

Using Distance Maps for Accurate Surface Representation in Sampled Volumes

Sarah F. Frisken Gibson

TR99-25 December 1999

Abstract

High quality rendering and physics-based modeling in volume graphics have been limited because intensity-based volumetric data do not represent surfaces well. High spatial frequencies due to abrupt intensity changes at object surfaces result in jagged or terraced surfaces in rendered images. The use of a distance-to-closest-surface function to encode object surfaces is proposed. This function varies smoothly across surfaces and hence can be accurately reconstructed from sampled data. The zero-value iso-surface of the distance map yields the object surface and the derivative of the distance map yields the surface normal. Examples of rendered images are presented along with a new method for calculating distance maps from sampled binary data.

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Using Distance Maps for Accurate Surface Representation in Sampled Volumes

Sarah F. F. Gibson

MERL – A Mitsubishi Electric Research Laboratory
201 Broadway, Cambridge, MA, 02139

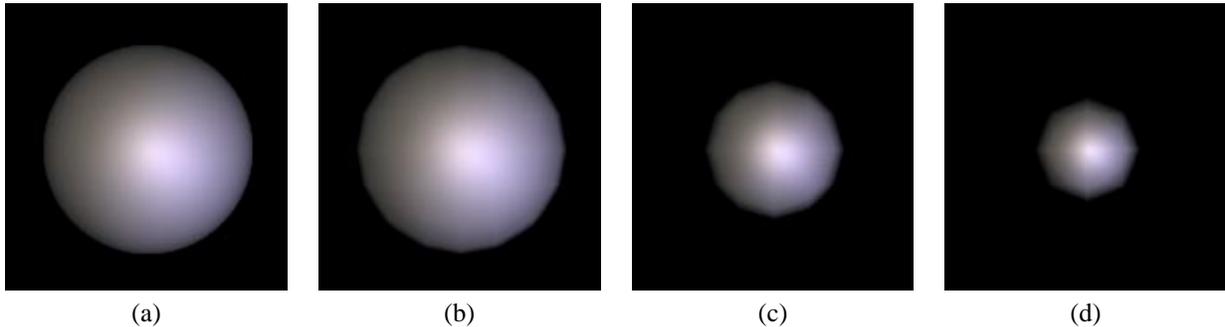


Figure 1: *Shaded, volume rendered spheres stored with two values per voxel: a value indicating the distance to the closest surface point; and a binary intensity value. The sphere in a) has radius 30 voxels and is stored in an array of size 64^3 . The spheres in b), c), and d) have radii 3 voxels, 2 voxels and 1.5 voxels respectively and are stored in arrays of size 10^3 . The surface normal used in surface shading was calculated using a 6-point central difference operator on the distance values. Remarkably smooth shading can be achieved for these low resolution data volumes because the function of the distance-to-closest surface varies smoothly across surfaces. (See color plate.)*

Abstract

High quality rendering and physics-based modeling in volume graphics have been limited because intensity-based volumetric data do not represent surfaces well. High spatial frequencies due to abrupt intensity changes at object surfaces result in jagged or terraced surfaces in rendered images. The use of a distance-to-closest-surface function to encode object surfaces is proposed. This function varies smoothly across surfaces and hence can be accurately reconstructed from sampled data. The zero-value iso-surface of the distance map yields the object surface and the derivative of the distance map yields the surface normal. Examples of rendered images are presented along with a new method for calculating distance maps from sampled binary data.

Keywords: Volume Rendering, Volume Graphics, Surgical Simulation, Medical Applications

1 Introduction

Objects in volume graphics are represented by discrete arrays of sampled data. Unlike surface-based object representations, volumetric data can embody interior structure. For this reason, volumetric models have advantages over surface models for visualization and physics-based modeling of complex objects. For example, a volumetric representation allows the visualization of interior anatomical structure from medical image data or the modeling of 3D object deformation using mass-spring models or finite element methods. In addition, while operations such as object cutting, slicing, or tearing are challenging for surface-based models, they can be performed relatively easily with a volumetric representation [5]. In applications such as surgical simulation, tissue complexity and the need to be able to model volume deformation and object cutting

and tearing make volumetric models very attractive.

While significant advances have been made both in the acceptance of volume graphics and in the sophistication of volume rendering and object manipulation algorithms, two important issues have limited its use: 1) the large number of elements in a volumetric object is challenging for data storage, image rendering, and physical modeling; and 2) due to the nature of discrete sampling, volumetric data sets do not represent arbitrarily oriented surfaces well. While hardware and software developments have addressed the first point (e.g. [15, 8, 24, 23]), the inability to represent surfaces remains an important limitation of volume graphics.

In computer graphics, an accurate representation of surfaces is important for a number of reasons. First, the use of a lighting and shading model for high quality rendering requires knowledge of surface location and orientation. Second, in physics-based graphics or in haptic interactions with graphical models, calculating impact forces requires the positions and surface normals of contact points. Errors in contact position and surface normals create artifacts in rendering and inconsistencies in physical modeling. While there is no explicit representation of surfaces in volumetric data, there may be prior knowledge of surfaces. For example, if the data is created from an analytic model or modified by applying a cutting plane to the volume, then a mathematical description of the object's surface may exist. If the data comes from a medical scan of human anatomy, then knowledge about the existence and smoothness of object surfaces could be exploited.

In his edited book, Kaufman states that: "volume visualization still lacks a simple yet general shading paradigm that provides high-fidelity shading for natural appearance and optimal recognition" [11, p. 171]. Shading is difficult in volume graphics because surfaces are not well represented in intensity-based sampled data. Most volumetric models are intensity-based: the image intensity varies relatively slowly over object interiors but changes abruptly

at object surfaces. During rendering, these abrupt changes are detected and used to determine the strength and direction of surfaces in the data. However, because an abrupt change in intensity represents high spatial frequencies in the data, high sampling rates are required to reconstruct the image and its gradient near surfaces¹. Existing approaches to shading in volume rendering assume that the volume is band limited and that data is sampled above the Nyquist rate. When this is not true, (for example in binary sampled data), rendered images have characteristic terraced or jagged surfaces. This image aliasing can be reduced by low-pass filtering the sampled data (e.g. [30]) but such filters blur surfaces indiscriminately, eliminating detail that may be important.

