

Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications

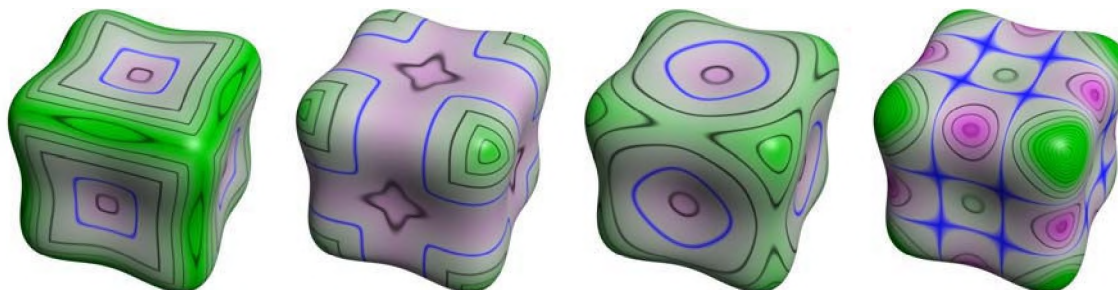
Gordon Kindlmann¹Ross Whitaker¹Tolga Tasdizen¹Torsten Möller²¹ Scientific Computing and Imaging Institute, University of Utah² Graphics, Usability, and Visualization (GrUVi) Lab, Simon Fraser University

Figure 1: Volume renderings of a 64^3 synthetic volume with four different curvature measures. Left to right: first principal curvature κ_1 , second principal curvature κ_2 , mean curvature $(\kappa_1 + \kappa_2)/2$, and Gaussian curvature $\kappa_1 \kappa_2$. Magenta indicates negative curvature, green indicates positive. Iso-curvature contours are in black, except for zero curvature in blue.

Abstract

Direct volume rendering of scalar fields uses a transfer function to map locally measured data properties to opacities and colors. The domain of the transfer function is typically the one-dimensional space of scalar data values. This paper advances the use of curvature information in multi-dimensional transfer functions, with a methodology for computing high-quality curvature measurements. The proposed methodology combines an implicit formulation of curvature with convolution-based reconstruction of the field. We give concrete guidelines for implementing the methodology, and illustrate the importance of choosing accurate filters for computing derivatives with convolution. Curvature-based transfer functions are shown to extend the expressivity and utility of volume rendering through contributions in three different application areas: non-photorealistic volume rendering, surface smoothing via anisotropic diffusion, and visualization of isosurface uncertainty.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism I.4.7 [Computing Methodologies]: Image Processing and Computer Vision—Feature Measurement

Keywords: volume rendering, implicit surface curvature, convolution-based differentiation, non-photorealistic rendering, surface processing, uncertainty visualization, flowline curvature

¹ {gk|whitaker|tolga}@cs.utah.edu

² torsten@cs.sfu.ca

IEEE Visualization 2003,
October 19-24, 2003, Seattle, Washington, USA
0-7803-8120-3/03/\$17.00 ©2003 IEEE

1 Introduction

Direct volume rendering depicts structure in scalar fields through a simple combination of mappings. At each rendered sample point, locally measured numerical quantities (mainly the data value itself) are mapped via the transfer function to optical quantities such as opacity and color. Basic computer graphics routines can then shade, composite, and project the samples into a coherent visualization.

In nearly all applications, some part of this process involves the first-order differential structure of the field: the *gradient*. Shading routines generally require the gradient direction as input to a surface-based lighting model. There is also a trend towards using gradient magnitude in the transfer function domain, enabling visualization of material boundaries based on data value and edge strength [Levoy 1988]. In either case, first-order differential field structure enhances the clarity and effectiveness of direct volume rendering. This paper further promotes the use of differential structure in direct volume rendering, by demonstrating a family of transfer functions based on *curvature*, using second-order derivatives.

The theory behind curvature measurement in volume visualization draws from two distinct fields of study. The first is differential geometry. We compute isosurface curvature directly from the volume data using a simple algebraic framework of differential invariants. Additional insight into the underlying field structure comes from geometry orthogonal to the isosurfaces, which is also described by an invariant. The second field of study guiding this work is signal processing and filter design. Numerical differentiation accentuates noise, and the problem increases with the order of the derivative. Care must be taken to measure second derivatives robustly and accurately. We describe a convolution-based derivative measurement scheme which allows control of the tradeoff between accuracy, continuity, and filter size. This curvature measurement method obviates intermediate surface construction, surface parameterization, and explicit finite difference schemes. The ability to compute reliable curvature information at arbitrary points in sampled volume data creates new opportunities for direct volume rendering with curvature-based transfer functions.

This paper selects different components of curvature information to serve as domain variables in multi-dimensional transfer func-

tions. We demonstrate that curvature-based transfer functions enhance the expressive and informative power of direct volume rendering with contributions in three different application areas. Non-photorealistic volume rendering is improved in Section 5 with contours exhibiting constant thickness in image space, and a flexible method of emphasizing ridge and valley surface creases. Section 6 shows how volume rendering with total curvature provides a technique for studying the behavior of volumetric surface processing algorithms. Section 7 uses flowline curvature as a way to qualitatively indicate regions of high geometric uncertainty in surface models created by isosurfacing.

2 Previous Work

This paper builds on a significant amount of previous research in volumetric curvature measurement and its application to volume visualization. Monga et al. [1992; 1994] use convolution with Gaussian kernels, as well as their first, second, and third derivatives to compute surface curvature characteristics, and to find ridge lines (extrema in normal curvature along the principal curvature direction). Interrante et al. [1995] use ridge and valley lines to simplify depiction of the skin surface for surgical planning. Their later work [1997; 1997] systematically explores a variety of methods for using strokes or textures aligned with principal curvature directions to effectively convey the configuration of semi-transparent isosurfaces relative to internal structures. Kinetic visualization [Lum et al. 2003] uses particle motion along the principal curvature direction to clarify depiction of surface orientation. In the work that most directly influences our own, Hladůvka et al. [2000] describe volume renderings with two-dimensional transfer functions using a simplified space of principal curvatures, and demonstrate that surface shape is an intuitive space in which to specify color and opacity.

Another thread of previous work adapts non-photorealistic rendering (NPR) techniques to direct volume rendering [Gooch and Gooch 2001; Ma et al. 2002]. Treavett and Chen [2000] create pen-and-ink style visualizations of volume data with a combination of volume-space and image-space measurements, including a simple curvature measure. Rheingans and Ebert [2001] modify the standard volume rendering pipeline by applying a variety of NPR techniques to enhance interesting features and regions, and to provide depth and orientation cues. Csébfalvi et al. [2001] use gradient information to create interactive visualizations emphasizing object contours, quickly revealing internal structures with little parameter tuning. Lu et al. [2002] mimic stipple drawing by using a variety of feature detection methods to place point primitives in the volume with carefully controlled density and location. Lum and Ma [2002] use commodity graphics hardware to create richly detailed renderings of volume datasets using a combination of NPR methods. Nagy et al. [2002] use graphics hardware to enrich interactive volume renderings with artistic shading methods and curvature-directed strokes and hatches.

