Volumetric Feature Extraction and Visualization of Tomographic Molecular Imaging

Chandrajit Bajaj[*] Zeyun Yu[†]
Department of Computer Sciences & Institute of Computational and Engineering Sciences
University of Texas at Austin, Austin, TX 78712

Manfred Auer[‡]
Laboratory of Sensory Neuroscience
The Rockefeller University New York, NY 10021

September 15, 2003

Abstract

Electron tomography is useful for studying large macromolecular complex within their cellular context. The associate problems include crowding and complexity. Data exploration and 3D visualization of complexes require rendering of tomograms as well as extraction of all features of interest. We present algorithms for fully automatic boundary segmentation and skeletonization, and demonstrate their applications in feature extraction and visualization of cell and molecular tomographic imaging. We also introduce an interactive volumetric exploration and visualization tool (Volume Rover), which encapsulates implementations of the above volumetric image processing algorithms, and additionally uses efficient multi-resolution interactive geometry and volume rendering techniques for interactive visualization.

1 Introduction

Most if not all proteins in a cell are organized into cellular machines that are often built from several dozens of individual proteins [1]. While certain cellular machines, such as the ribosome, are always built in only one and well defined way, other cellular machines are expected to vary in their exact 3D structure, while following a similar architectural principle. Cellular machines are dynamic with transient addition or loss of protein components. Hence, two such machines can

[*] bajaj@cs.utexas.edu
[†] zeyun@cs.utexas.edu
[‡] auerm@mail.rockefeller.edu

be expected to be similar in composition and architecture, but not necessarily identical in their 3D structure. Moreover, some of the most interesting cellular machines, however, are too rare or too fragile to be isolated and purified by biochemical means, and they only function in their cellular context, requiring for example the integrity of the cytoskeleton, the plasma membrane as well as extracellular matrix components. For such delicate yet biologically very important multiprotein complexes, electron tomographic imaging provides the only foreseeable way to obtain 3D structural information. All other structural techniques such as spectroscopic, diffraction or single-particle analysis cryo-electron microscopic techniques rely implicitly or explicitly on averaging of a large number of identical particles [20, 3, 52]. Electron tomography, in contrast can provide 3D structural information of such unique volumes as whole cells. Although cellular tomographic imaging is no means a new technique [23, 25], only recently it has received more attention [17, 11, 12, 36, 38] due to progress on the automation of data acquisition [28, 16], minimization of the electron dose for data collection [35], as well as hardware improvements from electron microscope manufacturers. Although still an expert technique, electron tomographic data collection is no longer the bottleneck, and user-friendly commercial packages for data collection are being offered.

While recording devices (CCDs) are becoming larger, and data collection becomes faster, the bottleneck in this emerging field lies more and more on the visualization and interpretation of the tomograms. So why are tomograms so much harder to study and interpret? The answer may lie in the following co-mingled reasons: First, most tomograms exhibit a low signal-to-noise ratio, and straightforward averaging techniques cannot be employed to enhance the signal. Second, the cellular machine does not reside in isolation but are embedded in their cellular context, and densely surrounded by other proteins that may or may not directly interact with the cellular machine, a concept also known as macromolecular crowding [18]. Third, one often doesn't know the exact composition and conformation of cellular machines at the time of investigation. The poor signal-to-noise ratio usually observed in tomograms complicates automated feature extraction, as well as the visualization of the volume. Hence, noise reduction is always utilized as a pre-processing step to improve the signal-to-noise ratio. Segmentation, is often necessary to obtain an unobstructed view into the machinery's architectural organization, and to reduce the complexity of the scenery to allow for biological interpretation Feature extraction is particularly challenging if the cellular machine of interest is in close contact to its cellular surrounding, and if there is no preconception of its 3D structure. In such cases, manual segmentation approaches appear somewhat subjective and become less feasible even with the help of 3D data re-slicing along non-orthogonal angles to obtain a more favorable view [24, 29, 31, 37, 22]. Moreover, they are unlikely to keep up with the amount of data that can be generated by modern-day electron microscopes.

Electron tomography may have its biggest impact in the emerging field of structural cell biology with the goal to visualize cellular compartments without prior anticipation of the machine's architectural organization. Hence when exploring uncharted territory, we need a tool for interactive exploration of a 3D density data set. Section-by-section inspection of the raw volume, followed by segmentation and rendering is a very time consuming process, and doesn't allow real-time data exploration and mining. Moreover one may fail to recognize the architecture of the complex if one has to segment the volume one slice at the time. Rendering of whole tomograms is usually beyond the graphical capabilities of computer desktop machines, due to the size of typical tomograms of 512x512x100 voxels. For fast interactive data exploration it is desirable to have a visualization tool in hand that allows simultaneous real time high-quality rendering of the whole tomogram at a lower resolution for navigation, as well as sub-volumes at full resolution for close inspection and analysis.

The ultimate aim is to interpret the densities obtained by tomographic imaging and segmentation using models of protein components. Model building has been the key in interpreting protein biological structures, and has led to a level of insight that was not available from the obtained electron density alone [51]. If the exact composition is known and the resolution is sufficient, protein structures can be fitted into the density maps, either manually using interactive 3D graphics programs (e.g., [26]) or semi-automatically [54, 59, 58]. Other approaches such as template matching [13] have been proposed for data exploration and analysis. The complexity of cellular 3D volumes requires some form of data reduction and simplification. Skeletonization [30, 64] is an additional way to simplify 3D data sets while retaining their characteristics, which is also important in comparing two complexes that are similar but not identical. Skeletons will be helpful in comparing two such cellular machines and describing their similarities and discrepancies.

The rest of this paper is as follows. Section 2 presents algorithms for fully automatic volumetric boundary segmentation as well as skeletonization, as applied to electron tomography imaging data. In section 3, we present an interactive volumetric exploration tool (Volume Rover) that we have developed which encapsulates implementations of the filtering, and curve/surface feature extraction algorithms, and additionally uses multi-resolution interactive geometry and volume rendering, for visualization. Finally, in section 4, we exhibit results of our application of the volumetric image processing and visualization of transmission electron tomographic three dimensional cell organelle data.

2 Volumetric Feature Extraction Algorithms

2.1 Gradient Vector Diffusion

It is sometimes more convenient to work on vector fields rather than the gray-scale intensities. A widely used vector field, is the gradient vector field, which has been employed for image segmentation [60, 62, 63]. We show in the present paper, how gradient vector fields are also useful for skeleton extraction. The gradient vector field calculated from the original (and even filtered) tomogram is often subject to noise. Even though the noise can be reduced by various types of scalar filters, the gradient vectors may still not “smoothly” vary over the image domain due to the errors of calculating derivatives on discrete and small neighborhoods. Furthermore, the gradient vectors often vanish in “flat” regions, which may make it difficult to locate critical points from the gradient vector field (see below). It is becoming mandatory to additionally “smooth” the gradient vector field prior to other image processing procedures such as segmentation or skeleton extraction. In [60], the authors described a PDE-based diffusion technique to smooth gradient vector fields. The gradient vectors are represented by Cartesian coordinates and similar partial differential equations (PDEs) are separately applied to each component of the vectors:

$$\begin{cases} \frac{du}{dt} = \mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2 + f_z^2) \\ \frac{dv}{dt} = \mu\nabla^2 v - (v - f_y)(f_x^2 + f_y^2 + f_z^2) \\ \frac{dw}{dt} = \mu\nabla^2 w - (w - f_z)(f_x^2 + f_y^2 + f_z^2) \end{cases} \tag{1}$$

where (u, v, w) is initialized with $\nabla f(x, y, z)$, and $f(x, y, z)$ is an edge map of the original image; that is, $f(x, y, z) = |\nabla I(x, y, z)|^2$. These diffusion equations are originally used for image segmentation [60]. For calculations of critical points (described in the following), however, these equations should be applied directly on the original image (that is, $f(x, y, z) = I(x, y, z)$).