In this paper, surfaces are represented with a function that varies smoothly across surfaces. Low spatial frequencies of such a function mean that surfaces in sampled data can be accurately reconstructed at relatively low sampling rates. In particular, this paper explores the use of a signed distance-to-closest-surface mapping to encode object surfaces in sampled volumes. The proposed distance map has several important properties illustrated in Figure 2. First, the gradient of the distance map yields the direction of the surface normal. Second, the zero-value iso-surface of the distance map is the object surface. Third, unlike intensity-based object representations, the distance map varies smoothly across object surfaces, avoiding high spatial frequencies at object boundaries. In fact, Section 4 shows that the distance map varies linearly near the surface when the curvature of the surface is small relative to the sampling distance. This linearity allows us to use a simple trilinear interpolation function and a 6-point central difference filter to accurately reconstruct the distance map and its derivative near surfaces [20]. Hence, the distance map provides a way to encode object surfaces into sampled, volumetric data so that they can be accurately reconstructed with relatively low-cost reconstruction filters.

This paper is organized as follows: Section 2 presents some background and related work. Section 3 presents shaded, volume rendered images to illustrate the effectiveness of the distance map method for encoding object surfaces. Section 4 presents a mathematical analysis of the distance-to-closest-surface function that can be used to determine an appropriate sampling rate for an object of known curvature. Section 5 presents a new way to calculate distance maps from binary data. In Section 6 future work related to using distance maps in rendering, haptics and physics-based modeling is discussed. Finally, Section 7 summarizes the importance and advantages of the distance map approach.

2 Background

Volumetric objects can be generated from measured images, simulated data, or by discretizing analytic functions defining objects or surfaces. The volumetric data may consist of binary values, grey-scale values, or a more complex data structure with additional information about visual or material properties of the sampled object [5]. When rendering these objects, shading provides visual cues about object shape and texture and can have a strong effect on the visual realism of a rendered scene. There are a number of image-space and object-space shading methods that have been used for shading volumetric objects [12]. In this paper, we assume the use of a shading model, such as the Phong illumination model [4], which requires an estimate of the object surface normal for shading calculations.

This paper focuses on volumetric data that originates as analytic functions, polygonal models, or binary samples. In the first two cases, prior knowledge of surfaces exist when data is encoded into the volume. In the second case, exact knowledge of the surface has

¹Nyquist’s sampling theorem states that a continuous function can only be reconstructed accurately if it is sampled at a rate of twice the highest frequency component of the function (e.g. [22]).

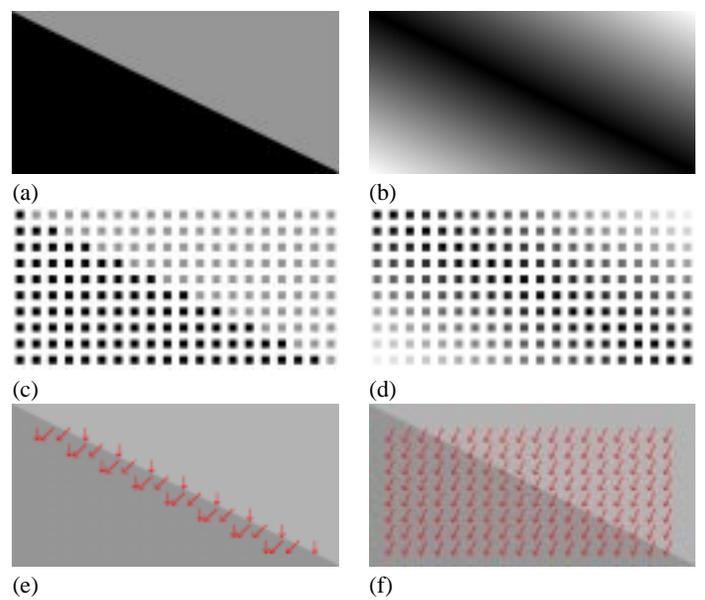


Figure 2: *a) A binary image of a grey object on a black background. b) The magnitude of the (signed) distance map for the binary image of a), where black corresponds to zero distance from the object edge. c) The sampled binary image. d) The sampled distance map. e) Gradient vectors calculated using central differences at sample points of the sampled binary image (the background shows the true shape of the binary image). f) Gradient vectors calculated using central differences at sample points of the sampled distance map. The sampled distance map provides a smoother and more accurate estimate of the edge normal than the sampled binary image does. The sampled distance map also provides an estimate of the edge normal farther from the object edge which could be useful in physics-based modeling. The zero value of the distance map provides the location of the surface.*

been discarded. Shading from grey-scale data can produce good quality images when the data is band limited and appropriate filters are used (e.g. [9]) so that the use of a distance map for surface normal calculation is not necessary. However, as discussed in Section 5, there are times when a binary segmentation of the grey-scale data is preferable for surface shading.

Some existing methods store surface normals of known analytic models [32] or polygonal models [17] at each data point. These sampled surface normals are used to reconstruct normals for samples between data elements during rendering. However, this method requires the storage of 3 floating point values per data element or quantization of the surface normals, which can result in quantization artifacts. In addition, the stored surface normals do not improve reconstruction of the surface location. Yagel et al. describe a system where, for volumetric objects generated from analytic surfaces, data elements contain pointers to a more complex description of the underlying surface that can be used for accurate calculation of surface intersections and normals. However, the complexity of this method greatly increases rendering times.

Binary data is generated by discretizing analytic or geometric object representations or by segmenting measured data. Binary data consists of one value for points inside the object and a second value for points outside of the object. Two issues make the estimation of surface normals from binary data especially difficult: 1) by discretizing the data or segmenting the volume, any prior information about an object surface has been discarded and the best that can be done is to estimate a new surface from the binary samples; and 2)

because of the binary nature of the data, there is always an abrupt change in data values at object surfaces, resulting in reconstruction artifacts due to undersampling. There are a number of methods that have been used to shade binary volumes (see reviews in [12, 33]). These include shading the six faces of each non-empty element, producing a blocky appearance, and various methods using look-up tables [17], smoothing filters, and surface estimation filters [28] which approximate the surface normal from the state of local neighbors. One approach that has produced good quality images is to blur the binary image by filtering with a low pass filter [30, 31, 1, 21]. This method creates a grey-scale image from the binary data where surface normals are well represented by the gradient of the smoothed image. When the filter value decreases monotonically with distance from the filter center, this method is closely-related to the distance-based method presented here. However, the application of a low-pass filter blurs the data indiscriminately and removes fine detail that can be critical in medical applications.