3 Curvature Measurement

The contribution of this section is a self-contained derivation and intuitive explanation of our curvature measurement method, ending with the three simple steps required to put it into practice. The steps can be implemented without perfect understanding of the underlying mathematics. Following Monga et al. [1992], we rely on convolution with continuous filters, rather than an explicitly constructed coordinate frame aligned with the gradient direction [Interrante 1997; Nagy et al. 2002]. We feel our convolution-based approach is significantly simpler than a previous method based on characterizing multiple planar curves passing through the point of interest [Hladůvka et al. 2000]. Though visualization results may bene-

fit from a pre-process smoothing of the data, no pre-computation is required for the curvature measurement itself.

The curvature of a surface is defined by the relationship between small positional changes on the surface, and the resulting changes in the surface normal. In sampled volume data, surfaces are implicitly represented as isosurfaces of reconstructed continuous data values $f(\mathbf{x})$. Assuming that the values of f increase as we move further inside objects of interest (e.g., a standard CT scan), the surface normal is defined as $\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$, with the gradient $\mathbf{g} = \nabla f = \left[\frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \frac{\partial f}{\partial z} \right]^T$. Curvature information is contained in $\nabla \mathbf{n}^T$, a 3×3 matrix. However, we do not want to evaluate the gradient of a (pre-computed) normalized vector, since this hinders direct convolution-based measurement of the original data. Expanding upon the derivation in [Mitchell and Hanrahan 1992]:

$$\begin{aligned} \nabla \mathbf{n}^T &= -\nabla \left(\frac{\mathbf{g}^T}{|\mathbf{g}|} \right) = -\left(\frac{\nabla \mathbf{g}^T}{|\mathbf{g}|} - \frac{\mathbf{g}^T \nabla |\mathbf{g}|}{|\mathbf{g}|^2} \right) \\ &= -\frac{1}{|\mathbf{g}|} \left(\mathbf{H} - \frac{\mathbf{g} \nabla^T (\mathbf{g}^T \mathbf{g})^{1/2}}{|\mathbf{g}|} \right) = -\frac{1}{|\mathbf{g}|} \left(\mathbf{H} - \frac{\mathbf{g} \nabla^T (\mathbf{g}^T \mathbf{g})}{2|\mathbf{g}|(\mathbf{g}^T \mathbf{g})^{1/2}} \right) \\ &= -\frac{1}{|\mathbf{g}|} \left(\mathbf{H} - \frac{\mathbf{g} (2\mathbf{g}^T \mathbf{H})}{2|\mathbf{g}|^2} \right) = -\frac{1}{|\mathbf{g}|} \left(\mathbf{I} - \frac{\mathbf{g} \mathbf{g}^T}{|\mathbf{g}|^2} \right) \mathbf{H} \\ &= -\frac{1}{|\mathbf{g}|} (\mathbf{I} - \mathbf{n} \mathbf{n}^T) \mathbf{H}. \end{aligned}$$

\mathbf{I} is the identity matrix, and \mathbf{H} is the Hessian matrix:

$$\mathbf{H} = \begin{bmatrix} \partial^2 f / \partial x^2 & \partial^2 f / \partial x \partial y & \partial^2 f / \partial x \partial z \\ \partial^2 f / \partial x \partial y & \partial^2 f / \partial y^2 & \partial^2 f / \partial y \partial z \\ \partial^2 f / \partial x \partial z & \partial^2 f / \partial y \partial z & \partial^2 f / \partial z^2 \end{bmatrix}. \quad (1)$$

The outer product of \mathbf{n} with itself, $\mathbf{n} \mathbf{n}^T$, is a linear operator that projects onto the one-dimensional span of \mathbf{n} . $\mathbf{I} - \mathbf{n} \mathbf{n}^T$ projects onto the orthogonal complement of the span of \mathbf{n} , namely, the tangent plane to the isosurface. Letting $\mathbf{P} = \mathbf{I} - \mathbf{n} \mathbf{n}^T$, then

$$\nabla \mathbf{n}^T = -\frac{1}{|\mathbf{g}|} \mathbf{P} \mathbf{H}. \quad (2)$$

Reading Equation 2 from right to left permits some intuitive understanding of $\nabla \mathbf{n}^T$. Vector calculus tells us that the Hessian matrix \mathbf{H} represents how the gradient \mathbf{g} changes as a function of infinitesimal changes of position in \mathbb{R}^3 [Marsden and Tromba 1996]. The changes in \mathbf{g} have a component along \mathbf{g} (the gradient can change length), and a component within the tangent plane (the gradient can change direction). For the purposes of describing curvature, only the latter component matters. It can be isolated with left-multiplication by \mathbf{P} . Finally, the $-1/|\mathbf{g}|$ scaling factor converts infinitesimal changes of the (un-normalized) gradient \mathbf{g} into infinitesimal changes of the unit-length normal \mathbf{n} .

Both \mathbf{P} and \mathbf{H} are symmetric matrices, but in general, $\nabla \mathbf{n}^T$ is not symmetric. However, if \mathbf{v} lies in the tangent plane, then $\mathbf{P} \mathbf{v} = \mathbf{v}$ and $\mathbf{v}^T \mathbf{P} = \mathbf{v}^T$, so for \mathbf{u} and \mathbf{v} in the tangent plane,

$$\mathbf{v}^T \mathbf{P} \mathbf{H} \mathbf{u} = \mathbf{v}^T \mathbf{H} \mathbf{u} = \mathbf{u}^T \mathbf{H} \mathbf{v} = \mathbf{u}^T \mathbf{P} \mathbf{H} \mathbf{v}.$$

That is, the restriction of $\nabla \mathbf{n}^T = -\mathbf{P} \mathbf{H} / |\mathbf{g}|$ to the tangent plane is symmetric, and thus there exists an orthonormal basis $\{\mathbf{p}_1, \mathbf{p}_2\}$ for the tangent plane in which $\nabla \mathbf{n}^T$ is a 2×2 diagonal matrix [Hoffman and Kunze 1971]. This basis can be easily extended to an orthonormal basis for all of \mathbb{R}^3 , $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{n}\}$. In this basis, the derivative of the surface normal is

$$\nabla \mathbf{n}^T = \begin{bmatrix} \kappa_1 & 0 & \sigma_1 \\ 0 & \kappa_2 & \sigma_2 \\ 0 & 0 & 0 \end{bmatrix}.$$

The bottom row is all zero because no change in position can make the normal \mathbf{n} change in length. Motion within the tangent

plane, along \mathbf{p}_1 and \mathbf{p}_2 , leads to changes of \mathbf{n} along the same directions, with proportionalities κ_1 and κ_2 respectively. There is no cross-term, or twisting of the normal, by the choice of $\{\mathbf{p}_1, \mathbf{p}_2\}$. By the definition of surface curvature [Do Carmo 1976], \mathbf{p}_1 and \mathbf{p}_2 are the principal curvature directions, while κ_1 and κ_2 are the principal curvatures. As one moves *along* the normal, off of the surface or deeper into it, the normal tilts according to σ_1 and σ_2 . This aspect of implicit surface curvature, termed *flowline curvature* by ter Haar Romeny et al. [1991], is explored further in Section 7. The above derivation of $\nabla \mathbf{n}^T$ is not novel. Others have exploited the fact that it has eigenvalues κ_1 , κ_2 , and 0 to compute curvature of implicit surfaces [Belyaev et al. 1998; Preusser and Rumpf 2002].

