Three Dimensional Border Identification

Xiaoqing Qu and Wayne Davis
Department of Computing Science
University of Alberta
Edmonton Canada T6G 2H1

Abstract

This paper presents a three dimensional border identification method based on the second derivative of intensity change. For a bright object surrounded by a darker background, the condition for a voxel being on a border is that the second derivative is negative and changes sign for neighbors in the gradient direction. To compute the second derivative, a 3D edge operator is applied to compute the gradient. The gradient directions are quantized to 26 vectors. Asymmetric Gaussian filters for 3D convolutions are designed for the 26 gradient vectors. Hence the computation of 3D convolution can be speeded up. Although the quantization of gradient directions to 26 vectors is not smooth, analysis shows that the condition for identifying border voxels is not sensitive to quantization errors. Moreover, there exists exactly one layer of voxels that satisfies the condition. This avoids a multi-layer problem, hence tracking of border voxels can simply be a breadth first search. An experimental result of 3D surface identification, tracking, and display on real medical data is given.

1 Introduction

This paper presents a three dimensional border identification method based on the second derivative of intensity change.

In 3D identification and display, thresholding is widely used. Algorithms based on thresholding includes 3D surface tracking [AFH81, GU89], volume rendering [Rey87], marching cubes [LC87].

In a variety of applications, however, many objects can not be identified by simple thresholding. Identification based on 3D edge detection could be a more general method. Some 3D edge operators have been proposed to detect edge elements [MR81] using gradients. Those edges with larger gradient magnitudes are possibly on a boundary.

A problem with gradients is that there is a flat area around the peak of the gradient. The edge elements whose magnitudes are greater than a given threshold are legal boundary edge candidates. Moreover, both the edge operator and the magnitude calculation introduce errors. It is difficult to locate the edges with real maximum gradient magnitudes. This causes a multiple layer problem in 3D border identification.

The 3D boundary following algorithm by [CR89] constructs a 3D boundary by stacking 2D boundaries, that are extracted using graph search algorithm [Mar76]. It is basically a 2D method.

Since the directional second derivative of an intensity change has a steep slope around a zero-crossing, that corresponds precisely to the peak of the gradient. Marr and Hildreth [MH80] suggested that zero-crossings can be used to detect edge elements.

In this paper, a border identification method based on the sign of the second derivative of an intensity change is proposed. Asymmetric Gaussian filters are designed to compute the second derivative. For a bright object surrounded by a darker background, the condition for a voxel being on a border is that the second derivative is negative and changes sign for neighbors in the gradient direction. From these conditions, there exists exactly one layer of border voxels. There is no multiple layer problem. Border voxel tracking can simply be a breadth first search. Comparing with heuristic search, the border search space is greatly reduced.

The analysis is first conducted in continuous space. The definition of border voxels is then given in discrete space.

2 Surface of Step Intensity Change

This section will show that the surface points for a step intensity change are zero-crossings.

Let $I(v)$ be a continuous intensity function, $v = (x, y, z) \epsilon R^3$, and R is the set of real numbers. Suppose there is a step intensity change across the object surface, Without loss of generality, suppose the intensity change occurs at the origin and in the x direction. Clearly, the origin is the desired surface point. It is necessary to find out what property the surface points possess so that the surface can be identified.

For a bright object surrounded by a darker background, the intensity around the surface can be modeled as a one dimensional step change, $I(x, y, z) = c(\pi/2 - \arctan(kx))$ for a sufficiently large constant $k > 0$. The constant c describes the change magnitude and k describes the intensity change rate. The smoothing filter $G(x, y, z)$ is a Gaussian filter with variance σ. Let $w(x, y, z)$ be the second derivative in the x-direction of the smoothed function $I(x, y, z)$. $w(x, y, z)$ is given by

$$\frac{\partial^2}{\partial x^2}[G(x, y, z) * I(x, y, z)], \tag{1}$$

Evaluate the second derivative with respect to $I(x, y, z)$ results in

$$w(x) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{x'^2}{2\sigma^2}} \frac{2ck^3(x - x')}{(1 + k^2(x - x')^2)^2}dx' \tag{2}$$

which is a function of x. Because the integrand is an odd function, $w(x) = 0$ at $x = 0$. To see how $w(x)$ changes sign cross $x = 0$, it is necessary to evaluate the sign of $w'(x)$ at $x = 0$. Note that the difference between the derivative of $x - x'$ with respect to x and x' is minus one, hence

$$w'(x) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{x'^2}{2\sigma^2}} \frac{d}{dx'}\left(\frac{-2ck^3(x - x')}{(1 + k^2(x - x')^2)^2}\right) dx'.$$