The smoothing on each component of the gradient vectors may cause some unwanted effects [62]. In our method, all vectors are represented by their polar forms (namely, magnitude and orientation) and the diffusion equations are applied to the magnitudes and orientations separately. This method proves to perform better for image segmentation around long-thin boundary concavities [62]. Moreover, the gradient vector diffusion based on polar-coordinate representations appear much more desirable for skeleton extraction where the gradient vectors on one side of the skeletons are much “stronger” than the vectors on the other side. In this case, the conventional method [60], cause the skeletons ‘drift” towards the side of the weaker vectors due to the unfair competition between vectors on both sides of the skeletons. The polar-coordinate method [62, 61] keeps the magnitudes less significant and thereby yields more accurate skeletons. A disadvantage of this method is that it is much more time-consuming, especially for 3D volumes [61]. A tradeoff between speed and accuracy is to apply a normalized version of Eq.(1) [63]. In this implementation, all the non-zero vectors are normalized into the same magnitude range, before diffusion in each iteration is applied. The ”skeleton-drifting” problem can be eliminated but the noise may be amplified such that superfluous and incorrect skeletons may be extracted and some post-processing operations may become necessary.

In this paper, we present a variant of the partial differential equations (PDEs) for gradient vector diffusion, to address nearly all the afore-mentioned problems. The new PDEs are similar to Eq.(1) except that they are now based on anisotropic diffusion:

$$\begin{cases} \frac{du}{dt} = \mu\nabla(g(\alpha) \cdot \nabla u) - (u - f_x)(f_x^2 + f_y^2 + f_z^2) \\ \frac{dv}{dt} = \mu\nabla(g(\alpha) \cdot \nabla v) - (v - f_y)(f_x^2 + f_y^2 + f_z^2) \\ \frac{dw}{dt} = \mu\nabla(g(\alpha) \cdot \nabla w) - (w - f_z)(f_x^2 + f_y^2 + f_z^2) \end{cases} \tag{2}$$

where $g(\cdot)$ is a decreasing function and α is the angle between the central vector and the surrounding vectors. For faster implementation, the calculation of the angle between two vectors is usually approximated by the inner-product of two vectors divided by their magnitudes. For instance, we can define $g(\alpha)$ as follows:

$$g(\vec{c},\vec{s}) = \begin{cases} e^{\kappa\cdot(\frac{\vec{c}\cdot\vec{s}}{\|\vec{c}\|\|\vec{s}\|}-1)} & if\ \ \|\vec{c}\| \neq 0\ \ and\ \ \|\vec{s}\| \neq 0 \\ \\ 0 & if\ \ \|\vec{c}\| = 0\ \ or\ \ \|\vec{s}\| = 0 \end{cases} \tag{3}$$

where κ is a positive constant; $\vec{c}$ and $\vec{s}$ stand for the central vector and one of the surrounding vectors, respectively. In our implementation, we consider a 6-neighborhood for each voxel.

A study of Eq.(2) would suggest the following: First, its implementation is similar to that of Eq.(1). Hence, the computational time is comparable to the conventional scheme [60]. Second, with the appropriate weighting function $g(\cdot)$, we overcome both the long-thin boundary concavity problem and more importantly the "skeleton-drifting" problem. Third, the amplification of noise no longer exists. In the following subsection, we shall also see how this anisotropic gradient vector diffusion is further applied to boundary segmentation, as well as skeleton extraction. In the case of skeleton extraction, we do not consider the second term in both Eq.(1) and Eq.(2), as they were originally used for better segmentation [60], but not for skeletonization.

2.2 Boundary Segmentation

Segmentation is a way to electronically dissect the cellular machine from its cellular surrounding, which often obscures a clear view into the machinery's architectural organization[18]. Segmentation is usually carried out either manually [24, 29, 31, 37, 22] or semi-automatically on a sub-volume of the tomogram [55, 19]. Manual segmentation can be tedious and often subjective even with the help of a sophisticated graphical user interface [31, 34]. Automated segmentation is still recognized as one of the hardest tasks in the field of image processing although various techniques have been proposed for automated or semi-automated segmentation. Commonly used methods include segmentation based on edge detection, region growing and/or region merging, active curve/surface motion and model based segmentation. In particular, two techniques in particular have been discussed in detail in the electron tomography community. One is called the *water-shed immersion method* [55] and the other is based on normalized graph cut and eigenvector analysis [19].

We have developed a method for image segmentation based on the *fast marching method* [45, 33, 46]. The fast marching method is a simplified and faster variant of the traditional *level set method* [46]. The basic idea of this method is that a contour is initialized from a pre-chosen seed point, and the contour is allowed to grow until a certain stopping condition is reached. Every voxel is assigned with a value called *time*, which is initially zero for seed points and infinite for all other voxels. Repeatedly, the voxel on the marching contour with minimal *time* value is deleted from the contour and the *time* values of its neighbors are updated according to the following equation:

$$||\nabla T|| \cdot F = 1 \tag{4}$$

where F is called the *speed function* that is determined by the image information (intensity,

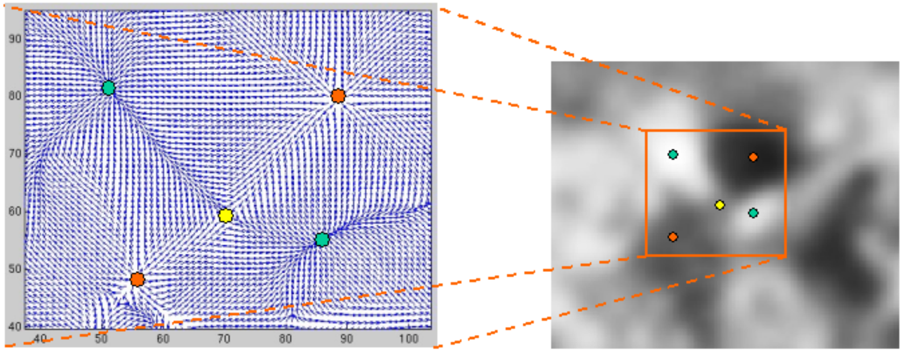


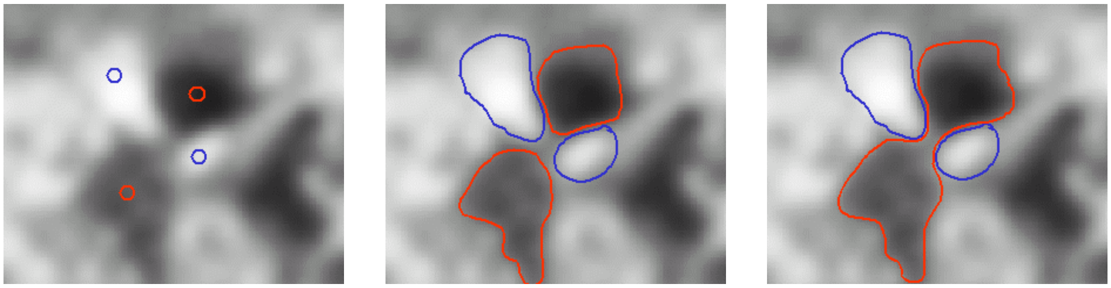
Figure 1: Computing critical points using gradient vector diffusion. A gray-scale 2D image is shown on the right, where a sub-region is considered for computing the gradient vector field. Anisotropic vector diffusion algorithm is applied and results are shown on the left. Five critical points are located including two maximal critical points (green), two minimal critical points (purple) and one saddle critical point (yellow).

gradient, and so on). The updated neighbors, if they are updated for the first time, are then inserted into the contour.