Multiplying $\nabla \mathbf{n}^T$ by \mathbf{P} has the effect of isolating κ_1 and κ_2 in the $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{n}\}$ basis:

$$\mathbf{G} = \nabla \mathbf{n}^T \mathbf{P} = \nabla \mathbf{n}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

Our surface curvature measurements are based on \mathbf{G} , which we term the *geometry tensor*. In practice, \mathbf{G} will be known only in the (X, Y, Z) basis of the volume axes, and will not have the readily transparent form of Equation 3. Matrix invariants provide the leverage to extract the desired curvature values κ_1 and κ_2 from \mathbf{G} , regardless of the principal curvature direction coordinate frame. The trace of \mathbf{G} is $\kappa_1 + \kappa_2$. The Frobenius norm of \mathbf{G} , notated $|\mathbf{G}|_F$ and defined as $\sqrt{\text{trace}(\mathbf{G}\mathbf{G}^T)}$, is $\sqrt{\kappa_1^2 + \kappa_2^2}$. κ_1 and κ_2 are then found with the quadratic formula.

To summarize, here are the steps needed to compute curvature at an arbitrary point in a scalar field:

1. Measure the first partial derivatives comprising the gradient \mathbf{g} . Compute $\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$, and $\mathbf{P} = \mathbf{I} - \mathbf{n}\mathbf{n}^T$.
2. Measure the second partial derivatives comprising the Hessian \mathbf{H} (Equation 1). Compute $\mathbf{G} = -\mathbf{P}\mathbf{H}\mathbf{P}/|\mathbf{g}|$.
3. Compute the trace T and Frobenius norm F of \mathbf{G} . Then,

$$\kappa_1 = \frac{T + \sqrt{2F^2 - T^2}}{2}, \quad \kappa_2 = \frac{T - \sqrt{2F^2 - T^2}}{2}.$$

If the data values inside regions of interest are *lower* than the background (e.g., inverted from a standard CT scan), the only change in the formulation is the sign of the geometry tensor: $\mathbf{G} = \mathbf{P}\mathbf{H}\mathbf{P}/|\mathbf{g}|$. Though not used in this paper, the principal curvature directions are easily found as eigenvectors of \mathbf{G} . The most important task in the curvature computation is measuring all the necessary partial derivatives (in \mathbf{g} and \mathbf{H}). It is for this task that we use convolution with continuous filters.

4 Measuring Derivatives with Convolution

Image-order volume rendering of discretely sampled data relies on convolution with continuous filters to reconstruct values at arbitrary locations, such as sample points along a ray. We perform three-dimensional reconstruction as a separable product of one-dimensional convolutions along the X, Y, Z axes. The combination of filters used on each axis determines whether the reconstructed quantity is the interpolated data value, or one of its many partial derivatives. Measuring $\partial^2 f / \partial x \partial y$, for example, is done with first-order derivative filters on the X and Y axes, and with a zero-order derivative (interpolation) filter on the Z axis. Accurate curvature measurement depends on a judiciously chosen combination of filters for zero, first, and second derivatives. For reasons of practicality and efficiency, we use piecewise polynomial filters with symmetric support. We apply the filter design framework of Möller

et al. [1998] to inform the choice of filters. It is the first time the framework has been used to create second derivative filters.

We review the spatial filter design framework by considering a continuous function $f(t)$ in one dimension. Sampling $f(t)$ at unit intervals produces a sequence of discrete values, notated $f[k]$, where k is an integer. From $f[k]$ we must approximate both the original function $f(t)$, as well as its first and second derivatives $f'(t)$ and $f''(t)$. Each of the three approximations is done by convolution with a continuous filter. The result of convolving $f[k]$ with continuous filter $w(t)$ is notated $f_r^w(t)$. By combining the N^{th} order Taylor expansion of f with the convolution definition of $f_r^w(t)$, Möller et al. [1998] derive:

$$f_r^w(t) = a_0^w f(t) + a_1^w f'(t) + a_2^w f''(t) + \dots + a_N^w f^{(N)}(t) + r_N^w \quad (4)$$

The a_n^w are termed *error coefficients*. From a formulaic standpoint, a_n^w encapsulates a convolution of w with the n -th term of the sampled Taylor expansion of f .

From a practical standpoint, however, the vector of error coefficients is the means by which we “dial up” a filter w , according to the desired derivative and accuracy properties. We describe both of these in turn. Looking at Equation 4, we can see that if a filter w does a perfect job of reconstructing $f(t)$, then $f_r^w(t) = f(t)$, so $a_0^w = 1$ and all other a_n^w (as well as the remainder term r_N^w) are zero. Similarly, for ideal first (or second) derivative filters, a_1^w (or a_2^w) is unity, and all other coefficients are zero. The measurement behavior of a filter w is governed by its first non-zero error coefficient. To specify that w should be a d -order derivative filter, we require $a_n^w = 0$ for all $n < d$, and set $a_d^w = 1$.

In spatial filter design, filter quality is quantified in terms of *accuracy*: the more accurate a filter is, the higher degree polynomial it can reconstruct exactly. A k^{th} degree error filter (k -EF) filter is one which can exactly reconstruct polynomials of degree $k-1$ or lower. In the context of Equation 4, we gain accuracy by requiring one or more of the error coefficients beyond a_d^w to be zero. To specify that w should be a d -derivative k -EF filter, we let $N = d + k$ and require $a_n^w = 0$ for all $n : d < n < N$. Note that the number of constraints we set (that is, the number of a_n^w we specify) is the sum of the derivative order and the accuracy level. This also turns out to be the lower bound on the filter support (the number of data value samples required in the convolution sum).

Continuity is also a desirable filter property, since it avoids visual artifacts in reconstruction, and it helps reduce post-aliasing [Marschner and Lobb 1994]. A continuous filter is C^M if it has M continuous derivatives. It can be shown that infinitely many piecewise polynomial filters satisfy any given derivative, accuracy, and continuity requirements [Möller 1999]. The filter $w(t)$ is ultimately implemented as a list of coefficients of piecewise polynomials in t . The coefficients are found as the solution of a linear system embodying the derivative, accuracy, and continuity properties described above [Möller 1999].

