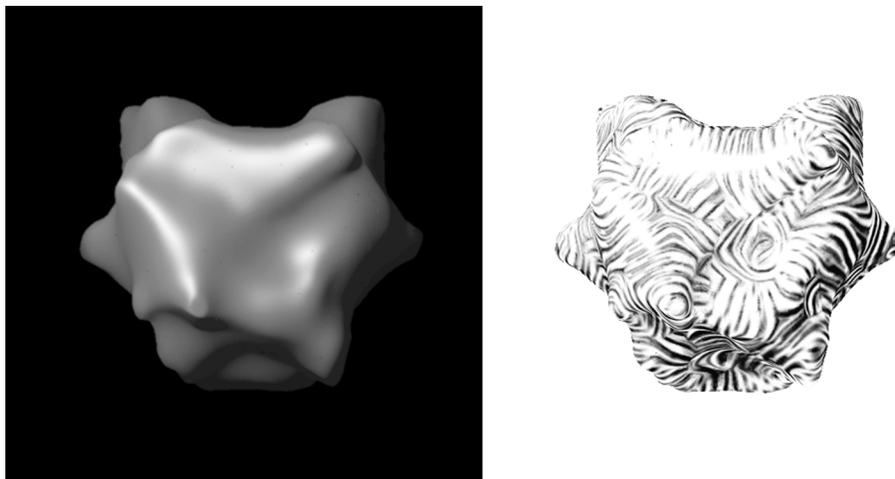


Figure 5: The width of a stroke at any point along its extent may be adaptively determined, at the time of rendering, by selecting texture values from any of a series of multiple predefined volumes, indexed by the value of a second function computed over the data. In this case the shading at each point is used to determine the volume from which the local texture sample is retrieved.

6.3 Stroke color

The most effective use of color (hue) in this application is as a label. Color can be used to help differentiate the inner object from the outer, or to convey information about a third scalar distribution. One particularly effective use of color for this application, which is shown in figure 6, is as an indicator of the depth distance between the outer and inner surfaces. By allowing hue to vary along with luminance in an approximation of a heated-object scale it is possible to convey the locations and distribution of “near” and “far” regions in a particularly intuitive manner, in which the luminance ordering of the different colors can be subjectively related to the amount of distance each color represents.

6.4 Limiting texture computations to a region of interest in the volume

One of the principal advantages of the texturing technique described in this paper is its applicability in situations where one needs to view arbitrary level surfaces in a 3D volume dataset. However, it may also be used with some efficiency when more limited regions of interest (ROI) are defined. In such cases one may evenly distribute input points among the voxels within the ROI, and trace streamlines around the voxels in the ROI only. The images in figures 4 and 6 were computed using such an approach; in these cases the texture is applicable only to a limited subset of the isovalue surfaces in the volume.

