

# Case Study: Cellar Scaffold Extraction Using Crest Point for Volume Rendering

Jiuxiang Hu<sup>1</sup> †, Page D. Baluch<sup>3</sup>, Anshuman Razdan<sup>1</sup>, Gregory M. Nielson<sup>2</sup>, Gerald E. Farin<sup>2</sup> and David G. Capco<sup>3</sup>

<sup>1</sup> Partnership for Research in Stereo Modeling (PRISM), <sup>2</sup> Dept. Computer Science, and <sup>3</sup> Dept. Biology  
Arizona State University (ASU), Tempe AZ 85287, USA.

---

## Abstract

*Extraction of scaffolds, such as the meiotic spindles, a 3D tubular framework consisting of the microtubules, from confocal laser scanning microscopy (CLSM) data of a cell is a challenge in biological image processing. It is of major importance in the research of microtubule anchor proteins, and molecular motor mechanics. However, the scaffold is hidden within CLSM data due to the nature of light excitation, and is difficult to visualize using traditional opacity and color transfer functions that depend only on local intensity. In this paper, we treat 3D CLSM data as a hyper-surface in  $\mathcal{R}^4$ , and show that the crest points of the hyper-surface correspond to the centerline of the hidden scaffold. We propose an automatic approach to extract the hidden scaffold from CLSM data. First, the spindle from the large data set is segmented using Weibull E-SD fields. We, next, apply the Savitzky-Golay (S-G) filter and Gaussian convolution to reduce the noise in the data and calculate the first and second derivatives. Lastly, direct volume rendering using ray casting is applied to visualize the volume data. We combine the local intensity and maximum curvature information to decide the opacity transfer function. Promising results are shown on simulated data sets as well as real CLSM data of mouse egg.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, J.3.1 Computer Applications Biology and genetics

---

## 1. Introduction

Cellular and structural biologists have used confocal laser scanning microscopy (CLSM) to visualize various targets within cells. This visualization helps not only recognize structural patterns within the cell but can be used to evaluate the location of signaling elements associated with the microfilaments. One of the most widely studied structures is the cytoskeleton that is proposed to function as a scaffold assisting in the spatial and temporal positioning of elements within the cell<sup>3,4</sup>. The concept of scaffolds assisting in signaling pathways is becoming widely accepted through the research of microtubule anchor proteins (MAPs), ARPs (actin related proteins) and molecular motor mechanics. Although CLSM can be very specific in locating species features within cells and tissue samples, the nature of light excitation involves scattered light creating a blurred image of

smaller features within the image. Features labeled within the image must be spaced greater than  $2\ \mu\text{m}$  to be visualized as separate features within a confocal image. Features that have less than  $2\ \mu\text{m}$  spacing appear as a single feature without specific detail. An example of this type of image distortion can be seen in Figure 3 in which the spindle scaffold is hidden within CLSM data appearing only as a intense region of color without structural definition.

The spindle scaffold is a thin 3D framework of the microtubules which are cytoskeletal elements that create a protective cage around the DNA in mouse eggs undergoing division. The thin framework is one of the most important geometric features in biological images and, at present, the common method of extraction is based on the classification of the local structures. Sato<sup>9</sup> and Frangi<sup>10</sup> independently employ eigenvalues of the Hessian matrix at each voxel to design a filter for vessel enhancement in 3D medical digital images. Two years later Sato et al.<sup>8</sup> generalize the previously introduced concept to enhance tubular, blob, and sheet-like

---

† Corresponding author, e-mail:hu.jiuxiang@asu.edu

structures. Tubular object centerline extraction is based on dynamic-scale intensity ridge traversal<sup>11</sup>. The definition of maximum convexity height ridge depends on the eigenvalues and eigenvectors of the Hessian matrix at each vertex. This approach, however, uses the concept of a multifeature thin framework extraction, and leaves the user to set the 5 parameters. When two features are too close, it is difficult to select a proper set of parameters to extract them individually.

Monga<sup>2</sup> and Prinet<sup>1</sup> treated the 3D volume data as a hyper-surface in  $\mathcal{R}^4$ , and showed that the crest points of this hyper-surface correspond to the centerline of the vascular framework in 3D images. They next applied maximum curvatures and direction at each vertex to extract a thin network from the 3D images. This method to extract tubular objects does not require any specific parameters and is easy to implement automatically. However, it requires the data to be free of noise, because it uses the third derivative to extract a crest point.