Spatial filter design supports the creation and characterization of second derivative filters, but unfortunately curvature is not simply a second derivative. Rather, Section 3 showed how curvature is extracted from both first and second derivatives, through algebraic transforms. We have conducted an empirical study to better understand how filter quality affects curvature measurements. Our study used measurements of mean curvature $(\kappa_1 + \kappa_2)/2$ on the Marschner-Lobb [1994] dataset, a 40^3 sample floating-point volume with isosurfaces resembling a graph of the sinc function. Figures 2 shows the analytical form of the data, including colormapping with correct mean curvature values.

Figure 3 shows results from a variety of filter combinations. Figure 3(a) uses the Catmull-Rom cubic spline and its first and second derivatives. Because the Catmull-Rom spline is C^1 but not C^2 ,

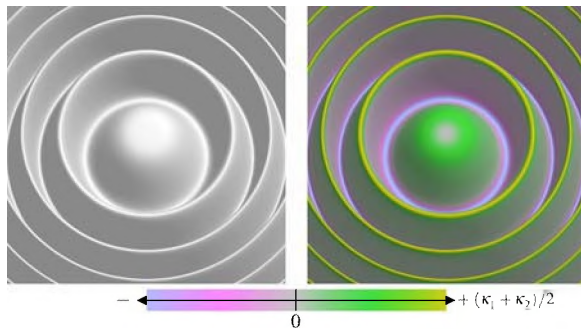


Figure 2: Marschner-Lobb dataset with ideal reconstruction. In (b), colors indicate mean curvature.

its second derivative is discontinuous, leading to markedly discontinuous curvature measurements. This example demonstrates how well-regarded reconstruction filters are not always the basis of useful derivative reconstruction filters, especially in the context of second derivatives. Figure 3(b), on the other hand, better captures the shape and curvature on the thin ridges, but at the expense of using an 8^3 sample support over which the convolutions are performed. Large filter supports are costly, especially in three dimensions, because of poor memory locality. Finding filters which give good results with reasonable supports is an ongoing research problem.

Seeking to provide practical advice on a filter set for efficient curvature measurement, we performed a search among the filters (created by the framework above) with 4^3 sample support. Allowing up to 3rd order continuity (C^3) and 4th order accuracy (4-EF), a total of 200 filter combinations were assessed by rendering the Marschner-Lobb dataset. These renderings (not shown) used an orthonormal projection from directly above the dataset, with shading turned off, to generate grayscale images of mean curvature. Based on the difference between the ideal and rendered curvatures, an “optimal” 4^3 filter set was found; see Figure 3(c). An interesting property of this filter set is that for the derivatives, high accuracy is more important than high continuity. For example, there are 4-support 2nd derivative filters with higher continuity than C^0 , but they are 1-EF, not 2-EF.

Figure 3(d) uses another set of filters, also created by the framework above, but better known through the BC-splines of Mitchell and Netravali [1988]. Interpolation is done by the Catmull-Rom filter (B.C = 0.0.5), which is the only 3-EF (the most accurate) interpolating BC-spline. The first and second derivatives are measured with the first and second derivatives of the cubic B-spline (B.C = 1.0), which is the only C^2 (the most smooth) BC-spline. In our quantitative comparison of 200 filter sets, these did nearly as well as the “optimal” 4^3 filters. Given that *any* 4^3 sample filter set will be quite limited in the feature detail it can reconstruct and measure, we recommend that when practical considerations dictate a 4^3 support, the filters listed in Figure 3(d) are sufficient; only slightly better results will come from filters with higher polynomial degree.

When larger filter supports can be employed, higher accuracy is more important than higher continuity. In most practical applications, the underlying data can be sufficiently modeled with low degree polynomials. 3-EF or 4-EF filters should be sufficient; those in Figure 3(b) are excellent. When measuring derivatives in noisy data, smoothing often improves accuracy. Such smoothing can be a pre-process. Or, it can be folded into the convolution, by slightly blurring values, instead of interpolating them. For instance, instead of using the Catmull-Rom filter as in Figure 3(d), we can use the B-spline. We have found that the resulting loss of accuracy on very small features is compensated for by the reduction of spurious curvature variations due to noise. For the remainder of this paper, we use the B-spline and its derivatives for our curvature measurements.

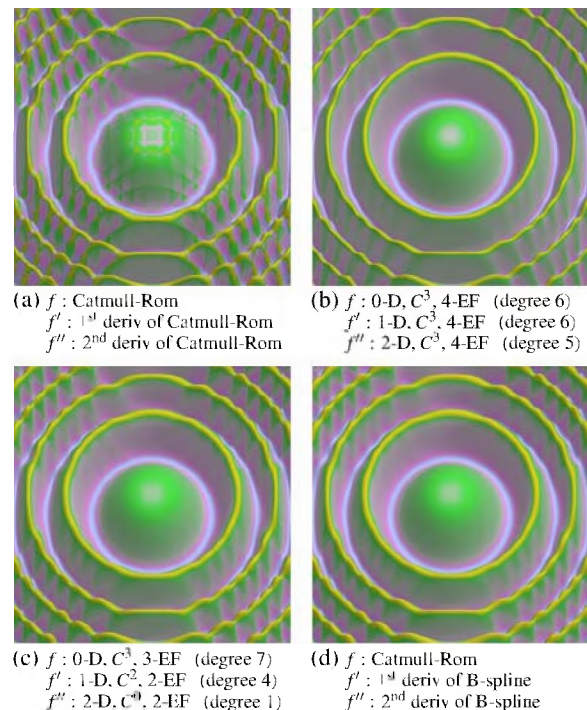


Figure 3: Marschner-Lobb dataset with various filtered reconstructions. Below each image is a list of the three filters used to reconstruct the data value and the first and second derivatives. The polynomial degrees are listed for (b) and (c). In (a) and (d) the polynomial degrees are 3, 2, and 1.

5 Non-Photorealistic Volume Rendering

5.1 Controlling Contour Thickness

Contours (sometimes referred to as *silhouettes*) are a basic part of all current approaches to non-photorealistic volume rendering. By emphasizing the transition between front-facing and back-facing surface locations, contours delineate object shape and clarify sites of occlusion. Contours of *polygonal* surfaces are drawn on edges between faces with different visibilities, as determined by a sign change in $\mathbf{v} \cdot \mathbf{n}$, the dot product between the view vector \mathbf{v} and the face normal \mathbf{n} [Gooch and Gooch 2001]. Lacking polygonal discretization, volume rendered contours are generally produced with a univariate function of $\mathbf{v} \cdot \mathbf{n}$. For instance, the surface color is darkened when $\mathbf{v} \cdot \mathbf{n}$ is within some user-determined range around zero.