There have been a number of studies of filters for image reconstruction and gradient estimation. Most of these are based on frequency analysis of the filters [3, 18, 2] and all of them apply to grey-scale data. Moller et al. [20] analyzed interpolation and gradient estimation filters in the spatial domain. By expressing the filtered image intensity using Taylor series expansion, they found that estimation errors can be expressed as a function of the filter coefficients, the sampling rate, and high order derivatives of the image function at the sampled point. In particular, they show that trilinear interpolation and a central difference gradient operator will exactly reconstruct the image intensity and gradient, respectively, for a linear image function. This result is used in Section 4.

There are other applications in rendering, modeling, image processing, and robotics that have used discrete distance maps. Distance maps have been used to accelerate volume rendering by reducing the sampling of empty space [35, 27]. Schroeder et al. [26] used distance maps to generate a swept volume for a moving polygonal model and then constructed a polygonal surface model of the swept volume using Marching Cubes [16]. In robotics, discrete distance maps are used for path planning by generating potential fields around obstacles that decrease with the square of the distance from the obstacles (e.g. [14, 10]). Yagel et al. [34] use discrete distance maps for visualizing thick regions of models that may be problematic in die casting. In image processing, a technique known as watershed segmentation uses a map of distances from edges or features to help separate adjoining features (e.g. [25]). This paper is distinct from these approaches. It uses distance maps to explicitly encode surfaces into sampled volumetric data for high quality shading in volume rendering.

3 Volume Rendering from Distance Maps

To create a distance map, we require a model of the object surface. When the data originates as an analytic function or a polygonal model, the surface is known a priori. If the data originates as a binary volume, a surface model must be estimated from the binary data. Given the surface model, each sample in the distance map is assigned the distance to the closest surface point on the model. It is assumed that objects are far enough apart so that there is no interference in the distance map close to object surfaces, or that individual objects are stored in unique volumes. In addition to sampled distance values which are used for surface normal calculation in volume rendering, volume elements may contain other values such as image intensity, color, opacity, etc. In the presented examples, volume elements consist of a floating point distance value and an integer (8-bit) intensity value. While floating point distances were used in this study, preliminary tests suggest that an 8-bit distance value will be sufficient.

To illustrate the power of distance-based surface normal estimation, several images are presented that have been volume rendered using a Phong illumination model. Simple geometric objects were used in these examples so that estimation errors and image artifacts could be easily recognized and interpreted. More complex objects are rendered from binary data in Section 5. Section 4 gives some guidance for choosing sampling rates based on surface curvature.

In the examples presented here, sampled distance maps were generated from the original object representation with positive distances inside the object and negative distances outside of the object. With this sign convention, outward pointing surface normals are in the direction of the negative gradient of the distance map. Distance maps for analytic functions were calculated analytically (e.g. from the equations defining a sphere and a torus), distance maps for polygonal objects (e.g. cube and tetrahedron) were generated from the mathematical description of polygonal faces using an method adapted from collision detection algorithms [19], and distance maps for binary data were generated from an algorithm described in Section 5.

For simplicity, these examples use binary object intensities, with a value of 255 inside the object and a value of 0 outside the object. While more sophisticated rendering is possible when grey-scale image values are used to assign colors and opacities, in these examples, a single color and opacity was assigned to all voxels inside the object. Distance values were used to estimate surface normals. A straightforward front-to-back ray-casting algorithm with ray termination based on accumulated opacities was implemented for volume rendering. A Phong illumination model with ambient, diffuse and specular components was used for image shading. Two light sources were used to illuminate each object model. For each sample point, (x, y, z) , the sign of the distance value is used to determine if the sample point is inside or outside of the object. If inside, the surface normal required by the lighting model is approximated from neighboring distance values using the central difference operator:

$$n = (n_x, n_y, n_z) = \begin{pmatrix} d(x+1, y, z) - d(x-1, y, z), \\ d(x, y+1, z) - d(x, y-1, z), \\ d(x, y, z+1) - d(x, y, z-1) \end{pmatrix}. \quad (1)$$

The distance maps of the torus and sphere for Figure 3 were calculated analytically using the following expressions: 1) distance to sphere surface centered at the origin, radius R :

$$d = R - \sqrt{x^2 + y^2 + z^2}.$$

2) distance to torus centered at the origin, inside radius, $R - r$, outside radius, $R + r$:

$$d = r - \sqrt{x^2 + y^2 + z^2 + R^2 - 2R\sqrt{x^2 + y^2}}.$$

For the sphere, $R = 30$ voxels and the data was stored in a volume with dimensions 64^3 voxels. For the torus, $R = 30$, $r = 15$ and the data was stored in a volume with dimensions 64^3 voxels.

The distance maps of the four spheres in Figure 1 were also generated analytically. The spheres have radii of 30, 3, 2, and 1.5 voxels and the last three spheres were stored in arrays of size 10^3 voxels. The shading in the smaller spheres is remarkably smooth for such sparse data sampling. Artifacts that are visible around the edge of the small spheres are due to the use of a binary representation of image intensity for estimating color and opacity, rather than to errors in the surface normal estimation.

The cubes and tetrahedra of Figure 4 were generated from polygonal models. Both objects have sides of length 30 voxels and are stored in arrays of size 64^3 voxels. Image artifacts appear as bright

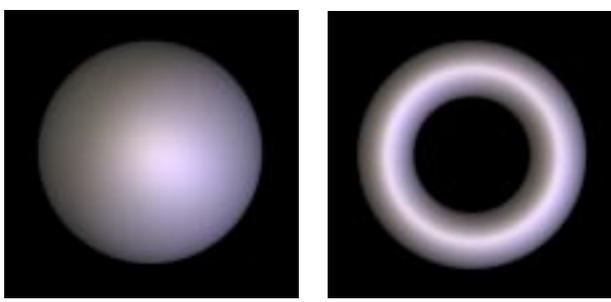


Figure 3: *Shaded, volume rendered sphere and torus stored as sampled distance-to-closest surface values. The sphere has radius 30 voxels and the torus has outside radius 30 voxels and inside radius 15 voxels. Both objects are stored in 64^3 arrays. Surface normals were estimated during rendering from 6 neighboring points using a central difference operator on the distance values. (See color plate.)*

and dark banding along object edges. These artifacts are due to high spatial frequencies at sharp edges and corners of the surface.