In this paper, we propose an automatic approach to scaffold extraction using crest point for volume rendering. The basic idea is to extract each microtubule based not only on its intensity values, but also its locally largest curvatures at each vertex of the volume data. However, the rapid increase in data set sizes which is required to collect image data around the spindle apparatus as well as the poor signal to noise ratio in the data set, make it difficult to compute the curvatures at vertices of the volume data efficiently and accurately. Therefore, we first segment the spindle from the large data set. Statistical segmentation<sup>6</sup> has been employed instead of binary classification in order to make the analysis focus on the volume data with smaller and proper size. Next, the Savitzky-Golay (S-G) filter<sup>13</sup> and Gaussian convolution are used to reduce the noise in the data and calculate the first and second derivatives. Last, direct volume rendering using ray casting is applied to visualize the volume data. To support our conclusions, we compare the results of Hessian matrix<sup>8-10</sup> to images produced by our algorithms.

## 2. Methods

### 2.1. Definitions

Let  $\mathcal{V}(N)$  denote an observed volumetric data which has  $N = N_x \times N_y \times N_z$  voxels, and  $f(\mathbf{x})$  be an intensity function of the volume, where  $\mathbf{x} = (x, y, z) \in \mathcal{V}$ .  $\mathbf{M}$  is a map from  $\mathcal{V}$  to  $\mathcal{U} \in \mathcal{R}^4$  defined as

$$\mathbf{M}: (x, y, z) \rightarrow (x, y, z, f(x, y, z)). \quad (1)$$

Then,  $\mathcal{S} = \{\mathbf{v} \in \mathcal{R}^4 | \mathbf{v} = (x, y, z, f(x, y, z))\}$  represents the hyper-surface traced by the image. We can compute the three principal curvatures  $k_1 \geq k_2 \geq k_3$  and their corresponding normalized eigenvectors  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$  from the  $3 \times 3$  Weingarten matrix<sup>7-1</sup> defined as follows:

$$\mathbf{W} = \frac{1}{D} \mathbf{H} (\mathbf{I} + (\nabla f)^\tau \nabla f)^{-1}, \quad (2)$$

at each vertex on  $\mathcal{S}$ , where  $\tau$  is the matrix transpose,  $\mathbf{I}$  is the  $3 \times 3$  identity matrix,  $D = \sqrt{1 + f_x^2 + f_y^2 + f_z^2}$ ,  $\nabla f$  and  $\mathbf{H}$  denote the gradient vector and the Hessian matrix, respectively. The eigenvector  $\mathbf{e}_1$ , corresponding to the largest eigenvalue  $k_1$ , represents the direction along which the amplitude of curvature is maximum. The gradient vector is defined as

$$\nabla f = (f_x, f_y, f_z), \quad (3)$$

where partial derivatives of volume intensity  $f(\mathbf{x})$  are represented as  $f_x = \partial f / \partial x$ ,  $f_y = \partial f / \partial y$ , and  $f_z = \partial f / \partial z$ . The gradient vector is widely used as a normal to an implicitly defined isosurface, and its magnitude provides us with a measure of 3D edge structure and can be used as a modulation factor in 3D imaging<sup>8-9</sup>. The Hessian matrix is given by

$$\mathbf{H} = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{pmatrix}, \quad (4)$$

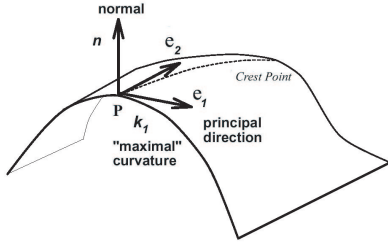
where second-order partial derivatives of  $f(\mathbf{x})$  are represented as  $f_{st} = \partial^2 f / \partial s \partial t$ , ( $s, t = x, y, \text{ or } z$ ). Let the eigenvalues of  $\mathbf{H}$  be  $\lambda_1, \lambda_2, \lambda_3$  ordered by  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ , and their corresponding normalized eigenvector be  $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ , respectively. The eigenvector  $\mathbf{t}_1$ , corresponding to the largest eigenvalue  $\lambda_1$ , represents the direction along which the second derivative is maximum, and  $\lambda_1$  gives the maximum second-order derivative value. Similarly,  $\lambda_3$  and  $\mathbf{t}_3$  give the minimum directional second-order derivative value and its direction, and  $\mathbf{t}_2$  is the cross product of eigenvectors  $\mathbf{t}_3$  and  $\mathbf{e}_1$ , i.e.,  $\mathbf{t}_2 = \mathbf{t}_3 \times \mathbf{e}_1$  and  $\lambda_2$  gives the directional second-order derivative value along  $\mathbf{t}_2$ .