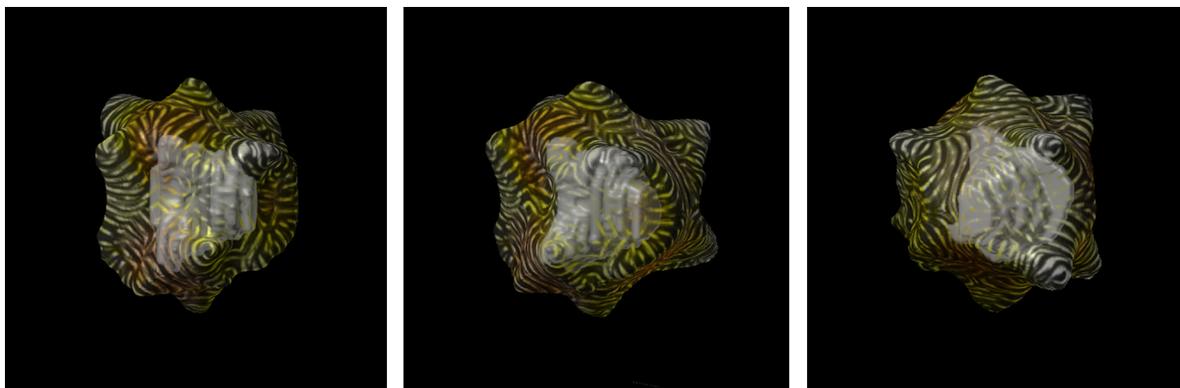


Figure 6: Color is used here to convey the relative magnitude of the depth distance between the two superimposed surfaces. Strokes are whitest where the surfaces are relatively widely separated, and become progressively redder as the proximity of the outer surface to the inner increases. Stroke length varies according to the magnitude of the principal curvature, as in figure 3, but stroke width is held constant. Spot diameter before LIC was about 3 voxels.

7 Conclusions And Future Work

Line direction is an essential element in surface shape description. An appropriate use of line can reveal the curvature of a 3D form in a single static image; inappropriate uses of line can make smoothly curving surfaces appear flattened or distorted, even when binocular disparity cues provide veridical depth

information. Artists and illustrators have historically emphasized the importance of stroke direction in line drawing, advising that “as a general rule, a subject offers some hint as to a natural arrangement of lines” [11] and warning that

... all a fastidious spectator’s pleasure in a drawing may be destroyed by a wrong use of direction ... no matter how fine the lines composing it may be, or how pretty the general effect [34].

This paper has described how the set of principal directions and principal curvatures, classical shape descriptors from differential geometry, can be used to define a natural flow of lines over the surface of an object, and used to guide the placement of a stroke texture that seeks to reveal shape in a perceptually intuitive way.

The method described here is fully automatic, easy to implement, and requires very little fine-tuning. The strokes are defined as static entities in 3D space, and when applied to the surface they create a texture pattern that is stable under changes in viewpoint or object orientation. The problem of defining an even stroke distribution, and avoiding the unæsthetic merging and colliding of elements, is finessed by the tracing, via LIC, of the empty space along with the full space in a volume of approximately Poisson-disk distributed point samples. A few simple parameters can be adjusted to globally or locally control seed point spacing, stroke length and stroke width, and the resulting scan-converted texture will be applicable to all level surfaces in a smooth volume distribution, facilitating data exploration. When investigations are known *a priori* to be limited to a specific region of interest within the volume, this ROI information can be easily incorporated into the texture definition process so that excess calculations may be avoided.

Acknowledgments

This work was supported by the National Aeronautics and Space Administration under NASA contract number NAS1-19480. The author thanks Kwan-Liu Ma for encouragement and helpful suggestions. Marc Levoy wrote the volume rendering platform within which this work was implemented, and Jim Chung provided the code from which the marching cubes isosurface extraction was developed. Radiation therapy data was provided by Dr. Julian Rosenman, Department of Radiation Oncology, University of North Carolina- Chapel Hill.

References

- [1] H. Battke, D. Stalling, H.-C. Hege. “Fast Line Integral Convolution for Arbitrary Surfaces in 3D”, Visualization and Mathematics, H.-C. Hege and K. Polthier, eds., Springer Verlag, 1997.
- [2] James M. Beck, Rida T. Farouki and John K. Hinds. “Surface Analysis Methods”, IEEE Computer Graphics and Applications, vol. 6, no. 12, December 1986, pp. 18-36.
- [3] Myron L. Braunstein and John W. Payne. “Perspective and Form Ratio as Determinants of Relative Slant Judgments”, Journal of Experimental Psychology, vol. 81, no. 3, pp. 584-590, 1969.
- [4] Brian Cabral and Leith (Casey) Leedom. “Imaging Vector Fields Using Line Integral Convolution”, Computer Graphics Proceedings, Annual Conference Series, 1993, pp. 263-270.
- [5] Bruce G. Cumming, Elizabeth B. Johnston and Andrew J. Parker. “Effects of Different Texture Cues on Curved Surfaces Viewed Stereoscopically”, Vision Research, vol. 33, no. 5/6, pp. 827-838, 1993.

