Journal of Electronic Imaging 12(3), 470–477 (July 2003).

Method of normal estimation based on approximation for visualization

Dihui Hong
Gangmin Ning
Ting Zhao
Mu Zhang
Xiaoxiang Zheng
Zhejiang University
Department of Biomedical Engineering
Hangzhou, Zhejiang 310027
China
E-mail: zxx@mail.hz.zj.cn

Abstract. *A normal estimation algorithm for visualization is presented that approximates the density function in a local neighborhood with a second-degree polynomial function. The coefficients of the polynomial function can be solved by minimizing the error of the approximation. This method is tested in several volume data sets and comparisons with other methods are presented. It is demonstrated that this method is a fairly robust technique for noise-contained data and is preferable for most applications.* © 2003 SPIE and IS&T. [DOI: 10.1117/1.1579698]

1 Introduction

The computation of data set gradients is an essential operation in visualization algorithms.[1] Gradient direction and magnitude are often used to shade and classify the volume data. If the gradient estimation is not done carefully, shading and classification will yield wrong gray values and opacities. It is easy to obtain a normal vector from a geometric model, but that is not the case when volume data sets such as computed tomography (CT) and magnetic resonance imaging (MRI) data are used. These sampled values do not contain any geometric information of an object's surface, thus normal estimation is required for restoration of geometric information from the sampled values.

Commonly used methods for normal estimation in volume rendering are based on interpolation and derivative. Moller *et al.* divided these methods into two classes and analyzed them in terms of accuracy and efficiency.[2] In the first class, the gradient estimation is decomposed in two steps by employing a digital derivative filter and a continuous interpolation filter (e.g., central difference + linear interpolation). The other schemes proposed estimate the normal at an arbitrary point in one step by convolving the volume data with the analytical derivative of the interpolation filter. For the interpolation and derivative method, the estimation error depends on the error of the derivative and interpolation filters. There have been various comparisons of accuracy of interpolation filters.[3–5] Although the sinc function provides an exact reconstruction of original signal and the gradient can be reconstructed exactly using the derivative of the sinc function (cosc) as a reconstruction kernel, it is defined over an infinite spatial range and therefore cannot be used practically. Since the reconstruction filter is usually not ideal, the choice of interpolation scheme has a great effect on image quality. The error resulting from reconstruction falls into three main categories: smoothing, postaliasing, and overshoot. It is shown that cubic spline interpolation function performs very well.[5–7] In a cubic spline function, a parameter remains free to be chosen. By adjusting this parameter, the performance of the gradient operator will vary. There is no reason to prefer any particular parameter, since we must settle for a cost/benefit between smoothing and postaliasing. The B-spline has the heaviest smoothing, but has low postaliasing, while the Catmull-Rom spline produces much less smoothing but has poor postaliasing properties .[8] Since the interpolating curve passes through all the given data points rather than fitting the data polygon, it is very sensitive to noise.

In this paper, a method based on approximation is introduced to overcome the problem by relaxing the strict requirement that the curve must pass through all given data points. Neumann *et al.* approximated the density function in a local neighborhood with a 3-D regression hyperplane.[9] However, a linear function is not sufficient to represent a density function, especially for a rapidly changing signal. Also it is computationally expensive because interpolation must be used to acquire the normal vector at an arbitrary point. In our approach, the density distribution in a local neighborhood is expressed by a second-degree polynomial function. The higher order of the polynomial not only improves the quality of the approximation but also enhances its noise reduction ability. The coefficients of this function are obtained by minimizing the error of the approximation and the gradient vector at an arbitrary point can be obtained

Paper: JEI01068 received Nov. 20, 2001; revised manuscript received Oct. 18, 2002; accepted for publication Jan. 7, 2003.
1017-9909/2003/$15.00 © 2003 SPIE and IS&T.

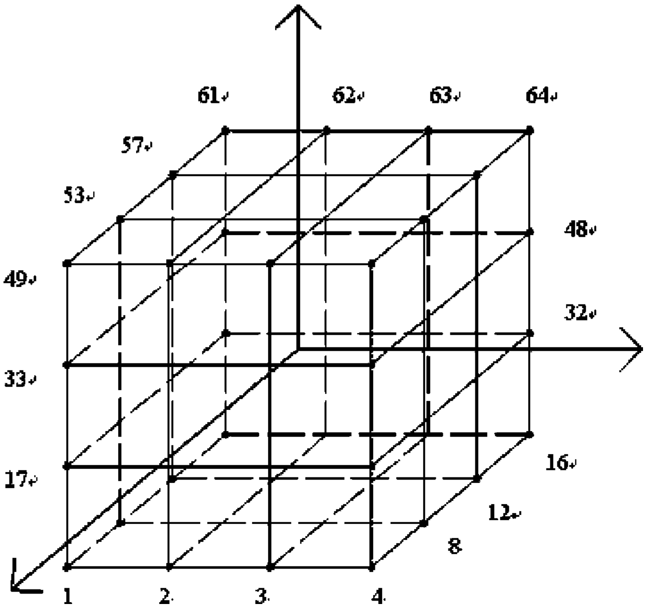

Fig. 1 Indexing the neighboring voxel.

from the analytical derivative of the density function. Because of symmetry, the solution of this equation can be simplified.