Figure 4: Contours based solely on $\mathbf{v} \cdot \mathbf{n}$

An unfortunate consequence of this approach is uncontrolled variation in the apparent contour thickness. As shown in Figure 4, where the surface is nearly flat, a large region of surface normals is nearly perpendicular to the view vector, making the contours too thick. Conversely, in fine structures, where the emphasis provided by contours could be especially helpful, they can be too thin.

We propose regulating the thickness of volume rendered contours by using curvature. Specifically, the normal curvature along the view direction, κ_v , restricts the range of surface normal orientations rendered as contour. We find κ_v in terms of the tangent projection \mathbf{P} and the geometry tensor \mathbf{G} , exploiting the fact that restricted to the tangent plane, \mathbf{G} is exactly the Weingarten map [Do Carmo 1976]:

$$\kappa_v = \frac{(\mathbf{P}\mathbf{v})^T}{|\mathbf{P}\mathbf{v}|} \mathbf{G} \frac{\mathbf{P}\mathbf{v}}{|\mathbf{P}\mathbf{v}|} = \frac{\mathbf{v}^T \mathbf{G} \mathbf{v}}{|\mathbf{P}\mathbf{v}|^2} = \frac{\mathbf{v}^T \mathbf{G} \mathbf{v}}{\mathbf{v}^T \mathbf{P} \mathbf{v}}.$$

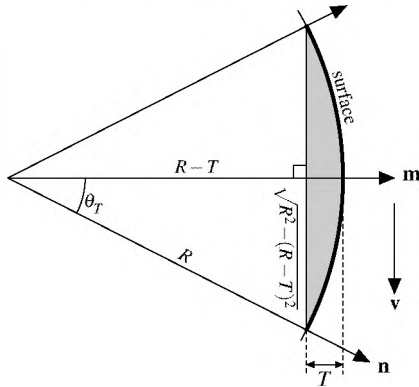


Figure 5: Creating contour thickness T based on surface normal \mathbf{n} , view vector \mathbf{v} , and radius of surface curvature $R = 1/\kappa_v$.

Contours with approximate image space thickness T are rendered with a two-dimensional transfer function of κ_v and $\mathbf{v} \cdot \mathbf{n}$. Our method is an approximation because κ_v is assumed to be constant over the contour extent. Figure 5 illustrates a slice through a surface, on the plane spanned by the surface normal and the view vector as it grazes the surface. Within this plane, \mathbf{m} is the single surface normal orthogonal to \mathbf{v} . A certain range of surface normals around \mathbf{m} must be rendered as contour so that, when projected on the image plane, the contour region appears to have thickness T . This range is a function of T and of R , the radius of curvature at this point along the contour. Letting θ_T denote the maximum angle between \mathbf{m} and a surface normal within the contour:

$$\begin{aligned} |\mathbf{m} \cdot \mathbf{n}| &\geq \cos(\theta_T) \Rightarrow |\mathbf{v} \cdot \mathbf{n}| \leq \sin(\theta_T) \\ \Rightarrow |\mathbf{v} \cdot \mathbf{n}| &\leq \frac{\sqrt{R^2 - (R-T)^2}}{R} = \sqrt{\frac{2RT - T^2}{R^2}} = \sqrt{\frac{T}{R} \left(2 - \frac{T}{R}\right)} \\ \Rightarrow |\mathbf{v} \cdot \mathbf{n}| &\leq \sqrt{T\kappa_v(2 - T\kappa_v)} \end{aligned} \quad (5)$$

Figure 6(a) graphs the inequality in Equation 5; it is simply a portion of a circle. When curvature κ_v is small, a narrow range of surface orientations should contribute to the contour, while a larger range of orientations must be colored to create the contour on high curvature features. The lookup table in Figure 6(b) is one way to implement the contour transfer function. High values of κ_v are clamped to $1/T$ to ensure that very small features are not missed. The transition between dark and light in the contour transfer function may be blurred somewhat to soften the contour edges, and to avoid accentuating minor surface variations near the contour.

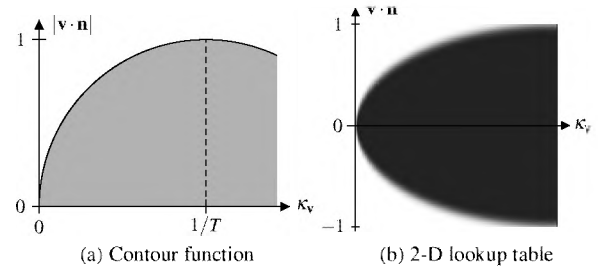


Figure 6: Graph of contour function, and the lookup table used to implement thickness-controlled contours in practice.



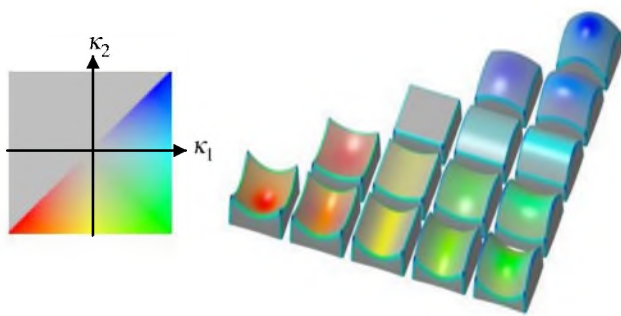
Figure 7: Thickness-controlled contours, using $T = 1$ (left) and $T = 2.5$ (right).

Figure 7 demonstrates thickness-controlled contours with two different values for the thickness parameter T . The contour thickness around the ears and back of the bunny are now equal, and the surface details on the lower body are more clearly depicted, in contrast to Figure 4. Also, varying T succeeds in adjusting the image space thickness of the contours. Since the dimensions of T are voxels, setting $T = 1$ means that contours should be about as thick as the size of one voxel projected onto the image plane. As with previous methods for volume rendered contours, it is possible for our thickness-controlled contours to be drawn at locations where there is actually no change in surface visibility. Also, contours may be drawn improperly in regions where curvature is too low to be measured accurately. We have not found either of these problems to be serious impediments to the illustrative purpose of contours.