- [6] James E. Cutting and Robert T. Millard. "Three Gradients and the Perception of Flat and Curved Surfaces", *Journal of Experimental Psychology: General*, vol. 113, no. 2, pp. 198-216, 1984.
- [7] Debra Dooley and Michael F. Cohen. "Automatic Illustration of 3D Geometric Models: Surfaces", *Proc. Visualization '90*, pp. 307-313.
- [8] Lisa K. Forsell. "Visualizing Flow Over Curvilinear Grid Surfaces Using Line Integral Convolution", *Proc. Visualization '94*, pp. 240-247.
- [9] Howard R. Flock and Anthony Moscatelli. "Variables of Surface Texture and Accuracy of Space Perceptions", *Perceptual and Motor Skills*, vol. 19, pp. 327-334, 1964.
- [10] James J. Gibson. "The Perception of Visual Surfaces", *American Journal of Psychology*, vol. 63, pp. 367-384, 1950.
- [11] Arthur Gupptill. Rendering in Pen and Ink, Watson-Gupptill Publications, 1976.
- [12] Hans Hagen, Stefanie Hahmann, Thomas Schreibern, Yasuo Nakajima, Bukard Wördenweber and Petra Hollemann-Grundstedt. "Surface Interrogation Algorithms", *IEEE Computer Graphics and Applications*, vol. 12, no. 5, September 1992, pp. 53-60.
- [13] Victoria Interrante, Henry Fuchs and Stephen Pizer. "Enhancing Transparent Skin Surfaces with Ridge and Valley Lines", *Proc. Visualization '95*, pp. 52-59.
- [14] Victoria Interrante, Henry Fuchs and Stephen Pizer. "Illustrating Transparent Surfaces with Curvature-Directed Strokes", *Proc. Visualization '96*, pp. 211-218.
- [15] Ming-Hoe Kiu and David C. Banks. "Multi-Frequency Noise for LIC", *Proc. Visualization '96*, pp. 121-126.
- [16] Jan Koenderink. Solid Shape, MIT Press, 1990.
- [17] Jan J. Koenderink and Andrea J. van Doorn. "Relief: pictorial and otherwise", *Image and Vision Computing*, vol. 13, no. 5, June 1995, pp. 321-334.
- [18] Willem C. de Leeuw and Jarke J. van Wijk. "Enhanced Spot Noise for Vector Field Visualization", *Proc. Visualization '95*, pp. 233-239.
- [19] Marc Levoy. "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications*, vol. 8, no. 3, May 1988, pp. 29-37.
- [20] Marc Levoy, Henry Fuchs, Stephen Pizer, Julian Rosenman, Edward L. Chaney, George W. Sherouse, Victoria Interrante and Jeffrey Kiel. "Volume Rendering in Radiation Treatment Planning", *First Conference on Visualization in Biomedical Computing, 1990*, pp. 4-10.
- [21] William Lorensen and Harvey Cline. "Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm", *Computer Graphics*, vol. 21, no. 4, July 1987, (SIGGRAPH '87 Conference Proceedings), pp. 163-169.
- [22] Olivier Monga, Serge Benayoun and Olivier D. Faugeras. "From Partial Derivatives of 3D Density Images to Ridge Lines", *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1992*, pp. 354-359.
- [23] Henry P. Moreton. "Simplified Curve and Surface Interrogation via Mathematical Packages and Graphics Libraries and Hardware", *Computer-Aided Design*, vol. 27, no. 7, pp. 523-543, 1995.
- [24] Darwyn Peachey. "Solid Texturing of Complex Surfaces", *Computer Graphics*, vol. 19, no. 3, July 1985, (SIGGRAPH '85 Conference Proceedings), pp. 279-286.
- [25] Ken Perlin. "An Image Synthesizer", *Computer Graphics*, vol. 19, no. 3, July 1985, (SIGGRAPH '85 Conference Proceedings), pp. 287-296.
- [26] Penny Rheingans. "Opacity-modulating Triangular Textures for Irregular Surfaces", *Proc. Visualization '96*, pp. 219-225.
- [27] Takafumi Saito and Tokiichiro Takahashi. "Comprehensible Rendering of 3-D Shapes", *Computer Graphics*, vol. 24, no. 4, August 1990, (SIGGRAPH '90 Conference Proceedings), pp. 197-206.