To implement the fast marching method, we needed to solve three key problems: (i) formulation of the speed function, (ii) initialization of the seed points, and (iii) determination of the stopping criterion. We formulate the speed function based on the gradient magnitude, as commonly used by other authors. The speed function is defined as an exponential function of the gradient magnitude; that is, $F = exp(\gamma||\nabla I||)$, where γ is a negative constant and I is the original image.

Initialization of the seed points are essential for correct segmentation of the components of interest. There are several ways to choose the initial seed points: mechanically by a user, using "balloons", or by the locus of the zero-crossing of the Laplacian of the smoothed images (see [47] for a summary). The aforementioned anisotropic gradient vector diffusion provides an efficient way to automatically locate relevant contour seed points. The gradient vector field derived from the original images (or tomographic volume) is diffused and the *critical* points are used as our seeds. Critical points corresponding to local maxima are those points where all the surrounding vectors point to these points. Similarly, critical points corresponding to local minima are those points where all the surrounding vectors point away from these points. Critical points corresponding to saddles are detected at those points where some surrounding vectors point to these critical points, from both sides along some directions, while the other surrounding vectors point away from these points. Fig.1 demonstrates a two-dimensional example, where all these three types of critical points are detected using anisotropic gradient vector diffusion. The critical points are detected in a similar way in the three-dimensional cases, as implemented in our image processing library, and available under the GNU public license.

In case of segmentation, we omit the saddle critical points. All the other seed points (min-



(a) Initial contours (b) Marching contours (c) Merged contours

Figure 2: Fast marching method using dual-contours. (a) We only consider four seed points in this image. The two blue contours correspond to the *maximum* critical points while the other two contours (red) correspond to the *minimum* critical points. (b) Whenever dual contours with different colors meet, they prevent each other from moving.(c) When dual contours with the same color meet, they merge into a single contour, and keep growing.

imum and maximum) are then grouped into two classes: *feature* seeds and *background* seeds. For example, if the features to be segmented have higher intensities than the background, then the *feature* seeds are the *maximum* critical points while the *background* seeds are the *minimum* critical points. Additional sub-groups may be chosen for applications requiring multi-material segmentation.

The stopping criterion of the marching contours is another important ”open” issue in the fast marching method. The fast marching algorithm described in [33] does not give an explicit stopping criterion such that the more expensive *level set method* had to be used to finalize the segmentation. In addition, like the fast marching method, even the *level set method* may have to deal with the “leaking” problem around the boundary gaps. In our current implementation for tomographic imaging data, we address this problem using dual contours: one starting from the *feature* seeds and the other starting from the *background* seeds. In the beginning, a contour is initialized at each seed point. Since all the seeds are classified into two (or more) groups, all the initial contours are accordingly classified into these groups. Each of the contours march (grow) simultaneously according to Eq.(4). Whenever two contours from the same group meet, they merge into a single contour. On the other hand, if two contours from different groups meet, both contours stop marching on the common boundaries. Both situations are illustrated in Fig.2, where we can see that the dual contours stop automatically. The idea of dual contours is analogous to the idea seen in the multi-label fast marching method [48, 49, 41] for motion analysis in video image processing, where an automatic stopping criterion was guaranteed due to the multiple contours marching towards the boundaries from opposite sides. The difference between these two approaches is the way to choose the seed points and, due to the different applications, the way to classify the seed points. In another application, the authors in [15] employed multiple contours to detect nuclei and cells. However, the multiple contours seen there were not used for designing an automatic stopping criterion.

With an appropriate parameter γ in the speed function F, we can guarantee that dual contours from different groups stop correctly on the boundaries where the magnitude of the image gradient is locally maximal. An interesting observation is that the seed points obtained in our method are quite similar to the seeds used in the watershed immersion method [55], and the marching process in our method is analogous to the immersion process. However, our method considers the image gradient, an important feature of image boundaries, which is not taken into account in the watershed immersion method. In addition, our method allows merging between similar group contours, starting from different seeds, which avoids the over-segmentation problem as commonly seen in the watershed immersion method. The gradient vector diffusion, which smoothes the vector field and hence reduces the number of the seed points, plays another key role in suppressing the over-segmentation problem.

2.3 Skeleton Extraction

The skeleton is recognized as one of the important imaging feature descriptors, and popular in image and video database indexing/retreival as well as pattern recognition. Commonly used computational methods for skeleton extraction include topological thinning, combinatorial methods based on polygon approximation of boundaries, approaches based on distance maps, hierarchical methods based on Voronoi diagrams (or dually, Delaunay triangulation), and some physically based techniques [30, 2, 27, 40, 21, 32, 64]. The majority of these techniques compute the skeletons from the object's boundaries. To apply these skeleton extraction methods to structure interpretation, it is necessary to first extract an appropriate level set of the density map under study. Unfortunately, the topology of the level set often changes significantly even in small ranges of density values, making the process extremely numerically sensitive. In addition, extracting iso-surfaces followed by boundary-based skeleton construction is computationally intensive. An alternate approach, which computes the "skeletons" directly from a scalar map, is based on Morse theory [39]. A Morse complex of a scalar map is constructed by detecting the critical points followed by tracing the integral linking curves by traversing the gradient vector field in their principal directions, see for e.g. [8]. The Morse complex of a volumetric image is an embedded graph consisting of a collection of vertices, and edges (curves), and possibly faces (surfaces). The faces (surfaces) can be further reduced (retracted) to a collection of curves, but these only provide a coarse approximation of the volumetric features.

In this paper, we present an approach to computing volumetric skeletons based on *anisotropic gradient vector diffusion* Like the Morse complex, the skeletons are directly extracted from the gray-scale volumes without computing the iso-surfaces. In the following we describe a detailed step-by-step algorithm .

Generate the initial gradient vector field. The initial gradient vector field is generated in the following way:

$$gvf(\vec{r}) = (I(\vec{r}) - I(\vec{r'})) \times \frac{\vec{r'} - \vec{r}}{\|\vec{r'} - \vec{r}\|} \tag{5}$$

where $I(\vec{r})$ is the intensity value at $\vec{r}$. $\vec{r'}$ is one of the (twenty-six) immediate neighbors of $\vec{r}$, which has lower intensity than the other immediate neighbors of $\vec{r}$. Note that this definition of gradient vector at a point is different from other conventional definitions (e.g., the one defined by finite difference). In the case of density maps where the features have lower intensities than the background, the gradient vectors are defined by:

$$gvf(\vec{r}) = (I(\vec{r'}) - I(\vec{r})) \times \frac{\vec{r'} - \vec{r}}{\|\vec{r'} - \vec{r}\|} \tag{6}$$

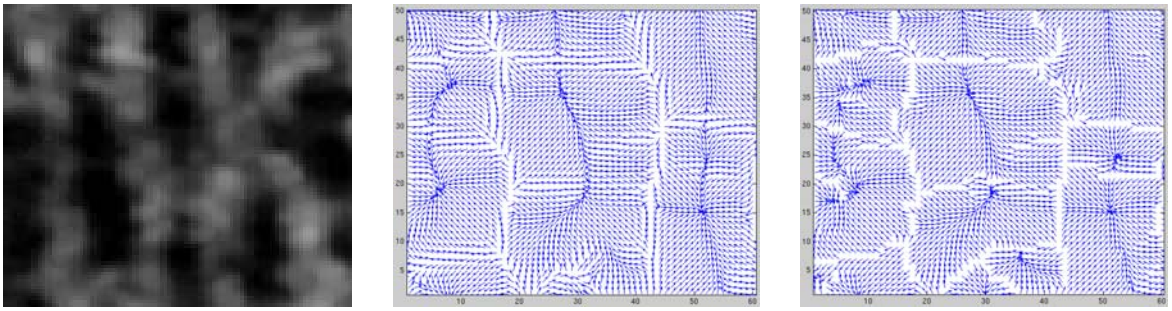
where $\vec{r'}$ is one of the (twenty-six) immediate neighbors of $\vec{r}$, which has higher intensity than the other immediate neighbors of $\vec{r}$. We shall demonstrate in the following the differences between Eq.(5), Eq.(6), and the classical definition by finite difference.