2 Method

Assuming that the normal vector at a point $P(x,y,z)$ in volume data is to be calculated, the per unit length in each coordinate direction is the interval between adjacent voxels, and the origin of the coordinate system is translated into the center of this point's close 4×4×4 neighborhood, as shown in Fig. 1. Thus the density function $f(x,y,z)$ in this neighborhood can be approximated by the following formula:

$$f(x,y,z)=Ax^2+By^2+Cz^2+2Fyz+2Gzx+2Hxy+2Ix \\ +2Jy+2Kz+D. \tag{1}$$

The gradient of the volume data set is denoted by n and it can be derived from the density function.

$$n=(Ax+Gz+Hy+I,\, By+Fz+Hx+J, \\ Cz+Fy+Gx+K).$$

To determine the gradient n, the parameters A, B, C, D, E, F, G, H, I, J, and K must be solved. The solution to the relevant parameters is given later.

To measure how well the density function approximates the given volume data, the concept of error distance is introduced. The error distance is the distance between a calculated data point from the approximation formulation and its ‘‘corresponding’’ point in the actual data. The accuracy of the approximation $f(x,y,z)$ can be measured by the error distance. To make the density function fit the volume data closely, the sum of these error distances as in Eq. (2) should be minimized.

$$E(A,B,C,D,F,G,H,I,J,K) \\ =\sum_{k=1}^{k=64} w_k(Ax_k^2+By_k^2+Cz_k^2+2Fy_kz_k+2Gz_kx_k \\ +2Hx_ky_k+2Ix_k+2Jy_k+2Kz_k+D-f_k)^2, \tag{2}$$

where x_k, y_k, and z_k denote the coordinates of the k'th neighboring voxel. The measured density value at the k'th voxel is denoted by f_k. The error caused by the estimation in k'th sample contributes to the error distance with a weighting function w_k.

The weighting function w_k is defined as

$$w_k=\frac{1}{d_k},$$

where $d_k=(x_k^2+y_k^2+z_k^2)^{1/2}$ is the distance of the k'th neighboring voxel from the origin. The indices k are assigned to the neighboring voxel locations.

The set of the parameters A, B, C, D, E, F, G, H, I, J, and K can be obtained by the least-squares method:

$$\frac{\partial E}{\partial A}=2\sum_{k=1}^{k=64} w_k(Ax_k^2+By_k^2+Cz_k^2+2Fy_kz_k+2Gz_kx_k \\ +2Hx_ky_k+2Ix_k+2Jy_k+2Kz_k+D-f_k)x_k^2=0$$

$$\vdots$$

$$\frac{\partial E}{\partial K}=2\sum_{k=1}^{k=64} w_k(Ax_k^2+By_k^2+Cz_k^2+2Fy_kz_k+2Gz_kx_k \\ +2Hx_ky_k+2Ix_k+2Jy_k+2Kz_k+D-f_k)z_k=0.$$

Therefore, the normal calculation is equivalent to solving a 10-D linear equation.

$$\mathbf{M}\begin{bmatrix} A \\ B \\ C \\ F \\ G \\ H \\ I \\ J \\ K \\ D \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{k=64} w_k f_k x_k^2 \\ \sum_{k=1}^{k=64} w_k f_k y_k^2 \\ \sum_{k=1}^{k=64} w_k f_k z_k^2 \\ \sum_{k=1}^{k=64} w_k f_k y_k z_k \\ \sum_{k=1}^{k=64} w_k f_k z_k x_k \\ \sum_{k=1}^{k=64} w_k f_k x_k y_k \\ \sum_{k=1}^{k=64} w_k f_k x_k \\ \sum_{k=1}^{k=64} w_k f_k y_k \\ \sum_{k=1}^{k=64} w_k f_k z_k \\ \sum_{k=1}^{k=64} w_k f_k \end{bmatrix}, \tag{3}$$

where **M** is the coefficient matrix. After simplification, **M** can be obtained as:

$$\mathbf{M}=\begin{bmatrix} m_1 & m_6 & m_5 & 0 & 0 & 0 & 0 & 0 & 0 & m_7 \\ m_6 & m_2 & m_4 & 0 & 0 & 0 & 0 & 0 & 0 & m_8 \\ m_5 & m_4 & m_3 & 0 & 0 & 0 & 0 & 0 & 0 & m_9 \\ 0 & 0 & 0 & m_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_9 & 0 \\ m_7 & m_8 & m_9 & 0 & 0 & 0 & 0 & 0 & 0 & m_{10} \end{bmatrix}, \tag{4}$$

where

$$m_1=\sum_{k=1}^{64} w_k x_k^4, \quad m_2=\sum_{k=1}^{64} w_k y_k^4,$$

$$m_3=\sum_{k=1}^{64} w_k z_k^4, \quad m_4=\sum_{k=1}^{64} w_k y_k^2 z_k^2,$$

$$m_5=\sum_{k=1}^{64} w_k x_k^2 z_k^2, \quad m_6=\sum_{k=1}^{64} w_k x_k^2 y_k^2,$$

$$m_7=\sum_{k=1}^{64} w_k x_k^2, \quad m_8=\sum_{k=1}^{64} w_k y_k^2,$$

$$m_9=\sum_{k=1}^{64} w_k z_k^2, \quad m_{10}=\sum_{k=1}^{64} w_k .$$

Because of the symmetry property, most elements in the coefficient matrix **M** are zeroes. Thus, the computation complexity in solving the gradient of data set is greatly reduced.

Equation (3) can be decomposed into two separate ones:

$$\begin{bmatrix} m_1 & m_6 & m_5 & m_7 \\ m_6 & m_2 & m_4 & m_8 \\ m_5 & m_4 & m_3 & m_9 \\ m_7 & m_8 & m_9 & m_{10} \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{k=64} w_k f_k x_k^2 \\ \sum_{k=1}^{k=64} w_k f_k y_k^2 \\ \sum_{k=1}^{k=64} w_k f_k z_k^2 \\ \sum_{k=1}^{k=64} w_k f_k \end{bmatrix}, \tag{5}$$

$$\begin{bmatrix} m_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_9 \end{bmatrix} \cdot \begin{bmatrix} F \\ G \\ H \\ I \\ J \\ K \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{k=64} w_k f_k y_k z_k \\ \sum_{k=1}^{k=64} w_k f_k x_k z_k \\ \sum_{k=1}^{k=64} w_k f_k x_k y_k \\ \sum_{k=1}^{k=64} w_k f_k x_k \\ \sum_{k=1}^{k=64} w_k f_k y_k \\ \sum_{k=1}^{k=64} w_k f_k z_k \end{bmatrix}. \tag{6}$$

Such linear equations as shown in Eqs. (5) and (6) can be solved easily.

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ u_2 & u_5 & u_6 & u_7 \\ u_3 & u_6 & u_8 & u_9 \\ u_4 & u_7 & u_9 & u_{10} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{k=1}^{k=64} w_k f_k x_k^2 \\ \sum_{k=1}^{k=64} w_k f_k y_k^2 \\ \sum_{k=1}^{k=64} w_k f_k z_k^2 \\ \sum_{k=1}^{k=64} w_k f_k \end{bmatrix},$$

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ u_2 & u_5 & u_6 & u_7 \\ u_3 & u_6 & u_8 & u_9 \\ u_4 & u_7 & u_9 & u_{10} \end{bmatrix}^{-1} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_2 & t_5 & t_8 & t_9 \\ t_3 & t_8 & t_6 & t_{10} \\ t_4 & t_9 & t_{10} & t_7 \end{bmatrix},$$

$$\begin{aligned}\Delta = & -u_{10}u_3^2u_5 + 2u_{10}u_2u_3u_6 - u_1u_{10}u_6^2 + u_4^2u_6^2 \\ & -2u_3u_4u_6u_7 + u_3^2u_7^2 - u_{10}u_2^3u_8 + u_1u_{10}u_5u_8 - u_4^2u_5u_8 \\ & +2u_2u_4u_7u_8 - u_1u_7^2u_8 + 2u_3u_4u_5u_9 - 2u_2u_4u_6u_9 \\ & -2u_2u_3u_7u_9 + 2u_1u_6u_7u_9 + u_2^2u_9^2 - u_1u_5u_9^2,\end{aligned}$$

$$t_1 = (-u_{10}u_6^2 + u_{10}u_5u_8 - u_7^2u_8 + 2u_6u_7u_9 - u_5u_9^2)/\Delta,$$