A *crest point* is a local shape feature of a surface, and defined by<sup>1-2</sup>

$$\nabla k_1(x, y, z) \cdot \mathbf{e}_1 = 0, \quad (5)$$

Crest points are defined using directional derivative of the maximum curvature of the volume data. A vertex in the volume is a crest point if the maximum curvature is extremal in the corresponding principal direction. Figure 1 illustrates the geometric meaning of a crest point in a 2D surface. By Equations (2) and (5), crest point extraction involves third-order derivatives, therefore, the volume should be smoothed before crest points are extracted. We are also specifically interested in a *line-like point* defined using the eigenvectors and eigenvalues of the Hessian matrix at  $\mathbf{x}$ . Two conditions must hold for  $\mathbf{x}$  to be a line-like point<sup>8-10</sup>. One, the point  $\mathbf{x}$  must be a ridge point. Second-derivative information distinguishes ridges from valleys, saddles, planes, and spheres. The Hessian matrix captures second-derivative information at a point in an image. For a point to be on a 1D ridge of an 3D hyper-surface, two of eigenvectors of the Hessian of  $\mathcal{V}(N)$  at  $\mathbf{x}$  must have negative eigenvalues. Given ordered two eigenvalues, we test this condition by verifying

$$0 > \lambda_2 \geq \lambda_3. \quad (6)$$



**Figure 1:** Illustration of the geometric meaning of crest point in a 2D surface.

Two, the ridge is central to a tubular object that has a nearly circular cross section in  $\mathbf{t}_2\mathbf{t}_3$  plane, and has longer axis along the  $\mathbf{t}_1$  direction. That is, the three eigenvalues of the Hessian matrix must satisfy the following conditions,

$$\frac{\lambda_2}{\lambda_3} \geq 1 - \varepsilon, \text{ and } \left| \frac{\lambda_1}{\sqrt{\lambda_2\lambda_3}} \right| < \delta \quad (7)$$

where  $\varepsilon > 0$  and  $\delta > 0$  are less than 1.

## 2.2. Savitzky-Golay Smoothing Filters

In this subsection, we will describe the S-G filter<sup>13</sup>, and show how it smoothes a CLSM data while still keeps the thin and close the linear features in the data.

The simplest type of digital filter replaces each data value  $f_i$  by a linear combination  $g_i$  of itself and some number of nearby neighbors,

$$g_i = \sum_{n=-n_l}^{n_r} c_n f_{i+n}, \quad (8)$$

where  $n_l$  is the number of points toward left of a data point  $i$ , while  $n_r$  is the number used to the right. This is called *moving window filter*, for some fixed  $n_l = n_r$ , compute each  $g_i$  as the average of the data points from  $f_{i-n_l}$  to  $f_{i+n_r}$ , and corresponds to Equation (8) with constant  $c_n = 1/(n_l + n_r + 1)$ . However, moving window filter will merge the line structures which are close to each other. The idea of Savitzky-Golay filtering is to find filter coefficients  $c_n$  that preserve higher moments, i.e., to approximate the underlying function within the moving window not by a constant, but by a polynomial of higher order, that is, to fit a polynomial of degree  $M$  in  $i$ , namely  $a_0 + a_1i + \dots + a_Mi^M$  to the value  $f_{-n_l}, \dots, f_{n_r}$ . Typically  $M$  is 2 or 4. The coefficients of S-G filter satisfy

$$c_n = \sum_{m=0}^M \left\{ (\mathbf{A}^\tau \cdot \mathbf{A})^{-1} \right\}_{0m} n^m, \quad (9)$$

where the matrix  $\mathbf{A}$  is  $(A_{ij})$ , and  $A_{ij} = i^j$ ,  $i = -n_l, \dots, n_r$ ,  $j = 0, \dots, M$ .