Diffuse the gradient vector field. The initial vector field computed above is then diffused by our anisotropic scheme (Eq. (2) without the second term). For comparison, we also demonstrate results of our earlier isotropic scheme (Eq. (1) without the second term). The PDE's are solved iteratively using finite difference techniques. In Fig. 3, we compare the diffused gradient vector fields from isotropic and anisotropic schemes. One major difference between these two schemes is that anisotropic diffusion preserves "sharp" features (the "blank" regions), where most of the surrounding vectors point away from the central point. These features also correspond to the skeletons of the original gray-scale image. The superiority of anisotropic diffusion to the isotropic diffusion is further demonstrated in the following.

Compute the skeleton magnitude map. To locate the "blank" regions of the diffused gradient vector field, we compute what we call *skeleton magnitude map* (SMM) using the following formula:

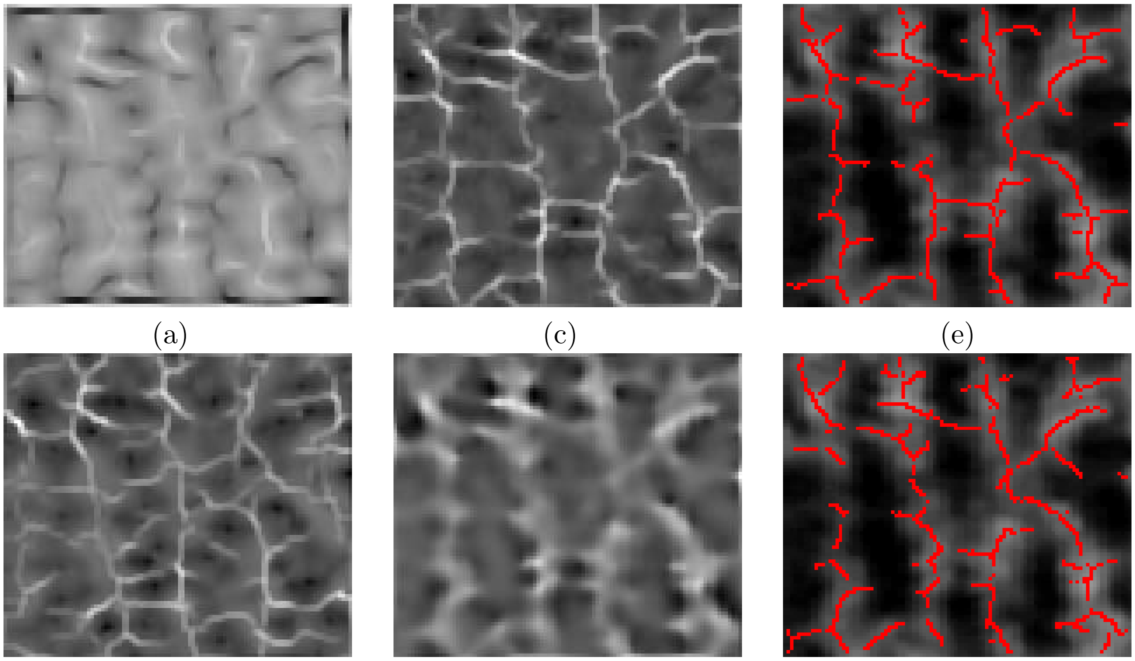
$$smm(\vec{r}) = \sum_{r' \in N(\vec{r})} \frac{gvf(\vec{r'}) \cdot (\vec{r'} - \vec{r})}{\|\vec{r'} - \vec{r}\|} \tag{7}$$

where $N(\vec{r})$ is the set of the twenty-six immediate neighbors of $\vec{r}$. The skeleton magnitude map is a scalar map defined on every voxel and indicates the likelihood of each voxel being on the skeletons. To demonstrate the influence of the different initialization strategies of the gradient vector field on the skeleton extraction, we show in Fig.4(a) the skeleton magnitude map of the original image (Fig.3(a)), generated by anisotropic vector diffusion but initialized by the classical finite difference scheme. We can see that this map does not give much information on the skeletons of either bright regions or the dark regions. In Fig.4(b) we show the SMM that is generated by anisotropic vector diffusion but initialized by Eq.(6). This map clearly indicates the skeletons of the dark regions in the original image (Fig.3(a)). Fig.4(c) gives the SMM, generated by the anisotropic vector diffusion but initialized by Eq.(5). From this map, we can clearly see the skeletons of the bright regions of the original image. In our experiments, we assume that all the



(a) Original image (b) Isotropic diffusion (c) Anisotropic diffusion

Figure 3: The comparison of isotropic and anisotropic vector diffusion schemes on a synthesized gray-scale image. For better illustration, the gradient vector fields ((b) and (c)) are restricted to the bottom-left quarter of the original image, and all vectors are normalized to the same magnitude range.



(a) (c) (e)

(b) (d) (f)

Figure 4: The comparison of skeleton magnitude maps by isotropic and anisotropic diffusions, and by different strategies for initialization of the gradient vector fields. The original image is shown in Fig.3(a). (**a**) SMM by anisotropic vector diffusion, initialized by a classical finite difference scheme. (**b**) SMM by anisotropic vector diffusion, initialized by Eq.(6). (**c**) SMM by anisotropic vector diffusion, initialized by Eq. (5). (**d**) SMM by isotropic vector diffusion, initialized by Eq. (5). (**e**) Skeletons extracted from (c). (**f**) Skeletons extracted from (d).

datasets contain features that have brighter intensities than the background. Hence, we restrict ourselves to the initialization scheme given by Eq.(5). In comparison to the anisotropic diffusion technique, Fig.4(d) shows the SMM generated by isotropic diffusion (Eq.(1) without the second term). A direct comparison between Fig.4(c) and Fig.4(d) would suggest that the isotropic vector diffusion normally tends to blur the skeletons while the anisotropic vector diffusion preserve "sharp" skeletons very well. This is analogous to the isotropic/anisotropic diffusion commonly seen in image smoothing [42, 56, 10].

Tracing Skeletons. The skeleton magnitude maps (SMM) can be directly rendered using our surface and volume-rendering tool, the Volume Rover (described in the next subsection). In this case, the skeleton magnitude maps are treated in a similar way to the original density maps, although the former one enhances the "contrast" of the skeletons. The direct rendering method, however, does not guarantee visualizing "thin" skeletons. Therefore, we trace the skeletons from the SMM such that the extracted skeletons (curvy lines or surfaces) have a thickness of only one voxel, utilizing[14]. This method was originally designed for image edge detection from gradient magnitude maps. The most promising strategies used in this method are non-maximal suppression and double-threshold. The non-maximal suppression is first applied to the gradient magnitude map in order to obtain "thin" edges and extract candidate edges. Two thresholds are assumed such that candidate edges above the higher threshold are always recognized as true edges and candidate edges that are connected to the true edges by a path of voxels with gradient magnitudes higher than the lower threshold are also recognized as true edges. This idea can be readily applied to our skeleton extraction by simply treating the skeletons as the edges. Fig.4(e)-(f) show the skeletons extracted by this method from the skeleton magnitude maps (c) and (d), respectively. From this perspective as well, the anisotropic vector diffusion scheme yields better results than the isotropic vector diffusion.