Integrate the right hand side of above equation by parts gives

$$w'(x) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} \left(-\frac{x'}{\sigma^2}\right)e^{-\frac{x'^2}{2\sigma^2}} \frac{2ck^3(x - x')}{(1 + k^2(x - x')^2)^2}\, dx'$$

and at $x = 0$,

$$w'(0) = \frac{2ck^3}{\sqrt{2\pi}\sigma^3}\int_{-\infty}^{\infty} \frac{x'^2 e^{-\frac{x'^2}{2\sigma^2}}}{(1 + k^2x'^2)^2}dx' > 0. \tag{3}$$

Hence $w(x)$ is monotonically increasing in the neighborhood of $x = 0$. In other words, $w(x)$ changes sign from negative to positive as it crosses zero at $x = 0$. $x = 0$ is therefore called a zero-crossing and the properties of zero-crossings can be used as conditions to detect surface points.

It is worthwhile to point out that the location of zero-crossings is not sensitive to the direction of the second derivative of an intensity change. Suppose the second derivative is taken in the direction l. The direction cosines of l are $\cos\alpha$, $\cos\beta$ and $\cos\theta$ for some $\alpha, \beta, \theta, < \pi/2$. (2) then becomes

$$w(x) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{x'^2}{2\sigma^2}} \frac{2ck^3(x - x')\cos^2\alpha}{(1 + k^2(x - x')^2)^2}dx'.$$

Because $\cos^2\alpha > 0$, the direction of l does not affect the zero location nor the sign of $w(x)$ but only the magnitude of $w(x)$.

To summarize, let $v = (x, y, z) \in R^3$ and l be a directed line at v pointing outside of the surface. If v is on the surface, it must satisfy the following conditions:

$$\frac{\partial^2}{\partial l^2}[G(v) * I(v)] = 0 \tag{4}$$

and

$$\frac{\partial^2}{\partial l^2}[G(v - \delta l) * I(v - \delta l)] < 0, \quad \frac{\partial^2}{\partial l^2}[G(v + \delta l) * I(v + \delta l)] > 0 \tag{5}$$

for a small $\delta l > 0$ in the direction of l. Alternatively, the conditions above can be written in terms of $w(v)$ as

$$\begin{aligned} w(v) &= 0 \\ w(v - \delta l) &< 0 \\ w(v + \delta l) &> 0. \end{aligned} \tag{6}$$

3 The Direction of The Second Derivative

As pointed out in the previous section, the direction of the second derivative does not affect the location of zero-crossings. But due to the quantization errors and noise, it is desired to take the second derivative in such a direction that has a maximum rate of change. This section shows the direction in which the second derivative of the intensity function has maximum change rate.

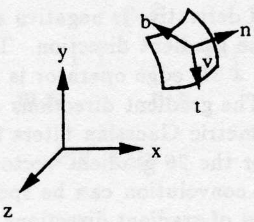

Figure 1: Coordinate system $n\ t\ b$ origined at v.

Suppose the gradient at v is $g(v)$ with the direction g. Take v as the origin and form a right-handed orthogonal coordinate system $\{v; n, t, b\}$ such that n is in the same direction as g, as shown in Figure 1. Let l be a directed line, and $\cos\alpha$, $\cos\beta$ and $\cos\gamma$ be the direction cosines of l respect to the (n, t, b)-coordinate system.

The derivative of the intensity function $I(n, t, b)$ in the direction of l is given by

$$\frac{\partial}{\partial l}I(n, t, b) = \left(\frac{\partial}{\partial n}\cos\alpha + \frac{\partial}{\partial t}\cos\beta + \frac{\partial}{\partial b}\cos\gamma\right) I(n, t, b). \tag{7}$$

At the origin v, because n is in the gradient direction, $\frac{\partial I}{\partial t}|_{(0,0,0)} = 0$ and $\frac{\partial I}{\partial b}|_{(0,0,0)} = 0$. Therefore,

$$\frac{\partial}{\partial l}I(0, 0, 0) = \frac{\partial}{\partial n}I(0, 0, 0)\ \cos\alpha. \tag{8}$$

If $\alpha = 0$, $\frac{\partial}{\partial l}I(0, 0, 0)$ takes its extreme value $\frac{\partial}{\partial n}I(0, 0, 0)$, the gradient value of v.