Figure 5 illustrates the difference in image quality for shading with distance-based surface normal estimation versus gradient estimation from a low-pass filtered version of the binary intensity values. In the filtered cases, the binary sampled object was smoothed with a Gaussian filter of radius 3, 6, and 9 (i.e. filter sizes of 7^3 , 13^3 , and 19^3) and then the surface normal is estimated from the smoothed data using central difference operator of Equation 1. It appears that the largest filter provides an acceptable shading for the 30 voxel radius sphere, but is not sufficient for the 3 voxel radius sphere. Figure 5 clearly illustrates the benefits of storing the distance map and using it to calculate surface normals when the object shape is known a priori. Section 5 addresses the challenge of creating a good distance map when the original object shape is unknown and the only representation of the object is a binary map.

4 Choosing Sampling Rates

In the previous section, it was shown that distance maps give remarkably good reconstructions of surface normals even for low sampling rates and simple interpolation and gradient estimation filters. The goal of this section is to provide both intuitive and analytic guidelines for choosing volume sampling rates so that surfaces can be accurately reconstructed. Unlike intensity-based volume representations, where the optimal sampling rate and estimation filters depend on the *sharpness* of the surface (or how quickly intensities fall off across surfaces), in the distance-based representation the optimal sampling rate and estimation filters depend on the *curvature* of the surface. For example, if the surface is planar, then its distance map is a linear 3D field and the surface can be accurately reconstructed with linear estimation filters from a low resolution distance map².

When shading surfaces in volume rendering, surface normals need only be calculated at sample points on or near the object surface. For this reason, the following analysis of the distance field is limited to regions within a few sample points from the object surface. The actual size of the region is determined by how far away

²In [20], Moller et al. showed that a linear function in \mathbb{R}^3 and its 3D derivative can be reconstructed exactly using trilinear interpolation and a central difference filter respectively. Because the shortest distance to a planar surface is linear in \mathbb{R}^3 , Moller's results indicate that a planar surface can be accurately represented in a sampled distance map and exactly reconstructed using these filters.

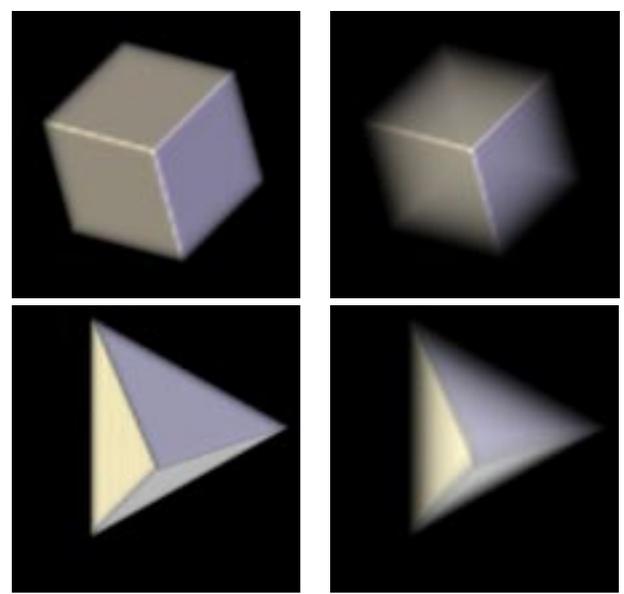


Figure 4: *Volume rendered geometric off-axis cube and tetrahedron (opaque and transparent versions). The cube is oriented at azimuth and elevation angles of 45 degrees off-axis before sampling. The cube sides are 30 voxels long and the cube is stored in an array of size 64^3 voxels. The tetrahedron has sides of length 30 voxels and is stored in an array of size 64^3 voxels. (See color plate.)*

from the surface the shading is calculated (usually within a few sample points from the surface) and the sizes of the gradient and interpolation filters (for trilinear interpolation the filter size is ± 1 , for the 6-point central difference operator with neighbors interpolated using trilinear interpolation, the filter size is ± 2).

There are 3 important cases to consider. In the first, illustrated in Figure 6, the surface is locally smooth and differentiable and relatively far from other surfaces. In the second, illustrated in Figure 7a), the surface may be locally smooth, but a fold in the surface or an abutting object causes singularities at points equidistant from two or more surface points. When such singularities lie within the region where shading is calculated, they can cause artifacts in the rendered image. Finally, as illustrated in Figure 7b), near edges or corners, the distance field can be non-linear arbitrarily close to the surface.

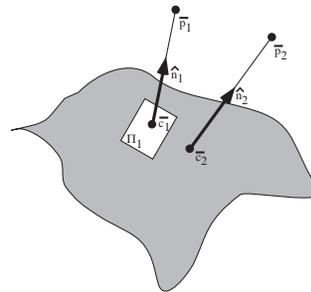


Figure 6: *A locally continuous and differentiable 3D surface.*

Considering the first case, and assuming that trilinear interpolation and a 6-point central differences gradient estimation filters will be used, the surface can be accurately reconstructed when the distance field is approximately linear within the region used to estimate surface normals. Because a smooth, differentiable surface