The skeletonization approach described above differs considerably from the traditional surface-based methods. One of the major differences is that we no longer worry about the sensitivity of choosing the iso-values, as we do not need to extract the iso-surfaces from the original volume. The skeleton magnitude map generated by anisotropic vector diffusion, in general, gives clear and accurate skeletons, (see for e.g. Fig.4). However, the connectivity of the obtained skeletons depends much on the parameters chosen in the skeleton-tracing step. Although Canny's method works quite well in most cases, errors do happen where the gradients of the skeleton magnitude maps are not well-defined (e.g., when several skeletons join together from different directions). We expect to improve the connectivity of the skeletons by more sophisticated skeleton-tracing algorithms.

3 Interactive Volume Exploration Tool

3.1 Surface and Volume Visualization

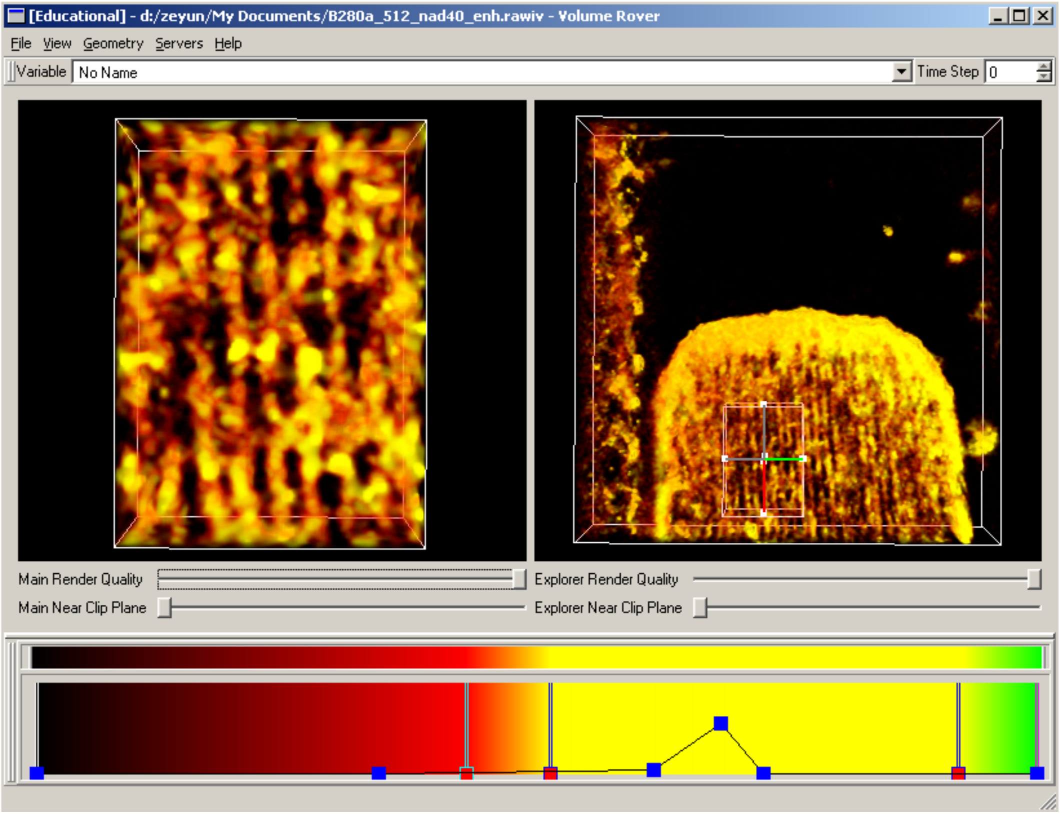
Typically, informative visualizations are based on the combined use of multiple techniques, including volume rendering, isocontouring, dynamic mesh reduction, global and local scalar, vector topology computation, feature extraction, etc. Informative visualization is thus a way to guide data-intensive computations to a spatial and temporal locales of interest and significance. Informative visualization consists of two primary components: Computation (rapid computation of isosurfaces, reduced meshes, volume rendering, etc., or more generally, of some "view" of the multivariate data) and Display (efficient rendering of the visualization with graphics primitives, including use of color, brightness, transparency, texture, volume, etc.).

We approach both of the key components through computer accelerated methods for contour extraction[6], dynamic mesh reduction for improved interactive display[9], real-time rendering working with compressed data streams [4, 5, 50], and using topological and volumetric quantitative signatures for feature extraction[53, 7]. We have encapsulated this combined functionality, along with the filtering and feature extraction techniques detailed below, into our volumetric exploratory visualization tool we call the Volume Rover.

The volume-rendering client can act as a 3D roving microscope, allowing users to visualize data that is too large to fit on a single machine. The graphical user interface allows for interactive visual selection of transfer function and isocontour, aka the contour spectrum [7, 43]. The user interface also allows the user to move and resize the sub-volume window. The data within the sub-window is then transmitted by the server to the client computer, and displayed interactively using fast texture based volume rendering that can be combined with rendered geometry [57, 44]. The rover connects to a data server that contains large datasets. The server can extract and resample sub-volumes of different sizes, which are then transmitted to the client for visualization. The client downloads data differentially, by only downloading the data that the client does not already have cached.

The Rover contains two views. One view contains a volume render of a sub-sampled version of the entire volume. The user specifies the sub-volume by interacting with a cuboid located in the sub-sampled volume. At the center of the cuboid are 3 axes, one for each dimension. The user can translate the sub-volume by clicking on one of these axes and then dragging along the axis. At the end of each axis is a resizing knob. The user can resize the sub-volume window by clicking on one of the resizing knobs and dragging along its corresponding axis. The user can also rotate the sub-window around each of its axes. After the user manipulates the sub-volume window, the client requests a sub-volume of the correct size and resolution from the data server. The data is downloaded from the server and rendered using fast texture-based volume rendering.

The data for the rover can either come from a remote data server , or can come from the local hard disk. In either case, if the data requested is too large to fit into graphics memory, it is filtered using Gaussian filtering and sub-sampled. The sub-sampled data is then loaded into the video card and rendered using texture based volume rendering. The Rover can visualize volumes up to 512x256x256 imaging without resampling.


[Educational] - d:/zeyun/My Documents/B280a_512_nad40_enh.rawiv - Volume Rover
File View Geometry Servers Help
Main Render Quality
Main Near Clip Plane
Explorer Render Quality
Explorer Near Clip Plane

Variable No Name

Time Step 0

Figure 5: The *Volume Rover* user interface. Note the interactive selection of a sub-region of interest via the exploratory sub-cuboid. The selected region of interest is dynamically updated and rendered (left sub-window view) at maximal resolution as one explores with the sub-cuboid inside the full volume being rendered in the right sub-window view.

When data come from the hard disk, the rover automatically caches a pyramid hierarchy of sub-sampled volumes. For each volume, the Rover filters and resamples the volume, creating a volume with half the resolution of the original volume. This process is repeated until the lowest resolution volume is 1x1x1. This speeds up the interactive exploration of the data since the data does not have to be resampled for every extraction.

In addition to the volume rendering, the user can request to see an isosurface rendering of the data. The rover performs isosurface extraction on the sub-volume portion of the data as well as the thumbnail data. The surface is rendered together with the volume. If the user moves the sub-volume, the rover obtains the new data and performs the isosurface extraction again. The new surface and new sub-volume are then rendered together. This allows the user to interactively explore the volume render as well as the isosurface render of the large data.

During the volume exploration or feature extraction process, it is also necessary on ocassion to zoom in to crop out volumetric regions of interest. The selected sub-region is then used for faster noise reduction and selected feature segmentation. The direct manipulation GUI (graphical user interface) of the Volume Rover allows us to visually identify an select specific volumetric sub-regions of interest (as shown in Fig.5).