Figure 4(a) shows the distribution of coefficients of S-G

filter when the window of width 27 (i.e.  $n_l = n_r = 13$ ) and  $M = 2$  and  $M = 4$ .

Figure 4(b) shows a numerical experiment using a 27 point smoothing filter (this is only for illustration purpose to show the shape of the curve, but for real data the window size is less than 9). The black curve shows a test function, constructed to have four "bumps" of varying width. Gaussian white noise of unit variance was added to the test function shown in gray. The red curve shows the result of smoothing by the moving window filter. One sees that the window of width 27 does quite a nice job of smoothing the broadest bump, but the narrower bumps suffer a considerable loss of height and increase of width. That means the thin line structures in the data will be missed. The green curve shows the result of smoothing with S-G filter of identical width, and degree  $M = 4$ . One sees that the heights and widths of the bumps are preserved well. A trade-off is that the broadest bump is less smoothed.

## 2.3. Volume Rendering and Algorithm

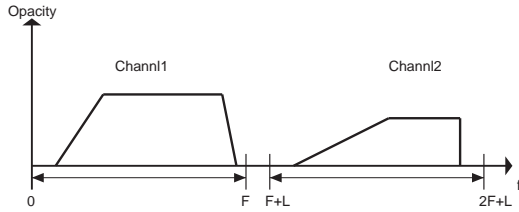
Direct volume rendering is slow because it is computationally intensive, therefore, we chose to a software package intended for conventional volume rendering. We carried out the method using the volume rendering modules in the Visualization Toolkit<sup>12</sup> (vtk version 3.1). These vtk modules generate both unshaded and shaded composite images from a scalar volume. In order to explore the relationship between the spindle and its scaffold or between scaffold and DNA, we have to render multichannel data, that is, there are at least two intensities, denoted by vector-valued  $(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots)$ , at each voxel, although the extraction of the spindle scaffold just use one channel. To render the multichannel data, we convert a vector-valued volume to a scalar volume by assigning the scaffold data and spindle or DNA to different intervals<sup>5 8</sup>. In our application, the first interval is linearly mapped by raw data and the second one corresponds to crest points (see Figure 2) as follows

$$g(\mathbf{x}) = \begin{cases} F_i(f_i(\mathbf{x}) - m_i)/M_i, & \text{if } \mathbf{x} \text{ not feature point} \\ F_i(f_i(\mathbf{x}) - m_i)/M_i + F_i + L, & \text{if } \mathbf{x} \text{ feature point} \end{cases} \quad (10)$$

where  $M_i$  and  $m_i$  is equal to the maximum and minimum intensity of the  $i^{\text{th}}$  channel of volume data,  $L \leq 5$  is a small gap between intervals, and  $F_i = (M_i - L)/2$ . The opacity and color transfer functions will depend on  $g(\mathbf{x})$  only.

Based on the above discussion, the algorithm for scaffold extraction for volume rendering is outlined as follows:

- Step 1** Segment the interest regions out of a volume data, and obtain a new volume data with a smaller size.
- Step 2** Smooth the new data by using a combination S-G and Gaussian filter with  $\sigma \leq 1$ .
- Step 3** Compute the Weingarten matrix and crest point described by Equations (2) and (5) using Gaussian convolution.



**Figure 2:** Multichannel data converted to scalar volume data.

**Step 4** Render multichannel volume data using ray casting whose opacity and color transfer depend not only on the intensity at vertex but also on whether it is a feature point or not.

If we calculate the Hessian matrix and its eigenvalues instead of the Weingarten matrix in **Step 3**, we can still extract line-like local structures.

### 3. Results

In this section we will look at two examples illustrating the proposed method. The first example examines artificial volume data added with Gaussian white noise. The second example uses real CLSM data, and demonstrate the hidden spindle scaffold extraction using crest points for volume rendering. In order to compare the difference of extraction methods between crest point and line-like point based on Hessian matrix, some results of line-like point extraction based on Hessian are given. The hardware we used is a Dell Precision workstation 330, with P4 1.4GHz CPU, and 1-GB RAM.