The second derivative of $I(n, t, b)$ in the direction l is given by

$$\begin{aligned} \frac{\partial^2}{\partial l^2}I(n, t, b) = (&\frac{\partial^2}{\partial n^2}\cos^2\alpha + \frac{\partial^2}{\partial n \partial t}2\cos\alpha\cos\beta + \\ &\frac{\partial^2}{\partial n \partial b}2\cos\alpha\cos\gamma + \frac{\partial^2}{\partial t \partial b}2\cos\beta\cos\gamma + \\ &\frac{\partial^2}{\partial t^2}\cos^2\beta + \frac{\partial^2}{\partial b^2}\cos^2\gamma)\ I(n, t, b). \end{aligned} \tag{9}$$

For a fixed l, $\frac{\partial^2}{\partial l^2} I(n,t,b)$ changes most rapidly in its gradient direction, that is the direction of $(\frac{\partial}{\partial n}\vec{n} + \frac{\partial}{\partial t}\vec{t} + \frac{\partial}{\partial b}\vec{b})\frac{\partial^2}{\partial l^2} I(n,t,b)$. Because at origin v, $\frac{\partial I}{\partial t}|_{(0,0,0)} = 0$ and $\frac{\partial I}{\partial b}|_{(0,0,0)} = 0$, to make the calculation simpler, an approximation is made that in a small neighborhood V of v, $\frac{\partial I}{\partial t}|_{v\epsilon V} = 0$ and $\frac{\partial I}{\partial b}|_{v\epsilon V} = 0$. Hence equation (9) is reduced to

$$\frac{\partial^2}{\partial l^2} I(n,t,b) = \frac{\partial I^2}{\partial n^2}\cos^2\alpha. \qquad (10)$$

For a given direction l in V, the gradient of $\frac{\partial^2}{\partial l^2} I(n,t,b)$ is then given by

$$\frac{\partial I^3}{\partial n^3}\cos^2\alpha\vec{n} + \frac{\partial I^3}{\partial n^2 \partial t}\cos^2\alpha\vec{t} + \frac{\partial I^3}{\partial n^2 \partial b}\cos^2\alpha\vec{b}$$
$$= \frac{\partial I^3}{\partial n^3}\cos^2\alpha\vec{n}. \qquad (11)$$

Equation (11) shows that for any given l, if at every point $(n,t,b)\epsilon V$ the second derivative is taken in the direction of l, the resulting $\frac{\partial^2}{\partial l^2} I(n,t,b)$ always changes rapidly in the direction of $\vec{n}$. Especially when $\alpha = 0$, l lies in the direction of $\vec{n}$, $\frac{\partial^2}{\partial l^2} I(n,t,b) = \frac{\partial^2}{\partial n^2} I(n,t,b)$ and changes most rapidly with the rate $\frac{\partial I^3}{\partial n^3}$. Consequently, the direction in which the second derivative of v should be taken is the gradient direction of v. Conditions (4) and (5) are then written as

$$\frac{\partial^2}{\partial g^2}[G(v) * I(v)] = 0, \qquad (12)$$

and

$$\frac{\partial^2}{\partial g^2}[G(v-\delta g)*I(v-\delta g)] < 0, \ \frac{\partial^2}{\partial g^2}[G(v+\delta g)*I(v+\delta g)] > 0, \qquad (13)$$

where $\delta g > 0$ and g is in the gradient direction of $I(v)$ at v.