$$\begin{aligned}t_2 = & (u_{10}u_3u_6 - u_{10}u_2u_8 + u_4u_7u_8 - u_4u_6u_9 - u_3u_7u_9 \\ & +u_2u_9^2)/\Delta,\end{aligned}$$

$$\begin{aligned}t_3 = & (-u_{10}u_3u_5 + u_{10}u_2u_6 - u_4u_6u_7 + u_3u_7^2 + u_4u_5u_9 \\ & -u_2u_7u_9)/\Delta,\end{aligned}$$

$$\begin{aligned}t_4 = & (u_4u_6^2 - u_3u_6u_7 - u_4u_5u_8 + u_2u_7u_8 + u_3u_5u_9 \\ & -u_2u_6u_9)/\Delta,\end{aligned}$$

$$t_5 = (-u_{10}u_3^2 + u_1u_{10}u_8 - u_4^2u_8 + 2u_3u_4u_9 - u_1u_9^2)/\Delta,$$

$$t_6 = (-u_{10}u_2^2 + u_1u_{10}u_5 - u_4^2u_5 + 2u_2u_4u_7 - u_1u_7^2)/\Delta,$$

$$t_7 = (-u_3^2u_5 + 2u_2u_3u_6 - u_1u_6^2 - u_2^2u_8 + u_1u_5u_8)/\Delta,$$

$$\begin{aligned}t_8 = & (u_{10}u_2u_3 - u_1u_{10}u_6 + u_4^2u_6 - u_3u_4u_7 - u_2u_4u_9 \\ & +u_1u_7u_9)/\Delta,\end{aligned}$$

$$\begin{aligned}t_9 = & (-u_3u_4u_6 + u_3^2u_7 + u_2u_4u_8 - u_1u_7u_8 - u_2u_3u_9 \\ & +u_1u_6u_9)/\Delta,\end{aligned}$$

$$\begin{aligned}t_{10} = & (u_3u_4u_5 - u_2u_4u_6 - u_2u_3u_7 + u_1u_6u_7 + u_2^2u_9 \\ & -u_1u_5u_9)/\Delta,\end{aligned}$$

$$F = \sum_{k=1}^{k=64} w_k f_k y_k z_k / m_4, \quad G = \sum_{k=1}^{k=64} w_k f_k x_k z_k / m_5,$$

$$H = \sum_{k=1}^{k=64} w_k f_k x_k y_k / m_6, \quad I = \sum_{k=1}^{k=64} w_k f_k x_k / m_7,$$

$$J = \sum_{k=1}^{k=64} w_k f_k y_k / m_8, \quad K = \sum_{k=1}^{k=64} w_k f_k z_k / m_9.$$

Thus, the normal vector of the point $P(x,y,z)$, which defined as $[X,Y,Z]$, is obtained:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2A & 2H & 2G \\ 2H & 2B & 2F \\ 2G & 2F & 2C \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 2I \\ 2J \\ 2K \end{bmatrix}.$$

3 Results and Discussion

For interpolation and derivative methods, the derivative filter's characteristics are analyzed by their Taylor series coefficients a_n^w and the error coefficients r_N^w. Some commonly used convolution filter combinations are computed and compared by this method in previous studies.[2,4] The principle idea is to choose the largest N such that all coefficients $a_n^w(n \leqslant N) = 0$, with the only exception being a_0^w for interpolation filters and a_n^w for the n'th-order derivative filters, which should equal to one. An N-EF (N'th degree error) filter will reconstruct a polynomial of (N-1)'th or lower degree without any error. Filter combination of linear interpolation and central difference is denoted as DL. The combination of a cubic interpolation filter (cubic cardinal splines, which is introduced in Ref. 10) with central difference is denoted by DC. Here C represents the class of analytic derivative cubic filters. In cubic spline function, a parameter α is left free to be chosen. From the error coefficients calculation, it is found that in the case that $\alpha \neq -0.5$, only the DL is a 3EF group. If $\alpha = -0.5$ (Catmull-Rom spline), it is proved that the worst behavior is observed for DL, and C is more accurate than DC, though in this case all three methods are 3EF. Since the C for $\alpha = -0.5$ is the optimal derivative filter and central difference with linear interpolation is the most commonly used method, these two interpolation methods are compared with our method in our experiments.