For all experiments and applications presented in this paper to test for equal-to-zero in Equation (5), we use

$$\left(\frac{\partial k_1}{\partial x}\right)^2 + \left(\frac{\partial k_1}{\partial y}\right)^2 + \left(\frac{\partial k_1}{\partial z}\right)^2 < 0.001, \quad (11)$$

and use 0.5 for  $\epsilon$  and  $\sqrt{2}/4$  for  $\delta$  in Equation 7. This allows cross section intensity patterns to deviate from a circular shape and resemble an ellipse with a 2:1 ratio between the lengths of the major and minor axes, but with a 4:1 ratio at least between the lengths of the axis along the eigenvector corresponding to the maximum eigenvalue and the major axis of the cross ellipse. For S-G filter, we use the different parameters for each experiment.

#### 3.1. Simulated data

We create a  $100 \times 100 \times 100$  voxel dataset containing three sine-shaped, parallel, tubular objects. Two of them close to each other are called twins, another far away is called single. The distance of centerline is 4 pixels between the twins and 16 pixels between single and the nearest one of the twins.

The distance of 6 pixels is representative of gap between two nearest fingers detectable in CLSM data. The upper part (defined by  $z > 50$ ) and the lower part (defined by  $z \leq 50$ ) in the simulated data are added different levels of identical noise, respectively. Noise standard deviations are 20 in the upper part and 50 in the lower. The noise-free background values are 20 in the upper and 50 in the lower and the noise-free tubes have Gaussian intensity profiles, value of 150 along the centerline. When we apply the S-G filter, we use 5 for the size of moving window and 4 for the degree of polynomial, and  $\sigma = 0.85$  for Gaussian filter.

Figure 6(a) shows the result of ray casting of the whole simulated dataset, and the upper part of the three tubular objects is clearer than the lower part of them. Figure 6(b) shows the results of our algorithms on the simulated dataset using the same opacity and color transfer function as in the result shown Figure 6(a). The three objects are almost uniformly visualized in the upper and lower part of the data, and the twins are separated more clearly. The results of line-like point extraction based on Hessian matrix are given in Figure 6(c) and (d), which are rendered using vector and boundary surface representation, respectively. A vector in Figure 6(c) indicates the direction of eigenvector corresponding to the maximum eigenvalue of Hessian matrix at a line-like point, and starts from it, and has unit length. The surfaces shown in Figure 6(d) characterize the boundary of the set of line-like point in the simulated data. The lower parts of the surfaces are fatter than the upper part of them, due to the different levels of noise.

#### 3.2. Real CLSM Data

Cell biologists use CLSM to visualize mouse eggs where they study signaling pathways that instruct eggs to leave meiotic metaphase II cell cycle arrest and continue in development into an embryo. Cell cycle arrest is maintained by proteins and enzymes found on the meiotic spindle. The meiotic spindle is a cytoskeletal structure that is used as a scaffold in eggs to organize both spatially and temporally signaling elements that regulate this event<sup>3 4</sup>. The meiotic spindle is about  $12 \mu\text{m}$  in length and is contained within an egg with a diameter of approximately  $60 \mu\text{m}$ . Using immunocytochemical, methods the egg is labeled with an alpha tubulin antibody so that any structure that is composed of this cytoskeletal protein can be visualized by laser excitation using the CLSM. As shown in Figure 3 the meiotic spindle (seen in green) is composed of alpha tubulin and can be clearly distinguished within the egg. Because of scattered light, features within the spindle greater than  $2 \mu\text{m}$  in distance appear as a blur. After segmenting the spindle region using Weibull E-SD fields<sup>6</sup>, the Gaussian and S-G filters are applied. The image data is filtered by these algorithms which will reduce the amount of scattered light that creates background noise. The data is then analyzed using the crest point algorithm that selects for dominant patterns found within the data.

In order to contrast the extraction of crest points for volume rendering, Figure 5 shows the results using line-like point collection based on Hessian matrix. The means of vector in Figure 5(a) and surface representation in (b) are the same in Figure 6(c) and (d), respectively. The line-like points in the same connected component in Figure 5(a) are labeled the same color. However, several fingers of the spindle have some huge "blobs", on the contrary, some fingers are broken. This is due to the lack of adaptive selection of parameters in Equation 7 to extract variant types of line-like points. Therefore, the extraction of line-like points is too more around some places but too less around other.