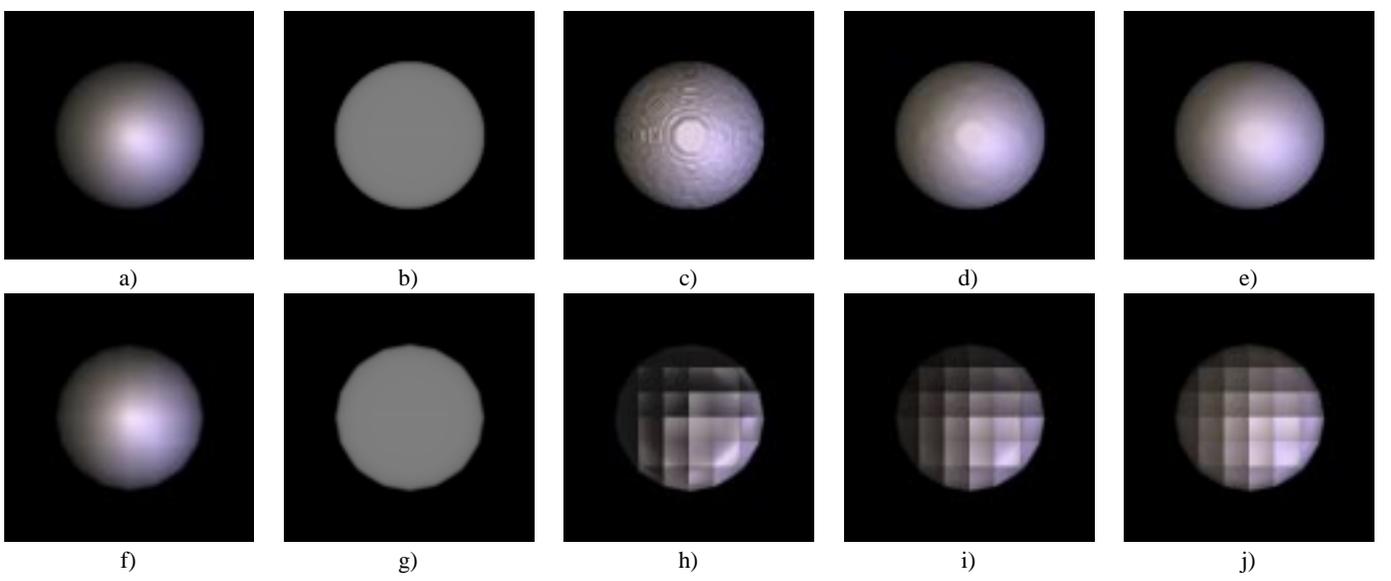


Figure 5: A comparison of volume rendered images. Images a) to e) were rendered from a sphere of radius 30 voxels, stored in an array of size 64^3 , rendered with 4 rays per voxel. Images f) to j) were rendered from a sphere of radius 3 voxels, stored in an array of size 10^3 , rendered with 400 rays per voxel. In a) and f) the distance map was used for estimating the surface normal, resulting in a smoothly shaded surface at both resolutions. In b) and g) no shading was used. In c) and h) central differences was applied to a Gaussian filtered version of the binary data with a filter of size $7 \times 7 \times 7$. In d) and i) the Gaussian filter size was $13 \times 13 \times 13$. In (e) and j) the Gaussian filter size was $19 \times 19 \times 19$. The largest filter results in acceptable surface shading of the higher resolution sphere. However, a filter of this size would result in significant loss of detail in a more complex object. None of the filters produce acceptable shading for the low resolution sphere.



Figure 7: Irregularities in the distance map. In a) folds in the surface create curves of singular points in the distance field, A and B, that are equidistant from two surfaces. When such singularities are within 2 samples from the surfaces, they can cause shading artifacts. In b) an edge or corner in the surface can create singular points or surfaces, C, and regions where the distance field is non-linear arbitrarily close to the object surface (between D and E). Dark bands along edges of the off-axis cube and tetrahedron in Figure 4 are examples of such artifacts.

looks planar within a small enough neighborhood of the surface, the distance field will appear to be linear as long as the sampling rate is high enough³.

In Figure 6, the distance from a point p_1 to its closest point on the surface, c_1 is the distance from p_1 to the tangent plane Π_1 defined by: $\hat{n}_1 \cdot \vec{x} + w_1 = 0$, where \hat{n}_1 is the unit normal to the surface at c_1 , $w_1 = -\hat{n}_1 \cdot c_1$, and $\vec{x} \in \mathbb{R}^3$. The distance from p_1 to c_1 is:

$$d(p_1) = (p_1 - c_1) \cdot \hat{n}_1$$

A second point, p_2 , is located within a local neighborhood of the

³Of course, following Moller et al., higher order reconstruction filters could be used to reconstruct a non-linear distance field with a lower sampling rate so that, just as in intensity-based volume rendering, there are trade-offs between filter complexity, sampling rate, and accuracy.

surface point c_1 . The distance from point p_2 to its closest point on the surface, c_2 is the distance from p_2 to the tangent plane at c_2 . The distance at p_2 is:

$$\begin{aligned} d(p_2) &= (p_2 - c_2) \cdot \hat{n}_2 \\ &= (p_2 - c_1) \cdot \hat{n}_1 + (p_2 - c_2) \cdot (\hat{n}_2 - \hat{n}_1) - (c_2 - c_1) \cdot \hat{n}_1. \end{aligned} \quad (2)$$

The first term of Equation 2 is the linear component of the 3D distance field. When this term dominates, the distance field is approximately linear. The remaining 2 terms are the non-linear components and they are largest in the direction perpendicular to the normal direction. They are affected by the surface curvature – how quickly the surface normal and the closest point position vary over the surface as p_2 moves around \mathbb{R}^3 . When the surface is locally planar, these terms are zero. Otherwise, for points close to the surface, the non-linear terms will be very small. For points farther from the surface, the term $(p_2 - c_2) \cdot (\hat{n}_2 - \hat{n}_1)$ will dominate the non-linear terms. For a given surface, the maximum expected magnitude of this term could be used to determine an optimal sampling rate for a given acceptable reconstruction error when linear estimation filters are used.

When there are singularities in the distance field or when edges or corners generate a non-linear distance fields arbitrarily close to the surface, there are a number of possible approaches for avoiding artifacts in the shaded image. Fortunately, it is easy to detect the presence of such irregularities in the distance map. When the distance map is approximately linear, the magnitude of the gradient calculated with the central difference filter of Equation 1 is constant and equal to twice the unit distance between samples. When the sample point is close to a singularity, an edge, or a corner of the surface, the gradient magnitude will be significantly different from this value and an irregularity can be detected. If an irregularity in the distance map is detected, then more sophisticated methods could be used to estimate the surface normal. Two such methods

include: 1) using a higher order interpolation and gradient estimation filters (such as filters that reconstruct spherical or cylindrical fields) and 2) using the optional gradient estimator from Tiede et al. [29], which estimates the surface normals of thin structures from an optimal subset of the 6 neighbors of the sample point.

5 Calculating Distances from Binary Data

One of the most common sources of binary volume data is segmented medical images, where different tissues or structures are each assigned a unique classification or type. When the original grey-scale data is not available or when consistent surfaces are not easily extracted from grey-scale images, then the distance map method can be used to provide high quality shading in volume rendering. However, unlike the analytic and polygonal models presented so far, binary volumetric data does not contain an accurate representation of object surfaces. This section discusses why generating surface normals from binary segmentations can be better than using the original grey-scale data, briefly describes the generation of distance maps from binary data using a new method for creating a smooth surface representation of objects in the binary volume, and presents images volume rendered from such distance maps.