We display isosurfaces of reconstructed test volumes using various gradient estimators to demonstrate the behavior of the gradient estimation methods. The images are rendered employing a simple raycaster to find the isosurfaces.[11–13] The volume data are resampled at an interval of 0.1 voxel length. At each sample point, an interpolation kernel is applied to compute the value at that point. If the reconstructed value is above a preset isovalue, the ray is terminated and the gradient estimation method is used to compute the normal vector. Shading is then performed using traditional Phong lighting model.[14]

To compare the estimated normal vector with the exact one, a differential density function $g(x,y,z) = (x-64)^2 + (y-64)^2 + (z-64)^2$ is designed. A volume data set is derived from the analytical formula by 256 gray levels discretizing. Figures 2(a) to 2(e) demonstrate the difference among the involved normal estimation methods. The mean error and standard deviation (stdv) of the calculated normal

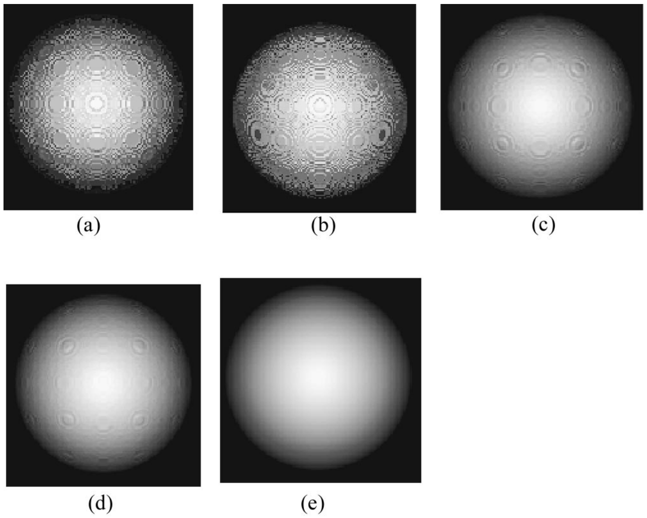

Fig. 2 Normal estimation on gray-scaled volumes of resolution 128×128×128 ball data using different methods. (a) Central difference+interpolation method, mean error 9.09°, standard deviation 10.43°; (b) derivative of Catmull-Rom spline, mean error 10.31°, standard deviation 12.07°; (c) linear regression method, mean error 2.12°, standard deviation 2.66°; (d) second-degree polynomial method, mean error 1.44°, standard deviation 1.80°; (e) shading with the inherited normal vector.

vector by different methods are given in terms of angles. The results show that the best quality image is achieved by our method and the central difference plus interpolation method gives the worst. For numerical accuracy analysis, interpolators versus approximators can be recognized from Refs. 5 and 15. However, in the interpolation process, the interpolating curve passes through all given data points, and these data may include noise, thus this method is noise sensitive. In practical application, volume data were obtained by sampling with finite bit-length converters, such as scanners and cameras. The finite bit-length property of volume acquisition devices, so-called quantization noise, affects the interpolation accuracy. In our experiment, the volume data are obtained by quantizing each sampled point with 256 gray levels. The discrepancies between the original value and quantized value introduce bias and interpolation method is sensitive to this bias. Therefore, poor results are produced. If the precision of volume data is double, the derivative of Catmull-Rom spline method gives much more accurate result (mean error=0.00307 deg, stdv =0.00317 deg). For the approximation method, a second-degree polynomial function is found to perform better than linear regression. This is in accordance with the fact that approximation accuracy is improved by increasing the order of the kernel. Furthermore, the voxel nearest to the terminated point should have the most significant effect on the normal vector estimation. However, in the linear regression introduced in the literature,[9] the weighting function of the terminated voxel is zero, so that errors are introduced.

Real-world signals usually contain departures from the ideal signal. Such departures are referred to as noise. One kind of noise that occurs in all recorded images is detector noise. This kind of noise is due to the nature of the radiation in discretization, i.e., the fact that imaging systems capture an image by counting photons. To evaluate each method's ability to attenuate noise, a white Gaussian noise model that has a zero mean was built according to Ref. 16 and then added to the volume data. In our experiments, we vary the variance (σ_n^2) of the noise distribution. Table 1 shows the effect of different methods.

To test our method in a highly fluctuating area, Marchner and Lobb data,[8] which are derived from the following function, were used:

$$f(x,y,z)=\frac{[1-\sin(\pi z/2)]+\beta[1+\rho_r(x^2+y^2)^{1/2}]}{2(1+\beta)},$$

where

$$\rho_r(r)=\cos\left[2\pi f_M \cos\left(\frac{\pi r}{2}\right)\right].$$

Volume data are sampled with a resolution of 200×200 ×200 in the range of $-1<x,y,z<1$, with $f_M=6$ and $\beta =0.25$. The rendering results are shown in Figs. 3(a) to 3(e). In this experiment, the quantization noise is much lower, so the accuracy of the interpolation method is satisfactory. Though there is no apparent difference among these result images, it is found that Fig. 3(a) is the best considering the mean error and standard deviation. The good performance of the central difference plus interpolation is consistent with that densely sampled data set increases this method's accuracy. This can be explained by

Table 1 Comparison of the accuracy of various method influenced by Gaussion noise using ball data.