Furthermore, the user can request a bilateral filtering, an anisotropic geometric diffusion evolution, a gradient vector diffusion, and skeleton feature extraction on the data set. The rover will then perform the filtering and feature extraction on the extracted sub-volume. If an isosurface is being rendered, it will be re-extracted from the newly filtered data. The new data is then displayed together with the new isosurface.

4 Applications and Results

In Fig.6 we demonstrate the segmentation of the densely grouped extracellular fibrillar proteins. Fig.7 shows the boundary segmentation of the intracellular actin-filaments (actin bundle). Fig.8 shows the boundary segmentation of extracellular filaments being secreted from frog saccular sensory epithelium supporting cells. It is worth noting that the criterion for classifying critical points may differ from data to data. All the maximum (or minimum) critical points may not necessarily classified into one group. Quite often we need to classify the maximum critical points with intensities below a threshold into the group of minimum critical points. Similarly some minimum critical points with intensities above a threshold (could be different from previous one) may be classified into the group of maximum critical points. As we said before, sometimes more than two groups may be necessary. For the three datasets shown in Fig.6-8, we consider only two groups, corresponding to feature and background respectively.

In Fig.9, and Fig.10, we show three examples of our skeleton extraction approach. Fig.9, shows the skeletons extracted from the densely grouped extracellular fibrillar proteins. We also show the skeletons embedded into the density maps, such that we can see the consistence between the skeletons and the density maps. All the skeletonal surfaces have a thickness of one voxel, although this is not much clear from the volume-rendered figures. In Fig.10, we show the skeletons extracted from the intracellular actin-filaments/actin bundle. Due to the complicated

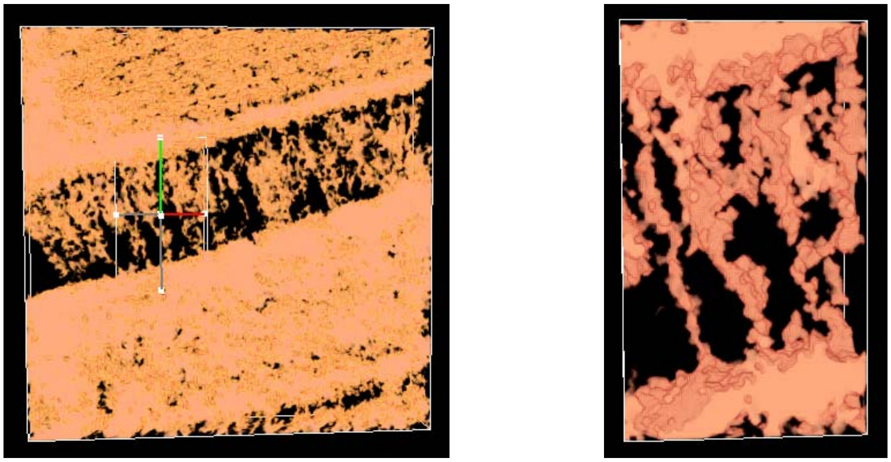


Figure 6: Boundary segmentation of the densely grouped extracellular fibrillar proteins. The original volume is first filtered and then the contrast is enhanced before we apply boundary segmentation. The left picture shows the overall volume while the right one gives a closer look at the segmented boundaries.

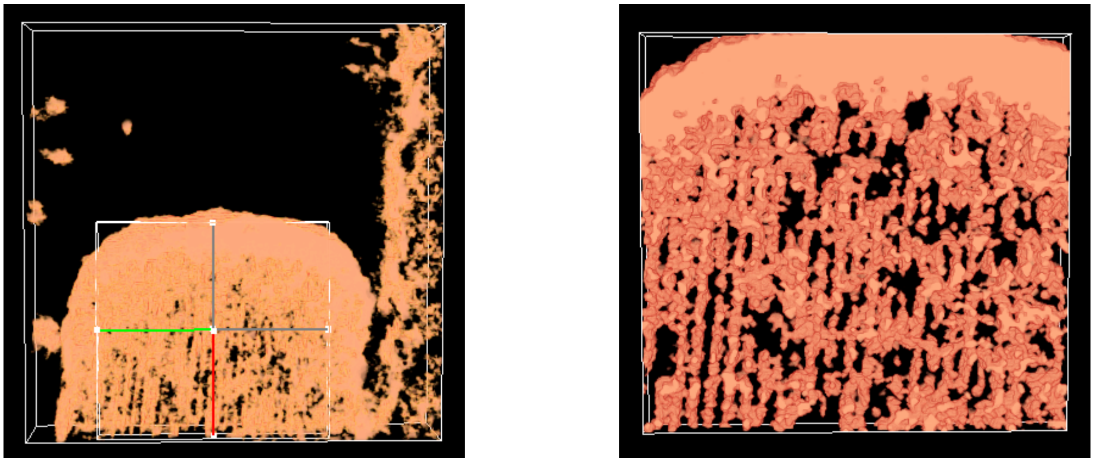


Figure 7: Boundary segmentation of the intracellular actin-filaments (actin bundle). The original volume is first filtered and then the contrast is enhanced before we apply boundary segmentation. The left picture shows the overall volume while the right one gives a closer look at the segmented boundaries.

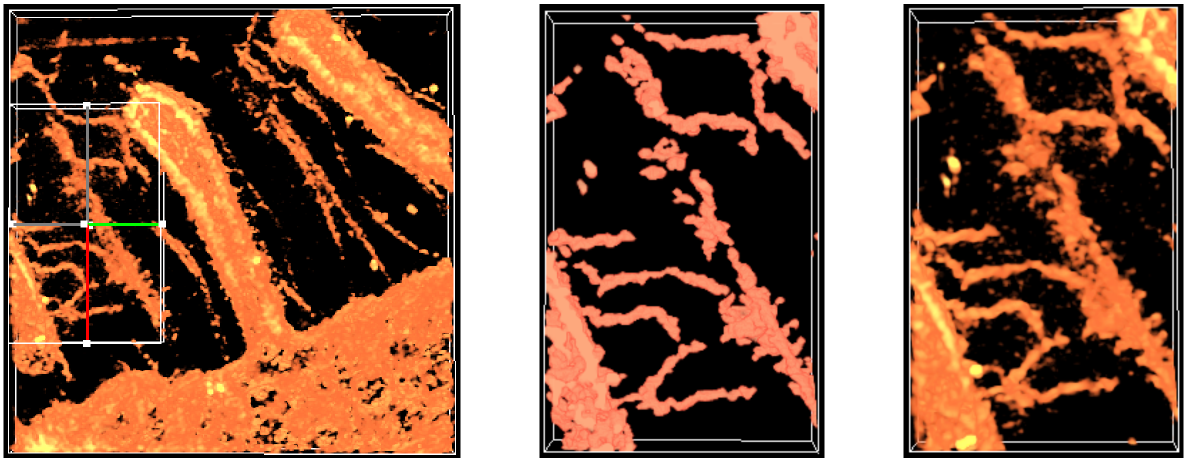


Figure 8: Boundary segmentation of extracellular filaments being secreted from frog saccular sensory epithelium supporting cells. The original volume is filtered before we apply boundary segmentation. The left picture shows the overall volume while the middle one gives a closer look at the segmented boundaries of the sub-volume as indicated in the left figure. As a comparison, the right picture shows the volume-rendered result of the same region.

structures that may ”hide” behind one another in the volume-rendering, we only show a ”thin” slice (with thickness of about 10 voxels), from which we can clearly see the actin-filaments and the links between them.

5 Conclusion

We have presented algorithms for fully automatic boundary segmentation and skeletonization, and have successfully applied them to cell and molecular tomographic imaging data. These algorithms have been implemented in C and are part of our image processing library. We have also developed an interactive volumetric exploration and visualization tool (Volume Rover) which encapsulates implementations of the above filtering, and curve/surface feature extraction algorithms, and additionally uses multi-resolution interactive geometry and volume rendering, for the visualization. Both the interactive visualization tool and our image processing library runs under Linux and Win2K desktop platforms, and are available for free download under the GNU public license, from http://www.ices.utexas.edu/CCV/software.