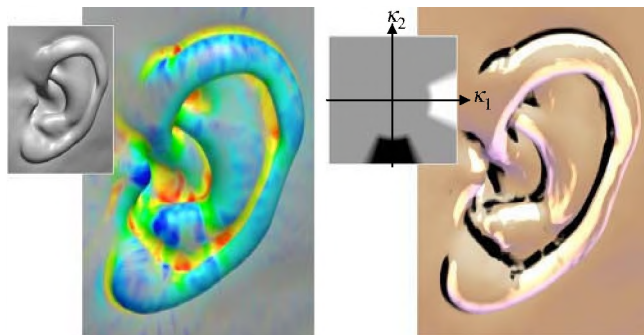
5.2 Emphasizing Valleys and Ridges

One ingredient missing in the current vocabulary of non-photorealistic volume rendering techniques is the use of curvature magnitudes to enhance depiction of surface details. Conveying overall surface *shape* is effectively done by using textures aligned with curvature directions [Interrante et al. 1997; Interrante 1997; Girshick et al. 2000; Lum et al. 2003], but we wish to avoid the overhead of pre-computing and storing the necessary volumetric texture. Though visually similar to depiction of ridge and valley lines, our approach is simpler than previous volumetric methods in that we do not use third derivatives [Monga et al. 1994], nor do we perform a local search to determine curvature extrema [Interrante et al. 1995]. Rather, we work in the two-dimensional space of principal curvatures (κ_1, κ_2) , similar to the approach of Hladůvka et al. [2000], with the important difference that the geometry tensor can distinguish between convex and concave surface geometry.

Figure 8 demonstrates transfer functions of the (κ_1, κ_2) principal curvatures. Figure 8(a) portrays curvature space with a sequence of quadratic surface patches (each a 20^3 volume) rendered with the transfer function shown. Figure 8(b) shows how curvature transfer functions can emphasize surface variations not otherwise read-



(a) Volume rendered diagram of (κ_1, κ_2) space. The colors in the (κ_1, κ_2) transfer function domain are mapped onto the patches with corresponding surface curvature.



(b) Left: Visualization of ear curvature using transfer function from (a); Right: ridge and valley emphasis implemented with inset transfer function, combined with Gooch shading

Figure 8: Principal curvature transfer functions

ily seen. Following a technical illustration convention of darkening valleys and lightening ridges [Gooch et al. 1999], we darken or lighten volume rendered samples according to their location in (κ_1, κ_2) space, using the transfer function shown in Figure 8(b). In addition to the curvature-based transfer function, Gooch shading is used to convey overall shape [Gooch et al. 1998].

Figure 9 highlights the complementary nature of our two curvature-based non-photorealistic volume rendering effects. Thickness-controlled contours delineate the shapes of large structures, while ridges and valleys highlight fine surface details. In these renderings, surface color is assigned as a function of three vectors (\mathbf{v} , \mathbf{n} , \mathbf{l} : view, surface normal, light) and three different kinds of curvature (κ_v , κ_1 , κ_2 : view direction, principal curvatures). We follow the example of previous work demonstrating the benefit of expressing multi-dimensional transfer functions as the separable product of simpler functions [Kniss et al. 2002]:

$$\text{RGB}(\mathbf{v}, \mathbf{n}, \mathbf{l}, \kappa_v, \kappa_1, \kappa_2) = g(\mathbf{n} \cdot \mathbf{l}) c(\kappa_v, \mathbf{n} \cdot \mathbf{v}) e(\kappa_1, \kappa_2).$$

Gooch shading comes from $g(\mathbf{n} \cdot \mathbf{l})$; $c(\kappa_v, \mathbf{n} \cdot \mathbf{v})$ creates thickness-controlled contours; $e(\kappa_1, \kappa_2)$ provides ridge and valley emphasis. The contour thickness T is the only parameter to tune in $c(\kappa_v, \mathbf{n} \cdot \mathbf{v})$. The ridge and valley emphasis does require subjective adjustments, but the intuitive nature of (κ_1, κ_2) space makes this relatively simple. By far the most challenging parameter setting remains the opacity assignment, for which we use a two-dimensional transfer function of data value and gradient magnitude, as guided by previous work [Kindlmann and Durkin 1998; Kniss et al. 2002].

6 Visualizing Surface Smoothing

Medical imaging and three-dimensional range sensing technologies produce high resolution descriptions of object surfaces, but they often suffer from noise. Level set methods for denoising the data have become popular because of their inherent ability to handle complex topology [Osher and Sethian 1988]. Level sets can be evolved by minimizing an energy function based on principal curvatures, the lowest-order differential invariants of surface shape [Tasdizen et al. 2002]. Minimizing the surface integral of *total curvature* $\kappa_T = \sqrt{\kappa_1^2 + \kappa_2^2}$ gives an isotropic surface smoothing, analogous to Gaussian image blurring. A variant of this energy function can be used to preserve prominent features on the surface while eliminating noise [Tasdizen et al. 2002], generalizing anisotropic image diffusion [Perona and Malik 1990] to surface processing.

We propose that curvature-based direct volume rendering is a natural and effective tool for visualizing curvatures and surfaces as they evolve during a level set solution. Level sets and direct volume rendering share an implicit surface representation on a common underlying grid. Because curvature-based volume rendering requires no pre-processing, it permits direct inspection of intermediate smoothing results, without an intervening meshing step. To visualize the spatial and temporal structure of total curvature on the evolving surface, we use two one-dimensional transfer functions. One assigns surface color as a function of total curvature; the other assigns opacity as a function of data value.

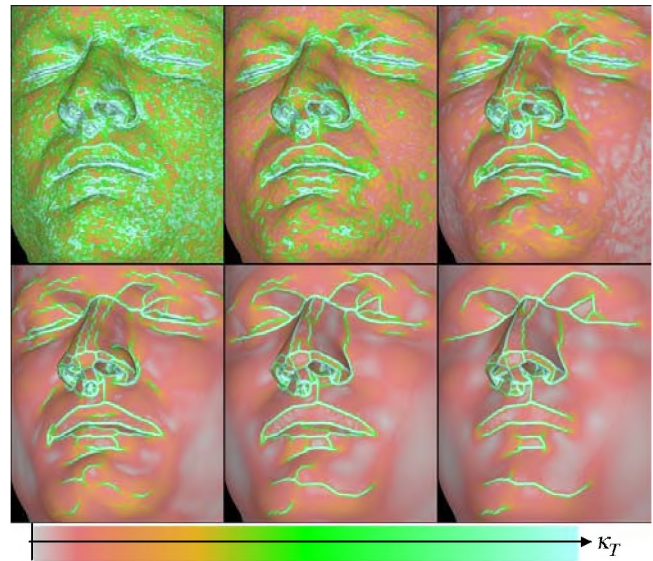


Figure 10: Six snapshots of anisotropic surface smoothing

Figure 10 shows volume renderings of six instances during the anisotropic smoothing of an isosurface from MRI data. All rendering parameters are fixed; only the volume dataset is changing. Noise in the original surface (top-left picture) is seen as randomly distributed high curvature points on the surface. As the surface evolves, most of the high curvature points fade away, while the rest coalesce into prominent creases. This is predicted by mathematical analysis of the anisotropic diffusion equations [Perona and Malik 1990]. Our visualization experiment verifies this property, as well as illuminating other features of the process. For instance, as the surface smoothly evolves, the network of creases undergoes discrete changes in connectivity. An interesting research topic suggested by these visualizations is steering the energy function in order to create or maintain a particular topology in the crease network.