While there are many cases where the gradient of grey-scale image data provides good estimates of surface normals [9], when surfaces are not easily extracted from the grey-scale data these estimates can produce severe artifacts. A problematic case is illustrated in the 2D MRI image of a human knee of Figure 8. Figure 8b) was calculated by applying a central difference operator to Figure 8a) at hand-segmented edge points along the femur, one of the bones in the knee. Because the bone surface is generally smooth and of uniform density, surface normals should have relatively constant magnitudes and slowly varying directions. However, because the gradient depends on the thickness and image intensity of materials that are adjacent to the bone surface, and because the materials adjacent to the femur vary significantly, the estimated normals change dramatically around the edge of the femur. Even when the magnitude of the gradient vector is normalized as in Figure 8c), unexpected and sudden changes in the direction of the image gradient can introduce severe artifacts into shaded images. In such cases, a binary segmentation of the image volume can provide better surface normal estimates than the original grey-scale images.

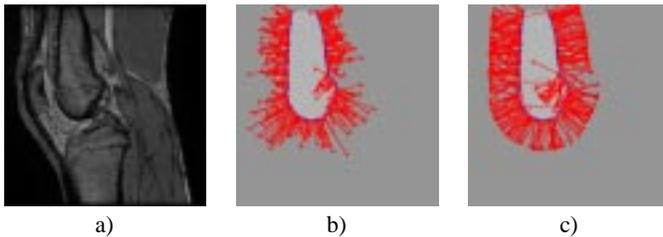


Figure 8: a) 2D Magnetic Resonance Image (MRI) cross section through a human knee. b) Image gradient vectors calculated using central differences on the grey-scale data at edge points of the segmented femur. c) Image gradient vectors with a normalized magnitude. The direction and magnitude of image gradients vary much more than we would expect the surface normals of the knee bone to vary, in some cases pointing inward when we expect an outward facing normal. Hence, applying a gradient operator to the grey-scale data does not always provide a good estimate of surface normals. (Data and segmented image courtesy of R. Kikinis, Brigham and Women’s Hospital, Boston MA).

In binary-sampled data, exact knowledge of the underlying object surface is missing and hence distance maps must be approxi-

mated from the binary data rather than from the true surface. [6] presents experimental results for 5 methods that estimate the distance map from binary sampled data. These include methods that estimate distances directly from the binary values (such as chess-board distance and the city-block distance metrics [25]) and methods estimate a surface from local binary values and then calculate distances from this surface. All of these methods produce artifacts in shaded surfaces which are especially significant when the voxel data is not isometric (i.e. when the data is sampled at different resolutions along the three major axes). In [7], a new method for generating a smooth surface representation from binary sampled data is presented. This method, Constrained Elastic Surface Nets, is described here briefly. Surface Nets produce surfaces that are free from aliasing and terracing artifacts and which can be used to generate volumetric distance maps. Examples of volume rendered images that used distance maps for shading object surfaces created from binary segmented MRI data are presented in Figures 9c) and 10.

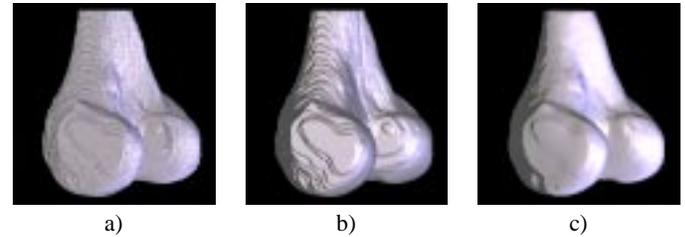


Figure 9: Volume rendered knee bone (femur) with in-plane resolution of 0.25 mm and between-plane spacing of 1.4 mm. In a) surface normals were calculated directly from the binary data. In b) the binary data was filtered with a Gaussian low-pass filter of size 13^3 and surface normals were estimated from the filtered data. In c) a distance map was generated from a constrained surface net applied to the binary data and surface normals were estimated from the distance map. Only the distance map method provides acceptable shading of the femur. In each case, a 6-point central difference gradient operator was used to estimate surface normals. Large diffuse and specular coefficients were used in rendering to emphasize surface artifacts. (See color plate.)

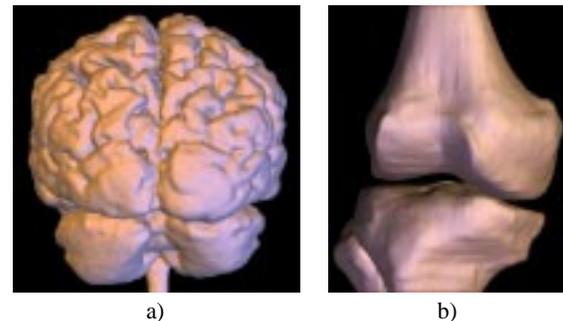


Figure 10: a) Brain and b) knee bones volume rendered with shading from a distance map representation. The size of the brain data set is $137 \times 146 \times 119$. The size of the knee data is $304 \times 300 \times 91$ (in-plane spacing 0.27 mm, between-plane spacing 1.0 mm). Both the brain [13] and knee data originated as binary segmented MRI. The distance maps were created from a constrained elastic surface net that was stretched over the surface of the binary data. (See color plate.)

5.1 Constrained Elastic Surface Nets

The goal of the surface net approach is to create a globally smooth surface model from binary segmented data that retains fine detail present in the original segmentation. Methods that apply local low-pass filters to the binary data can reduce aliasing but they are not effective at removing terracing artifacts that occur in binary data. In addition, low-pass filters can eliminate fine structures or cracks that may be especially important in medical applications. Surface nets are constructed by linking nodes on the surface of the binary-segmented volume and relaxing node positions to reduce energy in the surface net while constraining the nodes to lie within a surface cube defined by the original segmentation. Figure 11 illustrates how a linked net of surface points can smooth out terracing artifacts.