Acknowledgments

The research of C.Bajaj and Z.Yu was supported in part by NSF grants ACI-9982297, CCR-9988357, and a grant from UCSD 1018140 as part of NSF-NPACI, Interactive Environments Thrust. The research of M.Auer was supported in part by NIH grant DC00241. Thanks are

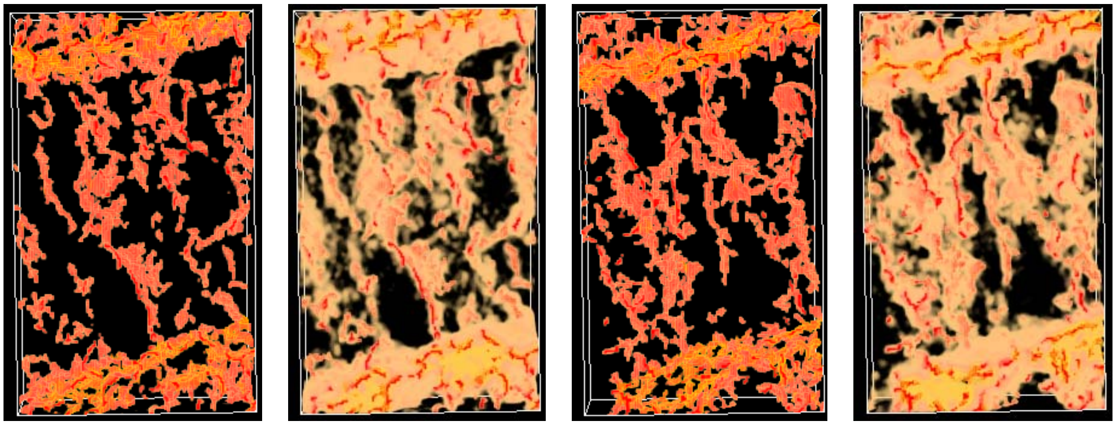


Figure 9: Skeleton extraction of the densely grouped extracellular fibrillar proteins. Left: a close look at the skeletons of a sub-volume of the entire dataset (the same region as shown in Fig.6(left)). Second from left: the skeletons (red) are embedded in the local density map. Third from left: the skeletons of another sub-volume of the same dataset. Right: the skeletons (red) embedded in the local density map.

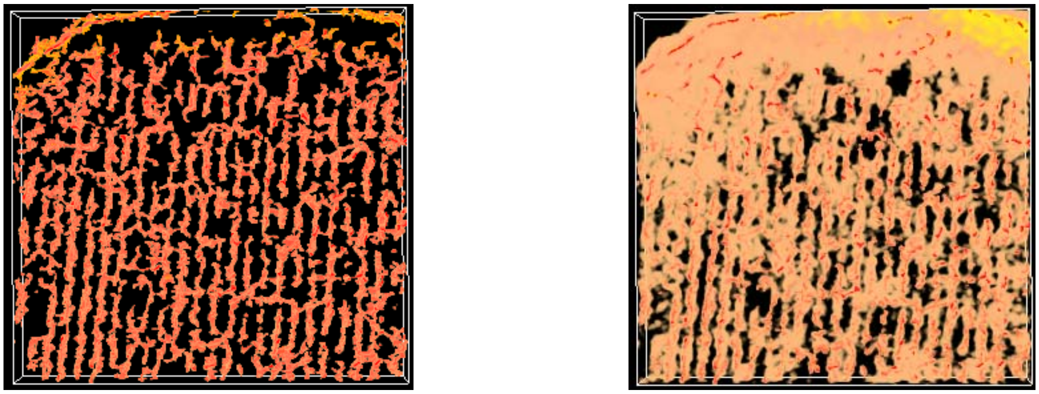


Figure 10: Skeleton extraction of the intracellular actin-filaments (actin bundle). Left: The skeletons of a sub-volume of the overall dataset (roughly in the same region as shown in Fig.7(left)). Right: skeletons (red) embedded in the local density map.

also due to Anthony Thane of CCV for the continued development of the Volume Rover Visualization, and to Dr. Ulrike Ziese and Bram Koster at University of Utrecht for recording the tomographic tilt series data. Manfred Auer would also like to thank Dr. Da Neng Wang (Skirball Institute, NYU), his mentor Dr. Jim Hudspeth as well as the Human Frontier Science Program Organization, and the Jane Coffin Childs Memorial Fund for Medical Research / Agouron Institute for postdoctoral fellowships.

References

[1] B. Alberts. The cell as a collection overview of protein machines: Preparing the next generation of molecular biologists. *CELL*, 92:291–294, 1998.

[2] C. Arcelli and G.S. Baja. Euclidean skeleton via centre-of-maximal-disc extraction. *Image and Vision Computing*, 11(3):163–173, 1993.

[3] M. Auer. Electron cryo-microscopy as a powerful tool in molecular medicine. *J. Molecular Medicine*, 78:191–202, 2000.

[4] C. Bajaj, I. Ihm, and S. Park. 3D RGB image compression for interactive applications. *ACM Transactions on Graphics*, 20(1):10–38, 2001.

[5] C. Bajaj, I. Ihm, and S. Park. Visualization-specific compression of large volume data. In *Proc. of Pacific Graphics*, pages 212–222, Tokyo, Japan, 2001.

[6] C. Bajaj, V. Pascucci, and D.R. Schikore. Fast Isocontouring for Improved Interactivity. In *Proceedings of the 1996 Symposium for Volume Visualization*, pages 39–46, 1996.

[7] C. Bajaj, V. Pascucci, and D. Schikore. The contour spectrum. *Proceedings of IEEE Visualization '97*, pages 167–175, November 1997.

[8] C. Bajaj, V. Pascucci, and D. Schikore. Visualization of scalar topology for structural enhancement. In *IEEE Visualization*, pages 51–58, 1998.

[9] C. Bajaj and D.R. Schikore. Error-bounded reduction of triangle meshes with multivariate data. In *Proceedings of Visual Data Exporation and Analysis III, SPIE vol 2656*, pages 34–45, 1996.

[10] C. Bajaj, Q. Wu, and G. Xu. Level-set based volumetric anisotropic diffusion for 3d image denoising. In *ICES Technical Report, University of Texas at Austin*, 2003.

[11] W. Baumeister, R. Grimm, and J. Walz. Electron tomography of molecules and cells. *Trends Cell Biol.*, 9:81–85, 1999.

[12] W. Baumeister and A. C. Stevens. Macromolecular electron microscopy in the era of structural genomics. *TIBS*, 25:624–631, 2000.

[13] J. Bohm, Frangakis, R. Hegerl, S. Nickell, D. Typke, and W. Baumeister. Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. *Proc Natl Acad Sci.*, 97:14245–14250, 2000.

[14] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[15] C. O. de Solorzano, R. Malladi, S.A. Lelivre, and S.J. Lockett. Segmentation of nuclei and cells using membrane related protein markers. *Journal of Microscopy*, 201(3):404–415, 2000.

[16] K. Dierksen, D. Typke, R. Hegerl, A.J. Koster, and W. Baumeister. Towards automatic electron tomography. *Ultramicroscopy*, 40:71–87, 1992.

[17] J. Frank (Ed.). *Electron Tomography.* Plenum Press, 1992.

[18] R.J. Ellis. Macromolecular crowding: obvious but underappreciated. *Trends Biochem. Sci.*, 26(10):597–604, 2001.

[19] A. S. Frangakis and R. Hegerl. Segmentation of two- and three-dimensional data from electron microscopy using eigenvector analysis. *Journal of Structural Biology*, 138(1-2):105–113, 2002.