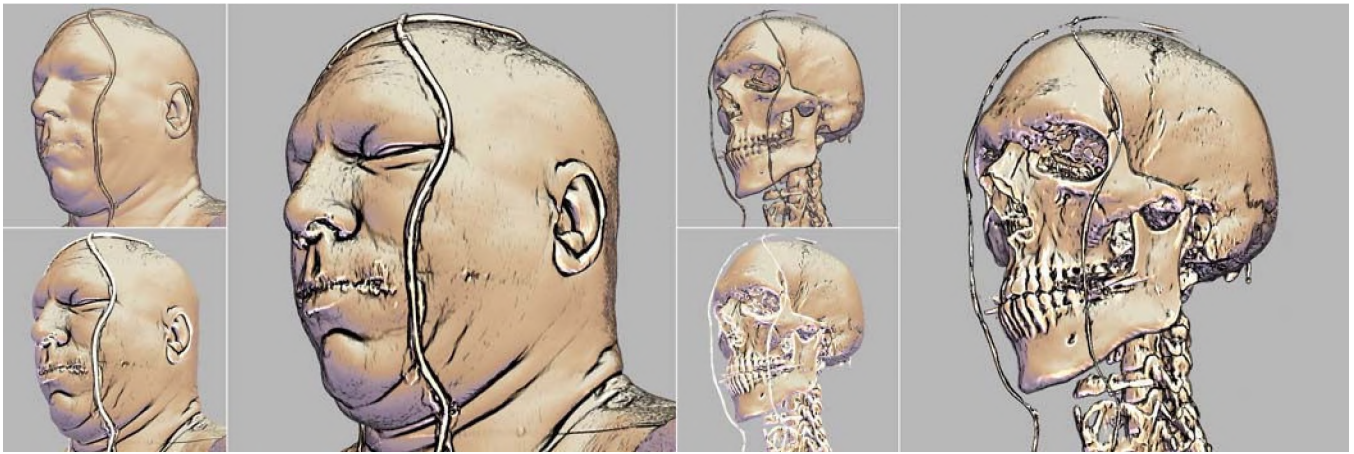


Figure 9: Curvature-based non-photorealistic volume rendering. The two small images depict the contribution of thickness-controlled contours (top) and ridge and valley emphasis (bottom). The large images show the combination of the two effects.

7 Visualizing Isosurface Uncertainty

If a grayscale image is thought of as a height field on which a fluid is flowing downward, *flowline curvature* is the curvature of the path of the flow, as viewed from above [ter Haar Romeny et al. 1991]. In three dimensions, flowline curvature κ_f characterizes the degree to which an isosurface changes its orientation as a function of small changes in isovalue. With zero flowline curvature, adjacent isosurfaces are parallel. Like the principal curvatures, flowline curvature is a differential invariant. However, instead of describing change within a single isosurface, it describes change between isosurfaces. It can be computed from $\nabla \mathbf{n}^T$ (Equation 2):

$$\kappa_f = |\nabla \mathbf{n}^T (\mathbf{nn}^T)|_F = \left\| \begin{bmatrix} \kappa_1 & 0 & \sigma_1 \\ 0 & \kappa_2 & \sigma_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\|_F$$

$$= \sqrt{\sigma_1^2 + \sigma_2^2}.$$

To our knowledge, flowline curvature has not been used in volume visualization. We propose that flowline curvature can be used to visualize the uncertainty of material boundaries as modeled by isosurface extraction. The physical shape of a boundary between two materials is a fixed and intrinsic property of the object being sampled in the volume dataset. At locations where small changes in isovalue produce large changes in isosurface orientation, the isosurface is probably a poor model of the material boundary. On the other hand, where the shape of an isosurface is robust against small isovalue changes, we can be more certain of its geometric accuracy. By colormapping flowline curvature onto volume rendered surfaces, we can qualitatively indicate uncertainty of surface shape. Regions of especially high flowline curvature are visually flagged as suspect.

One common material boundary that can be hard to extract via isosurfacing is the surface of bone in CT data. Figure 11 shows six pseudo-isosurface volume renderings of a thumb. Each is rendered with a univariate opacity function changing from transparent to fully opaque over a narrow range of data values, effectively creating a threshold at the specified isovalue. Flowline curvature is colormapped onto the surfaces through a single transfer function common to all the renderings. The (anatomically erroneous) holes on the bone surface are consistently lined by high flowline curvature, and the sites of hole formation can largely be predicted by flowline curvature patterns at lower isovalues. There are, however, locations with high *surface* curvature and low *flowline* curvature,

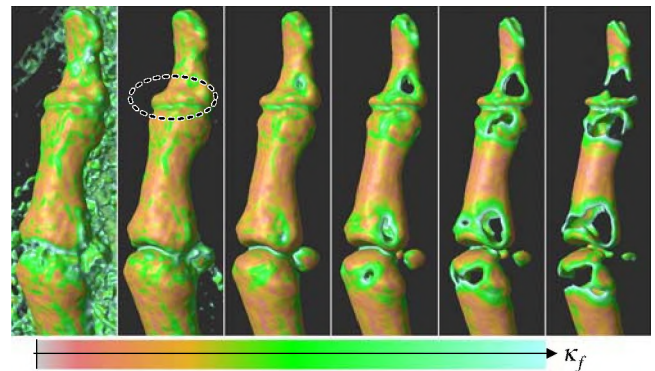


Figure 11: CT isosurfaces at values 1055, 1095, 1175, 1255, 1335, and 1405, colormapped by flowline curvature. Circle indicates site of high surface curvature and low flowline curvature.

such as the (circled) lower edge of the top bone in the second image. Because there is no single isovalue for which the resulting isosurface has consistently low flowline curvature, these visualizations illustrate how hard it is to find a CT isovalue which represents the bone surface with high certainty.

8 Discussion, Future Work

We have demonstrated that curvature plays a valuable role in creating informative volume visualizations. Our method conforms to a standard post-classification volume rendering pipeline: the color and opacity assignments at each raycast sample are generated from transfer functions, on the basis of locally computed quantities, such as gradient and curvature. Rheingans and Ebert [2001] have described how this approach engenders the difficult task of transfer function creation and modification. However, as noted in Section 5.2, we seek to control this complexity by enforcing separability of the transfer function into simpler, intuitive components. The conceptual orthogonality of the various NPR effects maps to orthogonality between the corresponding transfer function domains.

Biological surfaces tend to be somewhat noisy, and as a result,

where the surface curvature is very low, as on the scalp or the top of the skull, it is hard to measure curvature accurately. Adaptive methods which measure derivatives over a larger area in such circumstances could help, as could proper application of scale-space methods [ter Haar Romeny et al. 1991]. As mentioned in Section 4, more analysis is needed to determine the precise relationship between the characteristics of the convolution filters, and the quality of curvature measurements produced. Finally, our proof-of-concept raycaster does not exploit graphics hardware, and is far from interactive. Previous work on hardware-based convolution [Hadwiger et al. 2001] and hardware-based NPR [Lum and Ma 2002; Nagy et al. 2002], however, suggests the exciting possibility of high-quality curvature measurements at interactive rates.