	Method			
σ_n^2	Central Difference +Interpolation (deg)	Derivative of Catmull-Rom Spline (deg)	Linear Regression (deg)	Second-Degree Polynomial Approximation (deg)
0.5				
Mean error	38.88	30.69	12.06	8.73
Stdv	44.05	34.68	13.69	9.90
0.1				
Mean error	23.13	19.83	5.72	4.11
Stdv	26.42	22.80	6.48	4.65

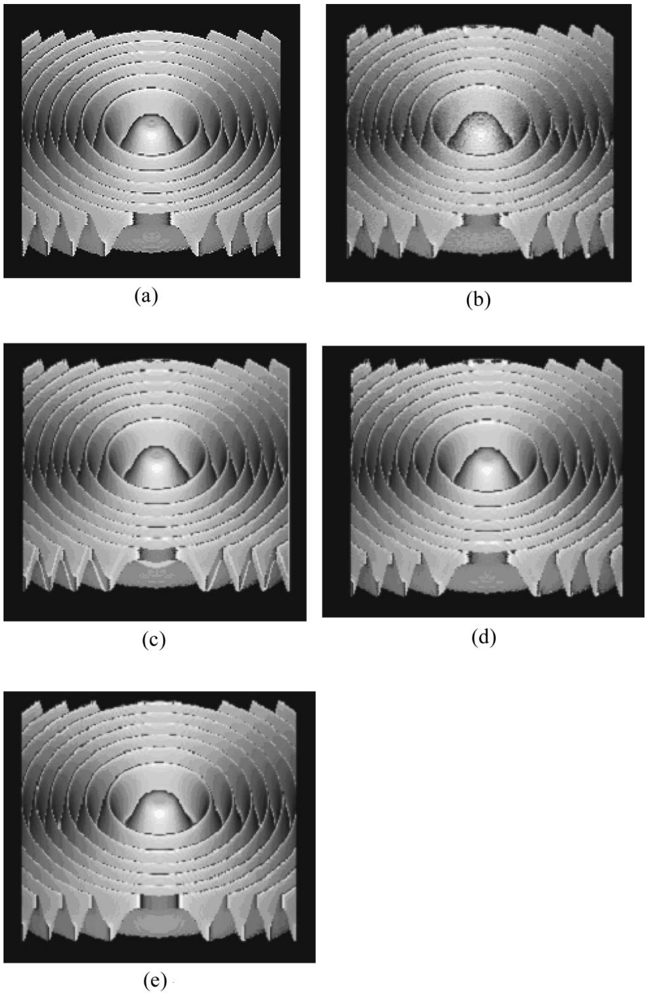

Fig. 3 Normal estimation on Marchner and Lobb data. (a) Central difference+interpolation method, mean error 1.38°, standard deviation 2.19°; (b) derivative of Catmull-Rom spline, mean error 4.74°, standard deviation 8.78°; (c) linear regression method, mean error 8.14°, standard deviation 17.91°; (d) second-degree polynomial method, mean error 2.86°, standard deviation 7.6°; (e) shading with the inherited normal vector.

the fact that the sampling rate determines the frequency spectrum replication. The higher the sampling rate, the wider the distance between the spectrum replication and the less will be the resulting signal overlap. Thus, the deficiencies of the central difference operator at higher frequencies do not impose a problem since no signal aliases exist in this frequency range.[2] As in Ref. 3, in frequency domain, while both the central difference+interpolation operator and the derivative of the Catmull-Rom spline show a good approximation of the ideal filter at a lower frequency, they show a rapid fall off at higher frequencies and have aliasing energy in the band at frequencies above π. The main disadvantage of the fall off at higher frequencies is attenuation of the signal's high-frequency components and the energy of the operator at frequencies above π cause artifacts and aliasing. The derivative of the Catmull-Rom spline has a slighter fall off at high frequencies than the central difference +interpolation, but have more energy above the foldover frequency π. Therefore, the performance of the Catmull-Rom derivative deteriorates when the signal includes high-frequency noise and has energy above frequency π. It is more sensitive to noise and has more artifacts, as shown in Fig. 3(b). For the approximation approach, it is not dc-constant and introduces additional smoothing. Though this is a disadvantage, it also has some desirable properties. Volume data often contain noise and unwanted energy, and attenuation of components above π in the frequency domain will reduce artifacts caused by noise and aliasing. This can be proved by results in Tables 1 and 2. From the preceding analysis, it can be concluded that although in Taylor series coefficient analysis, the method C for $\alpha = -0.5$ is the optimal derivative filter, it is not absolutely true. The aliasing and smoothing errors, which should be analyzed in the frequency domain, can not be examined by this method. Another point is that this method cannot be applied for rapidly changing signals and it is not suitable for analyzing the data in which the noise is contained.