The scattered light has been removed and the resulting image shows detailed structures found within the spindle. The resolution between features has been increased from  $2\mu\text{m}$  to about  $1-1.9\mu\text{m}$  accuracy. This image was also visualized as its original 3D composite image using ray casting that also demonstrated this increased detail (see Figure 7).

Increased resolution from the original confocal image to that of an analyzed image has revealed increased detail that has not been visualized with confocal data before. We can clearly see cytoskeletal "fingers" and are able to visualize potential astral microtubules that have only been seen in mitotic spindles of mouse somatic cells and not known to be found in the mouse meiotic spindles. The discovery of astral microtubules gives additional information to researchers because they are thought to position the spindle within a cell and assist in the cell cycle progression.

#### 4. Conclusions

We used the crest point algorithm to automatically extract the centerline of spindle scaffold for volume rendering after segmenting the spindle region using Weibull E-SD fields, and smoothing by S-G filters. We have consistently demonstrated this approach on controlled as well as on real volume data. Comparison with the extraction using line-like point based on Hessian matrix, the scaffold extraction using crest point needs less control parameters and is more efficient.

#### Acknowledgements

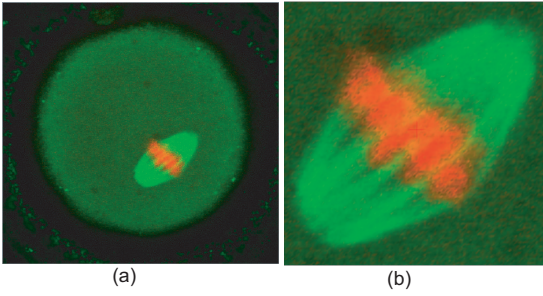
This work was supported by the Defense Advanced Research Projects Agency (MDA 972-00-1-0027) and the National Science Foundation (IIS-998016 and ACI-0083609) and the National Institutes of Health (HD 32621) and the Office of Naval Research (N00014-00-1-0281 & N00014-02-1-0287). We would like to thank PRISM, the W.M. Keck BioImaging laboratory and the Cell Biology laboratory in ASU for providing the data and computing resources.

#### References

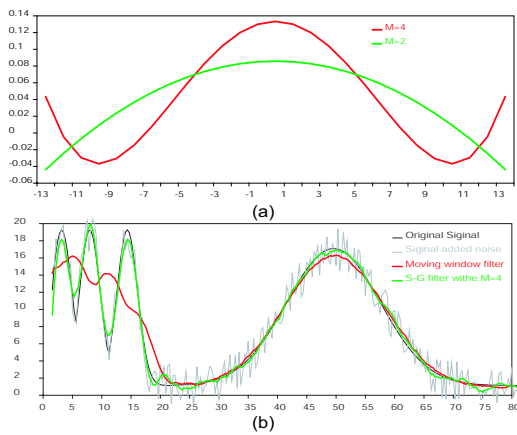
1. V. Prinet, O. Monga, C. Ge, S.L. Xie, and S.D. Ma, Thin network extraction in 3D images: application to

- medical angiograms. *Proc. of the 13th International Conf. on Pattern Recognition*, (3):386-390, 1996. 2
2. O. Monga, R. Lengagne, R. Deriche, Crest lines extraction in volume 3D medical images: a multi-scale approach *Proc. the 12th IAPR Internat. Conf. on Pattern Recognition*, (1):553-555, 1994. 2
3. D. P. Baluch, and D. G. Capco, Cellular Scaffolds in Mammalian Eggs. *Frontiers in Bioscience* (7):1653-1661, 2002. 1, 4
4. H. de Pennart, C. Cibert, C. Petzelt, B. Maro. Microtubule tracks can be detected in mouse oocytes with an antibody directed against a calcium transporter. *J. Cell Sci.* 107:1899-908, 1994. 1, 4
5. A. Razdan, K. Patel, G. Farin and D. G. Capco, Visualization of Multicolor CLSM data set, *Computers and Graphics*, (25):371-382, 2001. 3
6. J. Hu, A. Razdan, G. Nielson, G. Farin, D. P. Baluch, D. G. Capco, Volumetric Segmentation Using Weibull E-SD Fields. *IEEE Trans. Vis. and Computer Graphics*, (in press). 2, 4
7. G. M. Nielson, T. A. Foley, B. Hamann, and D. Lane, Visualizing and modeling scattered multivariate data. *IEEE Computer Graphics and Appl.*, (11):47-55, 1991. 2
8. Y. Sato, C. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis, Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Trans. Vis. and Computer Graphics* (6): 160-180, 2000. 1, 2, 3
9. Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. Three Dimensional Multiscale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. *Med. Image Anal.*, (2):143-168, 1998. 1, 2
10. A.F. Frangi, W.J. Niessen, K.L. Vincken, and M.A. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130-137, 1998. 1, 2
11. S.R. Aylward, E. Bullitt, Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction *IEEE Trans. Med. Imaging*, (21):61-75 2002. 2
12. W. Schroeder, K. Martin, B. Lorensen, The visualization toolkit an object-oriented approach to 3D graphics. *Prentice Hall*. 3
13. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes. *Cambridge University Press* (2nd edition) 1992. 2, 3