By the symmetric property of convolution, the second derivative in condition (12) can be taken with either $G(v)$ or $I(v)$. The left-hand side of (12) can therefore be written as $G(v) * \frac{\partial^2}{\partial g^2} I(v)$ or $\frac{\partial^2}{\partial g^2} G(v) * I(v)$. Because the intensity function $I(v)$ is usually unknown, the second derivative is taken with the Gaussian filter $G(v)$. Denote $\frac{\partial^2}{\partial g^2} G(v)$ by G_g'', the above equations become:

$$G_g''(v) * I(v) = 0, \qquad (14)$$

$$G_g''(v-\delta g)*I(v-\delta g) < 0, \ \ G_g''(v+\delta g)*I(v+\delta g) > 0. \qquad (15)$$

Write above equations in terms of $w(v)$, $w(v) = G_g''(v) * I(v)$, (6) becomes:

$$\begin{aligned} w(v) &= 0 \\ w(v-\delta g) &< 0 \\ w(v+\delta g) &> 0. \end{aligned} \qquad (16)$$

Since the conditions (14) and (15) involve three dimensional convolutions, the computation is very expensive. The next section will show how to simplify the computation.

4 Convolutions for the General Case

If the intensity function $I(v)$ is separable in the $\{v; n, t, b\}$ coordinate system, the 3D convolution in conditions (14) and (15) can be replaced by a 1D convolution. In general, however, $I(n,t,b)$ is not separable. But it will soon be seen that the symmetric Gaussian filter in the 3D convolution can be replaced by an asymmetric one with a smaller scale in the t, b axes. As a result, conditions (14) and (15) can be tested with a reasonable expense.

Transform (x,y,z) to the $\{v; n, t, b\}$ coordinate system, the left-hand side of condition (14) becomes

$$\tilde{w}(n,t,b) = G_n''(n,t,b) * I(n,t,b). \qquad (17)$$

First, a Gaussian filter is localized. It approximates to zero when its argument are out of the range $(-3\sigma, 3\sigma)$. Hence finite scale Gaussian filters can be used in convolutions. The integral interval in (17) therefore takes $(-3\sigma, 3\sigma)$ instead of $(-\infty, \infty)$.

Secondly, let $\tilde{I}(n', t, b)$ be the result of the 2D integration respect to t', b' in (17),

$$\tilde{I}(n', t, b) = G(t,b) * I(n,t,b) \qquad (18)$$

where $G(t,b)$ is the two dimensional Gaussian filter in t, b direction, then (17) becomes

$$\tilde{w}(n,t,b) = \int_{-3\sigma}^{3\sigma} \frac{\partial^2}{\partial n^2}\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(n-n')^2}{2\sigma^2}}\right) \tilde{I}(n', t, b) dn'$$

which is

$$\begin{aligned} \tilde{w}(n,t,b) &= \frac{\partial^2}{\partial n^2} G(n) * \tilde{I}(n,t,b) \\ &= G_n''(n) * G(t,b) * I(n,t,b). \end{aligned}$$

Note from (18) that $\tilde{I}(n,b,t)$ is a smoothed version of $I(n,b,t)$ in planes perpendicular to the n axis. Remember that $\vec{n}$ is the gradient direction at v, in whichever direction in the tb plane the intensity change rate at v is zero. It is reasonable to assume that in a small neighborhood of v, $I(n,t,b)$ does not significantly change in the tb direction. Hence the 2D Gaussian filter $G(t,b)$ with a smaller σ can be used in (18). This suggests the use of an asymmetric Gaussian filter with different scales, σ_1 in n axis and σ_2 in b, t axes and $\sigma_1 > \sigma_2$, i.e.

$$G(n,t,b) = \frac{1}{(\sqrt{2\pi})^3 \sigma_1 \sigma_2^2} e^{-\frac{n^2}{2\sigma_1^2} - \frac{t^2+b^2}{2\sigma_2^2}},$$

to smooth the intensity $I(n,t,b)$ at v. Denote $\frac{\partial^2}{\partial n^2} G(n,t,b)$ by $G_n''(n,t,b)$, where

$$G_n''(n,t,b) = \frac{1}{(\sqrt{2\pi})^3 \sigma_1 \sigma_2^2}\left(1 - \frac{n^2}{\sigma_1^2}\right) e^{-\frac{n^2}{2\sigma_1^2} - \frac{t^2+b^2}{2\sigma_2^2}}. \qquad (19)$$

σ_2 can take a rather small value, and the size of an asymmetric Gaussian filter is smaller than a symmetric one. As a result the computation expense can be reduced.

5 Border in Discrete Space

Because the data to be processed are defined on integer points of the coordinate system where they were collected, conditions (14) and (15) for identifying a surface in a continuous space must be adequately adapted to a discrete space. This section introduces conditions to identify border voxels in a discrete space.