We also tested these methods on the binary data sets. Figures 4(a) to 4(d) show the image of a neuron acquired by a laser-scanning confocal microscopy. After segmentation, binary volume data set is obtained. Binary data sets may include numerous high-frequency components, in this case, if interpolation method is utilized, artifacts will be introduced, as shown in Figs. 4(a) and 4(b).

In general, for approximation, high-degree polynomials have more flexibility than low-degree ones. For the same reason, the larger the neighborhood, the more control points and the more flexible the approximation polynomial.

Table 2 Comparison of the accuracy of various method influenced by Gaussion noise using Marchner and Lobb data.

	Method			
σ_n^2	Central Difference +Interpolation (deg)	Derivative of Catmull-Rom Spline (deg)	Linear Regression (deg)	Second-Degree Polynomial Approximation (deg)
0.001				
Mean error	44.94	46.42	27.03	23.49
Stdv	50.60	51.83	34.56	31.27
0.0001				
Mean error	25.85	26.91	13.84	11.99
Stdv	33.15	33.27	22.79	20.00

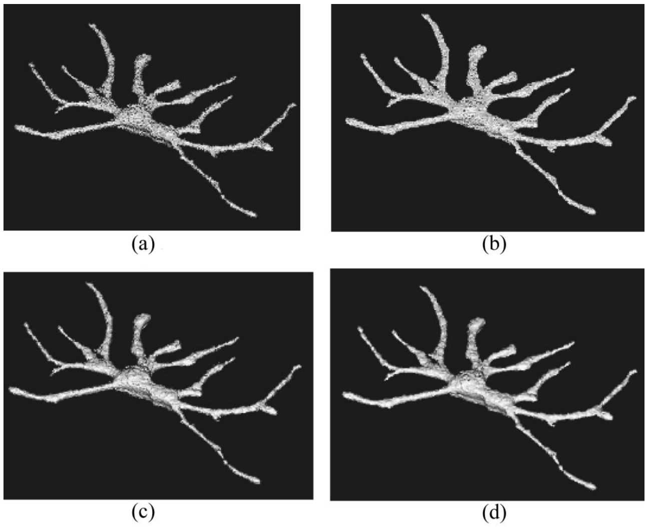

Fig. 4 Normal estimation on binary volume data about a neuron. (a) Central difference+interpolation method; (b) derivative of Catmull-Rom; (c) linear regression method; (d) second-degree polynomial method.

Hence, as the neighborhood size increases, the approximation polynomial becomes closer to the data set. In our experiment, the degree and the size of the neighborhood of the fitting polynomial were varied to investigate their impact on the estimation accuracy. From the preceding results, as expected, the performance of the second-degree polynomial is better than the linear regression. When the neighborhood is decreased from 4×4×4 to 3×3×3, the performance deteriorates whether the ball data or the Marchner and Lobb data are tested. Though further enlargement of the polynomial's order and the approximation neighborhood may improve the estimation quality, it also increases the computation complexity. Moreover, if the approximation neighborhood is too large, fine details such as bone fractures in medical images will be blurred.

4 Conclusion

A new approach to normal vector estimation was presented that is based on a second-degree polynomial function. In general, interpolators versus approximators are differentiable when ideal (noise-free) signals are dealt with. However, in practical applications, signals usually contain noise caused by a wide range sources, e.g., variations in the detector sensitivity, environmental variations or quantization errors, etc. From the frequency domain analysis, the derivative of the interpolation shows a rapid fall off at higher frequencies and has aliasing energy in the band at frequencies above π. These properties make the methods sensitive to noise. As demonstrated in the experiments, our approach is a fairly robust gradient estimation method for noise-containing data. Moreover, its performance in a binary data set is superior to that of other methods.

Acknowledgments

We wish to thank Dr. Hengyi Zhang for his encouragement on this paper and his enlightening advice. We also thank Dr. Caihua Shou for providing the neuron data. This work is supported by the National Nature Science Foundation of China, Grant No. 30170275, the Science and Technology Department of Zhejiang Province (No. 011106239), and the Key Laboratory for Biomedical Engineering of Ministry of Education of China.