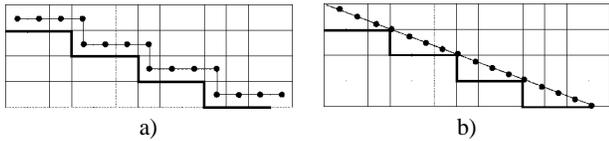


Figure 11: *Terracing artifacts in binary segmented data cause smooth surfaces to appear jagged. a) A linked net of surface nodes is constructed, placing one node at the center of each surface cube. b) Constrained elastic relaxation of the surface net smooths out terraces but keeps each surface node within its original surface cube.*

The first step in generating a surface net is to locate cubes that contain surface nodes. A cube is defined by 8 neighboring voxels in the binary segmented data, 4 voxels each from 2 adjacent planes. If all 8 voxels have the same binary value, then the cube is either entirely inside or entirely outside of the object. If at least one of the voxels has a binary value that is different from its neighbors, then the cube is a surface cube. The net is initialized by placing a node at the center of each surface cube and linking nodes that lie in adjacent surface cubes. Each node can have up to 6 links, one each to its right, left, top, bottom, front, and back neighbors.

Once the surface net has been defined, the position of each node is relaxed to reduce an energy measure in the links. In the examples presented here, surface nets were relaxed iteratively by considering each node in sequence and moving that node towards a position midway between its linked neighbors. The energy was computed as the sum of the squared lengths of all of the links in the surface net. Defining the energy and relaxation in this manner without constraints will cause the surface net to shrink into a sphere and eventually onto a single point. To remain faithful to the original segmentation, a constraint is applied that keeps each node in its original surface cube. This constraint favors the original segmentation over smoothness and forces the surface to retain thin structures and cracks. Figure 12 shows the application of surface nets to some 2D bitmaps. The surface nets generate relatively smooth surfaces for curves objects, produce sharp corners for rectangular objects, and preserve thin structures and cracks.

Because the surface net representation contains explicit links between neighboring nodes, it is relatively straightforward to create a triangulated surface from the relaxed surface net. [7] describes how the triangulated surface is created and how the distance map can be generated from the surface. It also discusses implementation and timing issues. Figures 9c) and 10 illustrate that this method can be used to generate smooth surfaces from binary segmented data at fairly low data resolutions that are free from aliasing and terracing artifacts.

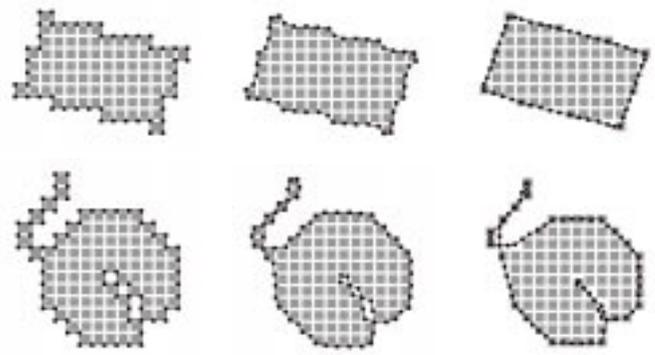


Figure 12: *Examples of surface nets applied to 2D binary objects. The top row shows the surface net at initialization, after one relaxation and after 30 relaxations for a tilted rectangle. The bottom row shows the surface net at initialization, after one relaxation and after 20 relaxations for an object with a crack and a thin protrusion. After relaxation, curved surfaces are relatively smooth, corners are sharp, and thin structures are preserved.*

6 Future Work

The ability to represent surfaces accurately in volume graphics opens up many research directions in volume rendering, haptics, and physics based modeling. We have begun to investigate a number of ways to improve the methods presented here and to explore new applications and algorithms based on the distance map representation. Some of these research directions are outlined here.

While this paper stored distances as floating point values, we have begun to look at methods for discretizing distance values into 8 bits. The work of [30, 31, 1, 21] suggest that 8 bits will be sufficient for many surfaces. We have begun to study the discretization errors and storage tradeoffs for these methods. We have also applied space-leaping methods to reduce the sampling of empty space and improve rendering times [35, 27].

In haptic rendering and physics-based modeling, the distance map yields both surface normals and penetration depths required for calculating interaction forces. In addition, distance maps can be used both for detecting and anticipating object collisions. In preliminary studies, we have found that a sampled distance map representation with a central difference surface normal estimator provides a smooth surface in haptic rendering even for a relatively low resolution sphere (20x20x20 voxels). We have also successfully tested a distance-based haptic model of the knee in Figure 10b). We intend to continue exploring application of the distance map to haptic and physics-based modeling.

7 Conclusions

Good surface representation is important for high-quality, realistic rendering in computer graphics. Unfortunately, while volumetric objects have advantages over surface models when objects have complex interior structure, traditional intensity-based sampled data do not represent arbitrary surfaces well. This paper has presented a distance map approach for representing surfaces in sampled volumetric data. In this approach, distance-to-closest-surface values are mapped to each sample point and surfaces are reconstructed from this distance map.

The distance map has some attractive properties. First, the zero-value of the distance map locates surfaces while the gradient of the distance map yields surface normals. Second, when the sampling

rate of the volumetric data is adequate (i.e. when it is large relative to the surface curvature), a low-cost 6-point central difference gradient estimator applied to the distance map can accurately reconstruct surface normals near the surface. Third, although folds and object edges or corners introduce singularities and non-linearities into the distance field that cause shading artifacts with a central difference gradient estimator, the presence of these irregularities in the distance map can be easily detected during rendering so that higher order filters or more sophisticated gradient estimation methods can be locally applied. Fourth, in addition to applications in volume rendering, the distance map approach can be used to reconstruct surface normals and penetration distances for applications in haptics and physics-based modeling.

The effectiveness of the distance map for encoding analytic and polygonal surfaces in volumetric data has been illustrated with a number of examples, including a torus, spheres with radii varying from 1.5 to 30 voxels, an off-axis cube and a tetrahedron. In addition, a new method for calculating distance maps from binary data has been described and illustrated with shaded, volume rendered images from binary-segmented medical data.

8 Acknowledgements

Thanks to R. Osborne and C. Rich for questions that eventually led to ideas in this paper. Thanks to B. Mirtich for many useful discussions. Thanks to H. Pfister, B. Freeman, H. Lauer and E. Gibson for helpful suggestions and comments. I am indebted to R. Kikinis and the SPL lab at Brigham and Women's Hospital for medical images, segmented data, and motivation behind this work.