[20] R. Glaeser. Electron crystallography: present excitement, a nod to the past, anticipating the future. *J. Struct. Biol.*, 128:3–14, 1999.

[21] T. Grogorishin, G. Abdel-Hamid, and Y.H. Yang. Skeletonization: An electrostatic field-based approach. *Pattern Analysis and Application*, 1(3):163–177, 1996.

[22] M.L. Harlow, D. Ress, A. Stoschek, R.M. Marshall, and U.J. McMahan. The architecture of active zone material at the frog's neuromuscular junction. *Nature*, 409:479 – 484, 2001.

[23] R. Hart. Electron microscopy of unstained biological material: The polytropic montage. *Science*, 159:1464–1467, 1968.

[24] D. Hessler, S. J. Young, and M. H. Ellisman. A flexible environment for the visualization of three-dimensional biological structures. *Journal of Structural Biology*, 116(1):113–119, 1996.

[25] W. Hoppe, J. Gassmann, N. Hunsmann, H.J. Schramm, and M. Sturm. Three-dimensional reconstruction of individual negatively stained yeast fatty-acid synthetase molecules from tilt series in the electron microscope. *Hoppe-Seyler's Z. Physiol. Chem.*, 355:1483–1487, 1974.

[26] T.A. Jones, J-Y. Zou, S.W. Cowan, and M. Kjeldgaard. Improved methods for building protein models in electron density maps and the location of errors in these models. *Acta Crystallogr.*, 409:110–119, 1991.

[27] R. Kimmel, D. Shaked, N. Kiryati, and A.M. Bruckstein. Skeletonization via distance maps and level sets. *Computer Vision and Image Understanding*, 62(3):382–391, 1995.

[28] A.J. Koster, H. Chen, J.W. Sedat, and D.A. Agard. Automated microscopy for electron tomography. *Ultramicroscopy*, 46:207–227, 1992.

[29] J.R. Kremer, D.N. Mastronarde, and J.R. McIntosh. Computer visualization of three-dimensional image data using imod. *J Struct Biol*, 116:71–76, 1996.

[30] L. Lam, S.W. Lee, and C.Y. Suen. Thinning methodologies - a compresensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992.

[31] Y. Li, A. Leith, and J. Frank. Tinkerbell-a tool for interactive segmentation of 3d data. *Journal of Structural Bioloby*, 120(3):266–275, 1997.

[32] G. Malandain and S.F. Vidal. Euclidean skeletons. *Image and Vision Computing*, 16(5):317–327, 1998.

[33] R. Malladi and J.A. Sethian. A real-time algorithm for medical shape recovery. In *IEEE International Conference on Computer Vision*, pages 304–310, 1998.

[34] M. Marko and A. Leith. Sterecon - three-dimensional reconstructions from stereoscopic contouring. *Journal of Structural Biology*, 116(1):93–98, 1996.

[35] B.F. McEwen, K.H. Downing, and R.M. Glaeser. The relevance of dose-fractionation in tomography of radiation-sensitive specimens. *Ultramicroscopy*, 60:357–373, 1995.

[36] B.F. McEwen and J. Frank. Electron tomographic and other approaches for imaging molecular machines. *Curr. Op. Neurobiol.*, 11:594–600, 2001.

[37] B.F. McEwen and M. Marko. Three-dimensional electron micros-copy and its application to mitosis research. *Methods Cell Biol*, 61:81–111, 1999.

[38] B.F. McEwen and M. Marko. The emergence of electron tomography as an important tool for investigating cellular ultrastructure. *J. Histochem Cytochem*, 49:553–563, 2001.

[39] J. Milnor. *Morse Theory.* Princeton University Press, 1963.

[40] R.L. Ogniewicz and O. Kubler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

[41] S. Osher and N. Paragios (editor). *Geometric level set methods in imaging, vision, and graphics.* United States Springer Vlg., 2003.

[42] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[43] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L.S. Avila, K. Martin, R. Machiraju, and J. Lee. The transfer function bake-off. *IEEE Transactions on Computer Graphics and Applications*, 21(3):16–22, 2001.

[44] S. Röttger, M. Kraus, and T. Ertl. Hardware-accelerated volume and isosurface rendering based on cell-projection. In *Proceedings of IEEE Visualization '00*, pages 109–116, 2000.

[45] J.A. Sethian. A marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.*, 93(4):1591–1595, 1996.

[46] J.A. Sethian. *Level Set Methods and Fast Marching Methods, 2nd edition.* Cambridge University pPress, 1999.

[47] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *International Conference on Computer Vision and Pattern Recognition*, pages 136–142, 1996.

[48] E. Sifakis and G. Tziritas. Fast marching to moving object location. In *Proc. 2nd Intl. Conf. on Scale-Space Theories in Computer Vision*, pages 447–452, 1999.

[49] E. Sifakis and G. Tziritas. Moving object localization using a multi-label fast marching algorithm. *Signal Processing: Image Communication*, 16(10):963–976, 2001.

[50] B.-S. Sohn, C. Bajaj, and V. Siddavanahalli. Feature based volumetric video compression for interactive playback. In *Preceedings of the IEEE/ACM Siggraph 2002 Symposium for Volume Visualization and Graphics*, pages 89–96, 2002.

[51] K.A. Taylor, H. Schmitz, M.C. Reedy, Y.E. Goldman, C. Franzini-Armstrong, H. Sasaki, R.T. Tregear, K. Poole, C. Lucaveche, R.J. Edwards, L.F. Chen, H. Winkler, and M.K. Reedy. Tomographic 3d reconstruction of quick-frozen, ca2+-activated contracting insect flight muscle. *Cell*, 99:421–431, 1999.

[52] M. van Heel, B. Gowen, R. Matadeen, E.V. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan. Single-particle electron cryo-microscopy: towards atomic resolution. *Q Rev Biophys*, 33:307–369, 2000.

[53] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Symposium on Computational Geometry*, pages 212–220, 1997.

[54] N. Volkmann and D. Hanein. Quantitative fitting of atomic models into observed densities derived by electron microscopy. *Journal of Structural Biology*, 125(2-3):176–184, 1999.

[55] N. Volkmann. A novel three-dimensional variant of the watershed transform for segmentation of electron density maps. *Journal of Structural Biology*, 138(1-2):123–129, 2002.

[56] J. Weickert. *Anisotropic Diffusion In Image Processing.* ECMI Series, Teubner, Stuttgart, ISBN 3-519-02606-6, 1998.

[57] R. Westermann and E. Thomas. Efficiently using graphics hardware in volume rendering applications. In *In SIGGRAPH '98*, pages 169–177, 1998.

[58] W. Wriggers and S. Birmanns. Using situs for flexible and rigid-body fitting of multiresolution single-molecule data. *J. Struct. Biol.*, 133:193–202, 2001.

[59] W. Wriggers, R. A. Milligan, and J. A. McCammon. Situs: A package for docking crystal structures into low-resolution maps from electron microscopy. *Journal of Structural Biology*, 125(2-3):185–195, 1999.

[60] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Processing*, 7(3):359–369, 1998.

[61] Z. Yu and C. Bajaj. Anisotropic vector diffusion in image smoothing. In *Proceedings of International Conference on Image Processing*, pages 828–831, 2002.

[62] Z. Yu and C. Bajaj. Image segmentation using gradient vector diffusion and region merging. In *Proceedings of International Conference on Pattern Recognition*, pages 941–944, 2002.

[63] Z. Yu and C. Bajaj. Normalized gradient vector diffusion and image segmentation. In *Proceedings of European Conference on Computer Vision*, pages 517–530, 2002.

[64] Y. Zhou and A. Toga. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5:196–209, 1999.