References

1. R. Yagel, D. Cohen, and A. Kaufman, ''Normal estimation in 3D discrete space,'' *Visual Comput.* **8**(5), 278–291 (1992).
2. T. Moller, R. Machiraju, K. Mueller, and R. Yagel, ''A comparison of normal estimation schemes,'' in *Proc. IEEE Conf. on Visualization*, pp. 19–26 (1997).
3. M. J. Bentum, B. B. A. Lichtenbelt, and T. Malzbender, ''Frequency analysis of gradient estimators in volume rendering,'' *IEEE Trans. Vis. Comput. Graph.* **2**(3), 242–254 (1996).
4. T. Moller, R. Machiraju, K. Mueller, and R. Yagel, ''Evaluation and design of filters using a Taylor series expansion,'' *IEEE Trans. Vis. Comput. Graph.* **3**(2), 184–199 (1997).
5. T. M. Lehmann, C. Gonner, and K. Spitzer, ''Interpolation methods in medical image processing,'' *IEEE Trans. Med. Imaging* **18**(11), 1049–1075 (1999).
6. D. P. Mitchell and R. A. Schowengerdt, ''Image reconstruction by parametric cubic convolution,'' *Comput. Vis. Graph. Image Process.* **23**, 258–272 (1983).
7. J. A. Parker, R. V. Kenyon, and D. E. Troxel, ''Comparison of interpolating methods for image resampling,'' *IEEE Trans. Med. Imaging* **2**(1), 31–39 (1983).
8. S. R. Marschner and R. J. Lobb, ''An evaluation of reconstruction filters for volume rendering,'' in *Proc. Visualization '94*, pp. 100–107 (1994).
9. L. Neumann, B. Csebfalvi, A. Konig, and E. Groller, ''Gradient estimation in volume data using 4D linear regression,'' *Eurographics* **19**(3), 351–357 (2000).
10. D. P. Mitchell and A. N. Netravali, ''Reconstruction filters in computer graphics,'' *Comput. Graph.* **22**, 221–228 (1988).
11. M. Levoy, ''Display of surfaces from volume data,'' *IEEE Comput. Graphics Appl.* **8**(3), 29–37 (1988).
12. S. Parker, M. Parker, Y. Livnat, P. P. Sloan, and C. Hansen, ''Interactive ray tracing for volume visualization,'' *IEEE Trans. Vis. Comput. Graph.* **5**(3), 238–250 (1999).
13. R. Yagel, D. Cohen, and A. Kaufman, ''Discrete ray tracing,'' *IEEE Comput. Graphics Appl.* **12**(5), 19–28 (1992).
14. J. F. Blinn, ''Models of light reflection for computer synthesized pictures,'' *Comput. Graph.* **11**(2), 192–198 (1977).
15. R. W. Schafer and L. R. Rabiner, ''A digital signal processing approach to interpolation,'' *Proc. IEEE* **61**, 692–702 (1973).
16. J. L. Danger, A. Ghazel, and E. Boutillon, ''Efficient FPGA implementation of Gaussian noise generator for communication channel emulation,'' in *Proc. Electronics, Circuits and Systems, ICECS 2000, the 7th IEEE Int. Conf.* Vol. 1, pp. 366–399 (2000).

Dihui Hong received her BS degree in biomedical engineering in 2000 from Zhejiang University, China, where she is currently working toward her MS degree. Her current focus is on the visualization of confocal microscopy data and her research interests include image processing, especially biomedical image processing techniques.

Gangmin Ning received his BS and MS degrees in biomedical engineering from Zhejiang University, China, in 1987 and 1990, and his Dr Ing degree in biomedical engineering from the Technical University of Ilmenau, Germany, in 2001. He is currently with the Department of Biomedical Engineering, Zhejiang University, China. His research interests include signal processing, cardiovascular dynamics, and the optimum strategy for the treatment of cardiovascular diseases.

Ting Zhao received his BS and MS degress in biomedical engineering from Zhejiang University, China, in 1999 and 2002, respectively. Currently he is a PhD student in the Department of Biomedical Engineering at Carnegie Mellon University. His current research interest is focused on automatic interpretation of microscope images.

Mu Zhang received his BS and MS degrees in biology from Zhejiang University, China, in 1995 and 1998, respectively. Currently he is a PhD student in the Department of Biomedical Engineering at Zhejiang University. His current research interest is focused on neuron signal transduction.

Xiaoxiang Zheng received her BS degree in electronic engineering from Zhejiang University, China, in 1968 and her MS and Dr degrees in medicine from the University of Tsukuba, Japan, in 1984 and 1993, respectively. She is currently a professor and dean of the Department of Biomedical Engineering, Zhejiang University. Her research interests include modeling and simulation in physiology, medical image processing, and techniques in pharmacological screening.