References

- [1] R. Avila and L. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of Visualization'96*, pages 197–204. IEEE, 1996.
- [2] M. Bentum, B. Lichtenbelt, and T. Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, 1996.
- [3] I. Carlbaum. Optimal filter design for volume reconstruction and visualization. In *Proceedings of Visualization'93*, pages 54–61. IEEE, 1993.
- [4] J. Foley, A. vanDam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 1992.
- [5] S. Gibson. Linked volumetric objects for physics-based modeling. Technical Report TR97-20, MERL – A Mitsubishi Electric Research Laboratory, 1997.
- [6] S. Gibson. Calculating distance maps from binary segmented data. Technical Report WP98-??, MERL – A Mitsubishi Electric Research Laboratory, 1998.
- [7] S. Gibson. Constrained elastic Surface Nets: generating smooth surfaces from binary segmented data. In *submitted to Medical Image Computation and Computer Assisted Surgery, 1998*, pages proceedings, MICCAI'98, 1998.
- [8] T. Guenther, C. Poliwoda, C. Reinhard, J. Hesser, R. Maenner, H.-P. Meinzer, and H.-J. Baur. VIRIM: A massively parallel processor for real-time volume visualization in medicine. In *Proceedings of the 9th Eurographics Hardware Workshop*, pages 103–108, Oslo, Norway, September 1994.
- [9] K. Hohne, M. Bomans, A. Pommert, M. Riemer, C. Schiers, U. Tiede, and G. Wiebecke. 3d visualization of tomographic volume data using the generalized voxel model. *The Visual Computer*, 6(1):28–36, February 1990.
- [10] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual voyage: Interactive navigation in the human colon. In *Computer Graphics Proceedings, Annual Conference Series, Proceedings of SIGGRAPH'97*, pages 27–34. ACM SIGGRAPH, 1997.
- [11] A. Kaufman. *Volume Visualization*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [12] A. Kaufman, D. Cohen, and R. Yagel. Chapter 4: Volumetric shading techniques. In A. Kaufman, editor, *Volume Visualization*, pages 169–173. IEEE Computer Society Press, 1991.
- [13] R. Kikinis, M. Shenton, D. Iosifescu, R. McCarley, P. Saiviroonporn, H. Hokama, A. Robatino, D. Metcalf, C. Wible, C. Portas, R. Donnino, and F. Jolesz. A digital brain atlas for surgical planning, model driven segmentation, and teaching. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):–, September 1996.
- [14] D. Koditschek. Planning and control via potential functions. In T. Lozano-Perez and O. Khatib, editors, *Robotics Review I*, pages 349–367. MIT Press, 1989.
- [15] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Computer Graphics Proceedings, Annual Conference Series, Proceedings of SIGGRAPH'94*, pages 451–457. ACM SIGGRAPH, 1994.
- [16] W. Lorensen and H. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Computer Graphics Proceedings, Annual Conference Series, Proceedings of SIGGRAPH 87*, pages 163–169. ACM SIGGRAPH, 1989.
- [17] S. Lu, D. Cui, R. Yagel, R. Miller, and G. Kinzel. A 3-d contextual shading method for visualization of diecasting defects. In *Proceedings of Visualization'96*, pages 405–407. IEEE, 1996.
- [18] S. Marschner and R. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of Visualization'94*, pages 100–107. IEEE, 1994.
- [19] B. Mirtich. *V-Clip: Fast and robust polyhedral collision detection*. Technical Report TR97-05, Mitsubishi Electric Research Lab, Cambridge, MA, July 1997.
- [20] T. Moller, R. Machiraju, K. Mueller, and R. Yagel. Evaluation and design of filters using a Taylor series expansion. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):184–199, June 1997.
- [21] A. Mor, S. Gibson, and J. Samosky. Interacting with 3-dimensional medical data: Haptic feedback for surgical simulation. In *Proceedings of Phantom User Group Workshop'96*, 1996.
- [22] A. Oppenheim and R. Schaffer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [23] R. Osborne, H. Pfister, H. Lauer, N. McKenzie, S. Gibson, W. Hiatt, and T. Ohkami. EM-Cube: An architecture for low-cost real-time volume rendering. In *Proceedings of SIGGRAPH/Eurographics Workshop on Graphics and Hardware*, pages 131–138. ACM SIGGRAPH, 1997.
- [24] H. Pfister. *Architectures for Real-Time Volume Rendering*. PhD thesis, State University of New York at Stony Brook, Computer Science Department, Stony Brook, NY 11794-4400, 1996. MERL Report No. TR-97-04.
- [25] J. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, FL, 1992.
- [26] W. Schroeder, W. Lorensen, and S. Linthicum. Implicit modeling of swept surfaces and volumes. In *Proceedings of Visualization'94*, pages 40–45. IEEE, 1994.
- [27] M. Sramek. Fast surface rendering from raster data by voxel traversal using chessboard distance. In *Proceedings of Visualization'94*, pages 188–195. IEEE, 1994.
- [28] G. Thurmer and C. Wurthrich. Normal computation for discrete surfaces in 3d space. In *Proceedings of Eurographics'97*, pages C–15–C–26. Eurographics Association, 1997.
- [29] U. Tiede, K. Hohne, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke. Surface rendering: Investigation of medical 3d-rendering algorithms. *IEEE Computer Graphics and Applications*, 10:41–53, 1990.
- [30] S. Wang and A. Kaufman. Volume sampled voxelization of geometric primitives. In *Proceedings of Visualization'93*, pages 78–84. IEEE, 1993.
- [31] S. Wang and A. Kaufman. Volume-sampled 3d modeling. *IEEE Computer Graphics and Applications*, 14:26–32, 1994.
- [32] R. Yagel, D. Cohen, and A. Kaufman. Discrete ray tracing. *IEEE Computer Graphics and Applications*, 12:19–28, 1992.
- [33] R. Yagel, D. Cohen, and A. Kaufman. Normal estimation in 3d discrete space. *The Visual Computer*, 10(2):116–124, 1993.
- [34] R. Yagel, S. Lu, A. Rebello, and R. Miller. Volume-based reasoning and visualization of diecastability. In *Proceedings of Visualization'95*, pages 359–362. IEEE, 1995.
- [35] K. Zuiderveld, A. Koning, and M. Viergever. Acceleration of ray-casting using 3d distance transforms. In *Visualization in Biomedical Computing, Proceedings of VBC'92*, pages 324–335. SPIE, 1992.