Let $I(v) \epsilon N^3$, $N = \{0, 1, 2, \ldots\}$, be an intensity function defined on a discrete domain, $v \epsilon Z^3$, $Z = \{0, \pm 1, \pm 2, \ldots\}$. $v = (x, y, z)$ is called a voxel. Let $G_g''(x, y, z)$ be a finite scale discrete Gaussian filter, resulting from sampling $G_g''(v)$ given in (14) in a finite interval $(-3\sigma, 3\sigma)$, and $w(x, y, z)$ be the result of the discrete convolution of $G_g''(x, y, z)$ with $I(x, y, z)$,

$$w(x, y, z) = G_g''(x, y, z) * I(x, y, z). \quad (20)$$

Transform the coordinate system $\{0; x, y, z\}$ to $\{v; n, t, b\}$ and express the above convolution in terms of (n, t, b) as

$$\tilde{w}(n, t, b) = \sum_i \sum_j \sum_k G_n''(i, j, k) I(n-i, t-j, b-k), \quad (21)$$

where $G_n''(n, t, b)$ is a finite describe filter from sampling the asymmetric Gaussian filter in (19). At the origin v,

$$\tilde{w}(0, 0, 0) = w(x, y, z).$$

The $w(x, y, z)$ in (20) is defined on voxels of integer coordinates and it is undefined between voxels. Conditions (16) in Section 3, however, infers that $w(x, y, z)$ is negative inside the object and positive outside. There exists exactly one layer of voxels on which $w(x, y, z)$ is negative and changes sign for neighbors in the g-direction. This layer is called the negative layer. Similarly, there exists exactly one positive layer of voxels. It is possible to define either the negative layer or the positive layer as the border. As the negative layer is inside the object, the border is defined as the negative layer of voxels. The zero-crossing voxels can be treated as either positive or negative, and are also included in the border set. Condition (14) therefore becomes $w(x, y, z) \leq 0$.

To adapt condition (15), it is necessary to determine the neighbor voxel set on which the condition is tested.

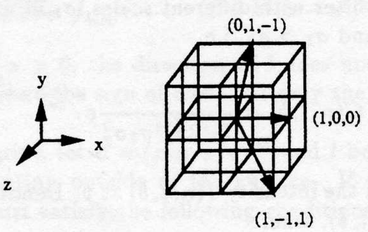

Figure 2: Three of the 26 gradient directions

The gradient of $I(v)$ at v is approximated by $\nabla_v = (\nabla_x, \nabla_y, \nabla_z)$ and is quantized to 26 directions, see Figure 2. In each of the directions, voxels are either face connected, edge connected, or vertex connected. If $\nabla_v = (1, 0, 0)$ and $v = (x, y, z)$ is on the border (the negative layer), the voxel $(x+1, y, z)$ is on the positive layer, $w(x+1, y, z) > 0$, and the voxel $(x-1, y, z)$ is inside, $w(x-1, y, z) < 0$. For $\nabla_v = (1, 1, 0)$, however, if v is on the border, the voxel $(x+1, y+1, z)$ is not on the positive layer but $(x+1, y, z)$ and $(x, y+1, z)$ are, as shown in Figure 3. Therefore it is necessary to verify that $w(x-1, y, z) < 0$, $w(x, y-1, z) < 0$, $w(x+1, y, z) > 0$, and $w(x, y+1, z) > 0$. These cases reveal that the condition (15) should be tested on face connected neighbor voxels for every non-zero component of ∇_v.

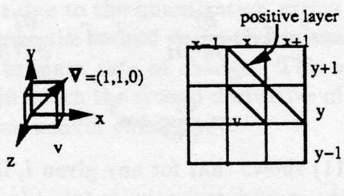

Figure 3: If the gradient components, ∇_x, ∇_y, of a border voxel v are not zero, $w(v)$ has to change sign in both x, y directions.