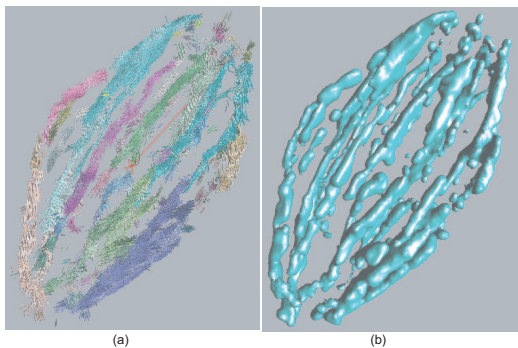




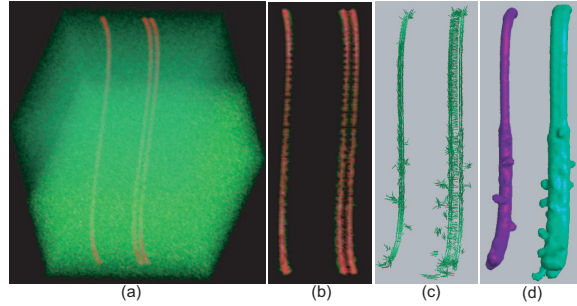
**Figure 3:** These images show single tiff slices. (a) a slice of raw data, (b) a slice of data segmented using Weibull E-SD highlighting the spindle region



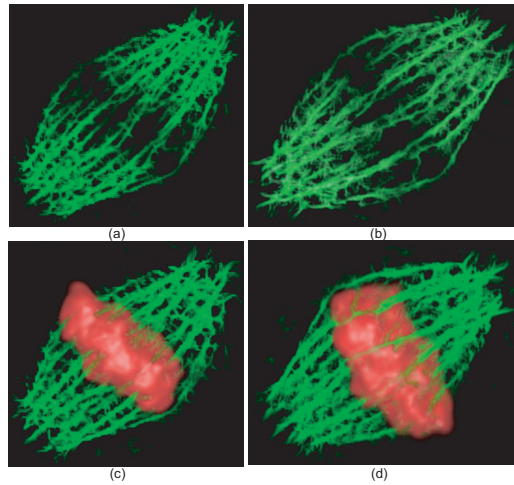
**Figure 4:** (a) The distribution of coefficients of S-G filter when the window of width is 27 (i.e.  $n_l = n_r = 13$ ) with  $M = 2$  and  $M = 4$ . (b) A numerical experiment using a 27 point smoothing filter.



**Figure 5:** The results of line-like point extraction based on Hessian matrix of the real mouse egg data, (a) vector representation, and (b) boundary surface representation of the set of line-like points.



**Figure 6:** A simulated data containing three sine-shaped, parallel, tubular objects. (a) The result of ray casting of the whole simulated dataset. (b) The results of our algorithms using the same opacity and color transfer function as in the result shown in (a). The results of line-like point extraction based on Hessian matrix are given in (c) and (d), which are rendered using vector and boundary surface representation, respectively.



**Figure 7:** (a) and (b) are the different viewpoints of spindle scaffold with increased detail using our extraction algorithms. (c) and (d) are the different viewpoints of multichannel ray casting which includes spindle scaffold (green) and DNA (red).