- [28] Han-Wei Shen, Christopher R. Johnson and Kwan-Liu Ma. “Visualizing Vector Fields Using Line Integral Convolution and Dye Advection”, Proc. 1996 Symposium on Volume Visualization, pp. 63-70.
- [29] Detlev Stalling and Hans-Christian Hege. “Fast and Resolution Independent Line Integral Convolution”, Computer Graphics Proceedings, Annual Conference Series, 1995, pp. 249-256.
- [30] Kent A. Stevens and Allen Brookes. “Probing Depth in Monocular Images”, Biological Cybernetics, vol. 56, pp. 355-366, 1987.
- [31] Kent A. Stevens. “The Information Content of Texture Gradients”, Biological Cybernetics, vol. 42, pp. 95-105, 1981.
- [32] Kent A. Stevens. “The Visual Interpretation of Surface Contours”, Artificial Intelligence, vol. 17, pp. 47-73, 1981.
- [33] James V. Stone. “Shape From Local and Global Analysis of Texture”, Philosophical Transactions of the Royal Society of London, B, vol. 339, pp. 53-65, 1993.
- [34] Edmund J. Sullivan. Line; an art study, Chapman & Hall, 1922.
- [35] James T. Todd and Robin Akerstrom. “Perception of Three-Dimensional Form from Patterns of Optical Texture”, Journal of Experimental Psychology: Human Perception and Performance, vol. 13, no. 2, pp. 242-255, 1987.
- [36] James T. Todd and Francene D. Reichel. “Ordinal Structure in the Visual Perception and Cognition of Smoothly Curved Surfaces”, Psychological Review, vol. 96, no. 4, pp. 643-657, 1989.
- [37] James T. Todd and Francene D. Reichel. “Visual Perception of Smoothly Curved Surfaces from Double-Projected Contour Patterns”, Journal of Experimental Psychology: Human Perception and Performance, vol. 16, no. 3, pp. 665-674, 1990.
- [38] Greg Turk. “Generating Textures for Arbitrary Surfaces Using Reaction-Diffusion”, Computer Graphics, vol. 25, no. 4, July 1991 (SIGGRAPH '91 Conference Proceedings), pp. 289-298.
- [39] Greg Turk and David Banks. “Image-Guided Streamline Placement”, Computer Graphics Proceedings, Annual Conference Series, 1996, pp. 453-460.
- [40] Victor Vasarely. Vasarely III, Éditions du Griffon Neuchâtel, 1974.
- [41] Jarke J. van Wijk. “Spot Noise-Texture Synthesis for Data Visualization”, Computer Graphics, vol. 25, no. 4, July 1991 (SIGGRAPH '91 Conference Proceedings), pp. 309-318.
- [42] Georges Winkenbach and David H. Salesin. “Computer-Generated Pen-and-Ink Illustrations”, Computer Graphics Proceedings, Annual Conference Series, 1994, pp. 91-100.
- [43] Georges Winkenbach and David H. Salesin. “Rendering Parametric Surfaces in Pen and Ink”, Computer Graphics Proceedings, Annual Conference Series, 1996, pp. 469-476.
- [44] Andrew P. Witkin. “Recovering Surface Shape and Orientation from Texture”, Artificial Intelligence, vol. 17, 1981, pp. 17-45.
- [45] Andrew Witkin and Michael Kass. “Reaction-Diffusion Textures”, Computer Graphics, vol. 25, no. 4, July 1991 (SIGGRAPH '91 Conference Proceedings), pp. 299-308.