To summarize, for a bright object surrounded by a darker background, if $v(x, y, z)$ is a border voxel, it must simultaneously satisfy the following inequalities:

$$\begin{array}{lll} w(x, y, z) \leq 0, & & \\ w(x-1, y, z) < 0, & w(x+1, y, z) > 0, & \text{if } \nabla_x \neq 0, \\ w(x, y-1, z) < 0, & w(x, y+1, z) > 0, & \text{if } \nabla_y \neq 0, \\ w(x, y, z-1) < 0, & w(x, y, z+1) > 0, & \text{if } \nabla_z \neq 0. \end{array} \quad (22)$$

Similarly, for a dark object surrounded by a brighter background, the border voxels can be defined as the positive layer of voxels.

Inequalities (22) are the conditions to identify border voxels in a discrete space. Obviously, there is only one layer of voxels that will satisfy the conditions. This makes subsequent surface tracking easy. Computation of $w(x, y, z)$ in a discrete space will be discussed in the next section.

6 The Design of the Discrete Filter

To compute $w(x, y, z)$ given in (21), the $\{v; n, t, b\}$ coordinate system and asymmetric Gaussian filters used in a discrete space need to be specified. This section gives the setting of $\{v; n, t, b\}$ coordinate system at a voxel and the structures of discrete asymmetric Gaussian filters in the implementation.

To compute $w(x, y, z)$, a $\{v; n, t, b\}$ coordinate system is established at every voxel v. Figures 4(a) to 6(a) show the three $\{v; n, t, b\}$ for three voxels with gradient vectors $(1, 0, 0)$, $(0, 1, -1)$ and $(-1, 1, -1)$. The $\{v; n, t, b\}$ for other gradient vectors are just 90 degree rotations of the three. The axis $\vec{n}$ at v is always set in the gradient direction of v, so the system $\{v; n, t, b\}$ changes from voxel to voxel.

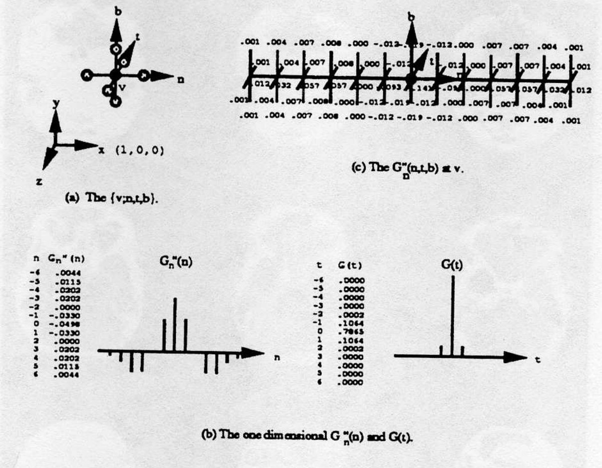

Figure 4: The $\{v; n, t, b\}$ system and the G''_n filter for the gradient vector $(1, 0, 0)$.

Since G''_n is separable, $G''_n(n,t,b) = G''_n(n)G(t)G(b)$, the scale, or length, of G''_n in each dimension is the length of the corresponding one dimensional filter. The length of the one dimensional $G''_n(n)$ is determined by variance $\sigma 1$, the length of one dimensional $G(t)$ and $G(b)$ is determined by variance $\sigma 2$, $\sigma 1 > \sigma 2$. Figures 4(b) to 6(b) depict $G''_n(n)$ of $\sigma 1 = 2.0$ and $G(t)$ of $\sigma 2 = 0.5$ for the three gradient vectors. The unit on the n or t axis is the distance between pair of voxels along the axis. For the gradient vector $(1,0,0)$ shown in Figure 4, $G''_n(n)$ spans approximately 13 voxels, $G(t)$ and $G(b)$ spans approximately 3 voxels. For the gradient vector $(0,1,-1)$, however, the distance between pairs of voxel on the n axis is longer than that for $(1,0,0)$, so the $G''_n(n)$ in Figure 5(b) is narrower, and the $G''_n(n)$ in Figure 6(b)is even narrower. Consequently, the sum of the discrete convolution in (21) is from -6 to 6 for i, -1 to 1 for j and k.

Figures 4(c) to 6(c) show the three discrete asymmetric $G''_n(n,t,b)$ filters of scales $\sigma_1 = 2.0$ and $\sigma_2 = 0.5$ for the three gradient vectors. The line segments in the Figures represent connections between voxels. There is a voxel located in every line intersection, join and end point. A weight, $G''_n(i,j,k)$, is labeled beside every such point.