9 Acknowledgments

We thank the anonymous reviewers for constructive comments. The bunny CT scan in Figures 4 and 7 is courtesy of the Stanford Volume Data Archive. The CT data in Figures 8(b), 9, and 11 is courtesy of the Visible Human Project. Funding for this research includes NIH grants 5P20HL68566-02 and 2P41RR12553-04.

Every rendered image in this paper can be regenerated exactly with open-source software and public datasets; see <<http://www.sci.utah.edu/~gk/vis03>>.

References

- BELYAEV, A., PASKO, A., AND KUNII, T. 1998. Ridges and ravines on implicit surfaces. In *Computer Graphics International '98*, 530–535.
- CSÉBFAI, B., MROZ, L., HAUSER, H., KÖNIG, A., AND GRÖLLER, E. 2001. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum* 20, 3, C452–C460.
- DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- GIRSHICK, A., INTERRANTE, V., HAKER, S., AND LEMOINE, T. 2000. Line direction matters: An argument for the use of principal directions in 3D line drawing. In *First International Symposium on Non-Photorealistic Animation and Rendering*, 43–52.
- GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, MA.
- GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings SIGGRAPH '98*, 447–452.
- GOOCH, B., SLOAN, P.-P., GOOCH, A., SHIRLEY, P., AND RIESENFELD, R. 1999. Interactive technical illustration. In *1999 ACM Symposium on Interactive 3D Graphics*, 31–38.
- HADWIGER, M., THEUSSL, T., HAUSER, H., AND GRÖLLER, E. 2001. Hardware-accelerated high-quality filtering on pc graphics hardware. In *Proceedings Vision, Modeling, and Visualization 2001*.
- HLADŮVKA, J., KÖNIG, A., AND GRÖLLER, E. 2000. Curvature-based transfer functions for direct volume rendering. In *Spring Conference on Computer Graphics 2000*, vol. 16, 58–65.
- HOFFMAN, K., AND KUNZE, R. 1971. *Linear Algebra*. Prentice-Hall, Inc., Englewood Cliffs, NJ, ch. 10.2.
- INTERRANTE, V., FUCHS, H., AND PIZER, S. 1995. Enhancing transparent skin surfaces with ridge and valley line. In *Proceedings IEEE Visualization 1995*, 52–59.
- INTERRANTE, V., FUCHS, H., AND PIZER, S. M. 1997. Conveying the 3D shape of smoothly curving transparent surfaces via texture. *IEEE Transactions on Visualization and Computer Graphics* 3, 2 (April-June), 98–117.
- INTERRANTE, V. 1997. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. In *Proceedings SIGGRAPH '97*, vol. 31, 109–116.
- KINDLMANN, G., AND DURKIN, J. 1998. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium On Volume Visualization*, 79–86.
- KNISS, J., KINDLMANN, G., AND HANSEN, C. 2002. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 8, 3 (July-Sep), 270–285.
- LEVOY, M. 1988. Display of surfaces from volume data. *IEEE Computer Graphics & Applications* 8, 5, 29–37.
- LU, A., MORRIS, C., EBERT, D., RHEINGANS, P., AND HANSEN, C. 2002. Non-photorealistic volume rendering using stippling techniques. In *Proceedings IEEE Visualization 2002*, 211–218.
- LUM, E., AND MA, K.-L. 2002. Hardware-accelerated parallel non-photorealistic volume rendering. In *International Symposium on Non-Photorealistic Animation and Rendering*, 67–74.
- LUM, E. B., STOMPEL, A., AND MA, K.-L. 2003. Using motion to illustrate static 3D shape – kinetic visualization. *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (April-June), 115–126.
- MA, K.-L., HERTZMANN, A., INTERRANTE, V., AND LUM, E. B. 2002. Course 23: Recent advances in non-photorealistic rendering for art and visualization. In *Course Notes for SIGGRAPH 2002*.
- MARSCHNER, S., AND LOBB, R. 1994. An evaluation of reconstruction filters for volume rendering. In *Proceedings of Visualization '94*, R. D. Bergeron and A. E. Kaufman, Eds., 100–107.
- MARSDEN, J., AND TROMBA, A. 1996. *Vector Calculus*. W.H. Freeman and Company, New York, ch. 2.6, 4.2.
- MITCHELL, D., AND HANRAHAN, P. 1992. Illumination from curved reflectors. In *Proceedings SIGGRAPH '92*, 283–291.
- MITCHELL, D., AND NETRAVALI, A. 1988. Reconstruction filters in computer graphics. In *Proceedings SIGGRAPH '88*, 221–228.
- MÖLLER, T., MÜLLER, K., KURZION, Y., MACHIRAJU, R., AND YAGEL, R. 1998. Design of accurate and smooth filters for function and derivative reconstruction. In *IEEE Symposium on Volume Visualization*, 143–151.
- MÖLLER, T. 1999. *Spatial Domain Filter Design*. PhD thesis, Ohio State University.
- MONGA, O., BENAYOUN, S., AND FAUGERAS, O. 1992. From partial derivatives of 3D density images to ridge lines. In *IEEE Conference on Vision and Pattern Recognition (CVPR)*.
- MONGA, O., LENGAGNE, R., AND DERICHE, R. 1994. Crest lines extraction in volume 3D medical images: a multi-scale approach. Rapport de Recherche 2338, INRIA, July.
- NAGY, Z., SCHNEIDER, J., AND WESTERMANN, R. 2002. Interactive volume illustration. In *Proceedings Vision, Modeling, and Visualization 2002*.
- OSHER, S., AND SETHIAN, J. 1988. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation. *J. Computational Physics* 79, 12–49.
- PERONA, P., AND MALIK, J. 1990. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12, 7 (July), 629–639.
- PREUSSER, T., AND RUMPF, M. 2002. A level set method for anisotropic geometric diffusion in 3D image processing. *SIAM J. Appl. Math.* 62, 5, 1772–1793.
- RHEINGANS, P., AND EBERT, D. 2001. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics* 7, 3 (July-Sep), 253–264.
- TASDIZEN, T., WHITAKER, R., BURCHARD, P., AND OSHER, S. 2002. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings IEEE Visualization 2002*, 125–132.
- TER HAAR ROMENY, B., FLORACK, L., KOENDERINK, J., AND VIERGEVER, M. 1991. *Lecture Notes in Computer Science*, vol. 511. Springer-Verlag, ch. Scale-space: Its natural operators and differential invariants, in Information Processing in Medical Imaging, 239–255.
- TREAVETT, S., AND CHEN, M. 2000. Pen-and-ink rendering in volume visualization. In *Proceedings IEEE Visualization 2000*, 203–210.