The $G''_n(n,t,b)$ in Figure 4(c) spans 6 face connected voxels each side along n axis, one voxel each side along t and b axes. So the filter size is 65.

The $G''_n(n,t,b)$ in Figure 5(c) spans 6 edge connected voxels each side along n axis, one voxel each side along t axis, and one voxel each side along b axis. The filter size is 63.

The $G''_n(n,t,b)$ in Figure 6(c) spans 6 vertex connected voxels each side along n axis, 3 voxels in tb plane, and totally 52 voxels.

7 The Experimental Results

A surface tracking algorithm using conditions 22 to identify border voxels is given in [QD92]. Figure 7 shows the

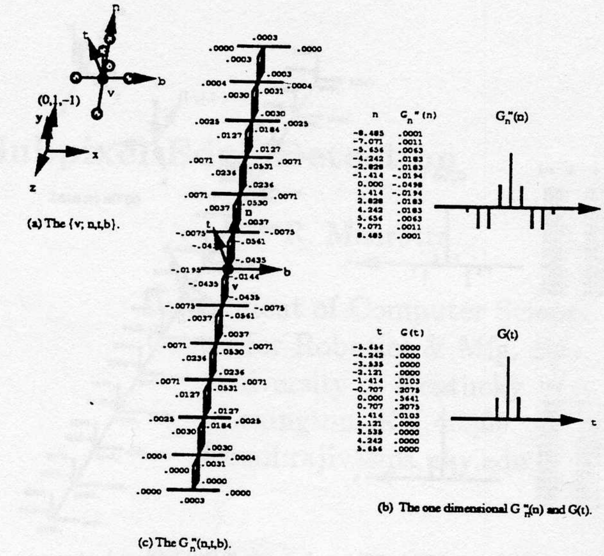

Figure 5: The $\{v; n, t, b\}$ system and the G''_n filter for the gradient vector $(0, 1, -1)$.

surface of a medical object from real data. The data size of the medical object is of $208 \times 208 \times 79$, resulting from linearly interpolating 14 CT slices producing 5 between successive pairs. The gray values were also linerly mapped to the range from 0 to 255. The object has brighter color than background, therefore, its border is the negative layer of voxels. There are 84708 border voxels detected, and total 132022 voxel faces on the surface. To display the surface, the border voxels are converted to the extended cuberille model (see [QD92]), where four types of voxels are defined to reflect the 26 gradient vectors. The image is displayed on Silicon Graphics Iris station 4D/35.

8 Conclusion

This paper presents a three dimensional border identification method based on the second derivative of intensity change. The conditions to identify border voxels are given in page 4. For a bright object surrounded by a darker background, if a voxel is on a border, it must satisfy condition (22) so that the second derivative at the voxel is negative and changes sign for neighbors in the gradient direction. To compute the second derivatives, a 3D edge operator is applied to compute gradients. The gradient directions are quantized to 26 vectors. Asymmetric Gaussian filters for 3D convolutions are designed for the 26 gradient vectors. The long scale of a Gaussian filter coincides with a gradient vector, with short scales in the perpendicular directions. Hence the computation of the 3D convolution can be speeded up. Although the quantization of gradient directions to 26 gradient vectors is not smooth, analysis shows that the condition for identifying border voxels is not sensitive to the quantization errors. Moreover, there exists exactly one layer of voxels that satisfies the condi-

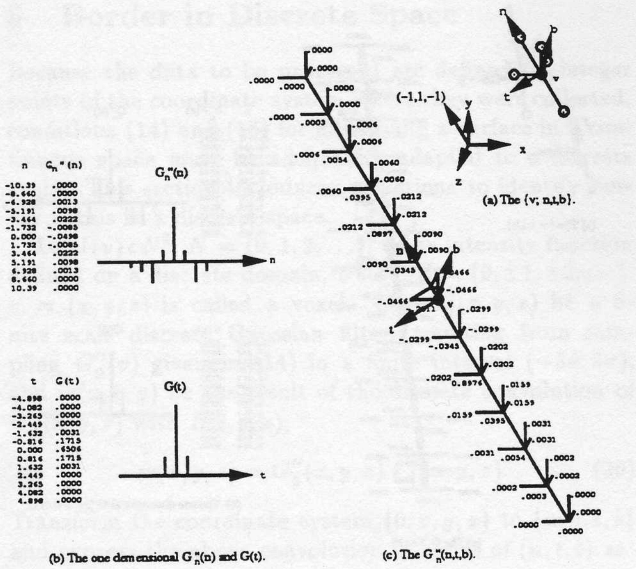

Figure 6: The $\{v; n, t, b\}$ system and the G''_n filter for the gradient vector $(-1, 1, -1)$.

tion. This avoids the multiple layer problem, hence the border tracking algorithm can be as simple as a breadth first search. An experimental result of 3D surface identification, tracking, and display from real medical data is given.

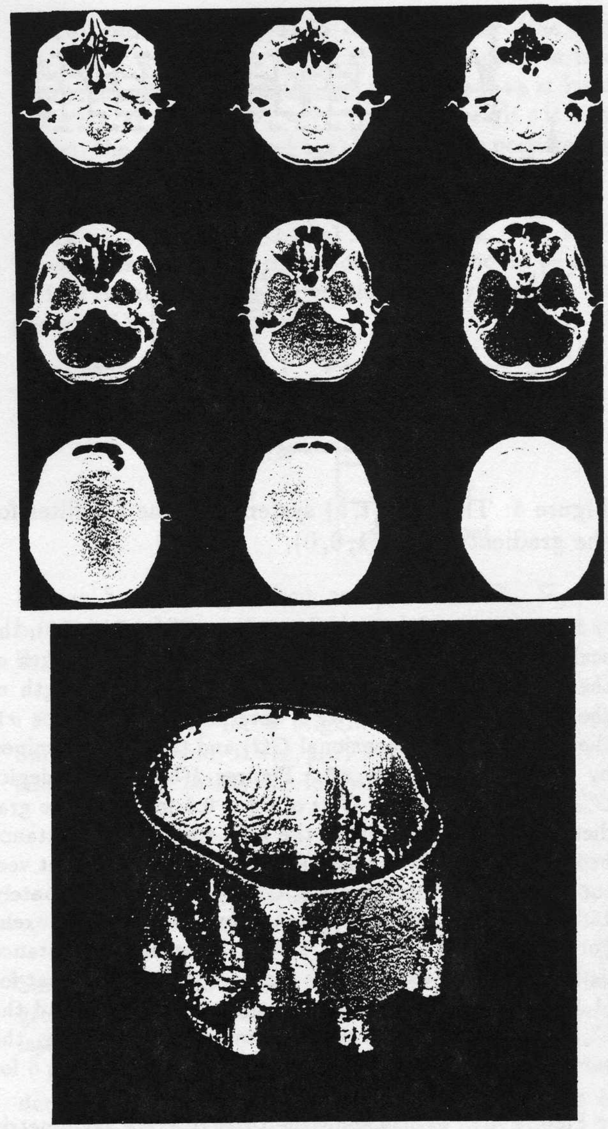

Figure 7: The data slices from real data and the surface of a medical object.

References

[AFH81] E. Artzy, G. Frieder, and G. T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Vision, Graphics, and Image Processing*, 15:1–24, 1981.

[CR89] John Danilo Cappelletti and Azriel Rosenfeld. Three-dimensional boundary following. *Computer Vision, Graphics, and Image Processing*, 48:80–92, 1989.

[GU89] D. Gordon and J. K. Udupa. Fast surface tracking in three-dimensional binary images. *Computer Vision, Graphics, and Image Processing*, 45:196–214, 1989.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *ACM Computer Graphics*, 21(4):163–169, 1987.

[Mar76] Alberto Martelli. An application of heuristic search methods to edge and contour detection. *Communication of the ACM*, 19(2):73–83, February 1976.

[MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proceeding of the Royol Society of London*, 207:187–217, 1980.

[MR81] D. G. Morgenthaler and A. Rosenfeld. Multidimensional edge detection by hypersurface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(4):187–217, July 1981.

[QD92] X. Qu and W. A. Davis. An extended cuberille model for identification and display of 3d objects from 3d gray value data. *Proceedings of Graphics Interface'92*, page to be appear, 1992.

[Rey87] R. A. Reynolds. A dynamic screen technique for shaded graphics display of slice-represented objects. *Computer Vision, Graphics, and Image Processing*, 38:275–298, 1987.