Segmentation-free Skeletonization of Grayscale Volumes for Shape Understanding

Sasakthi S. Abeysinghe*
Washington University in St. Louis

Matthew Baker†
Baylor College of Medicine

Wah Chiu‡
Baylor College of Medicine

Tao Ju§
Washington University in St. Louis

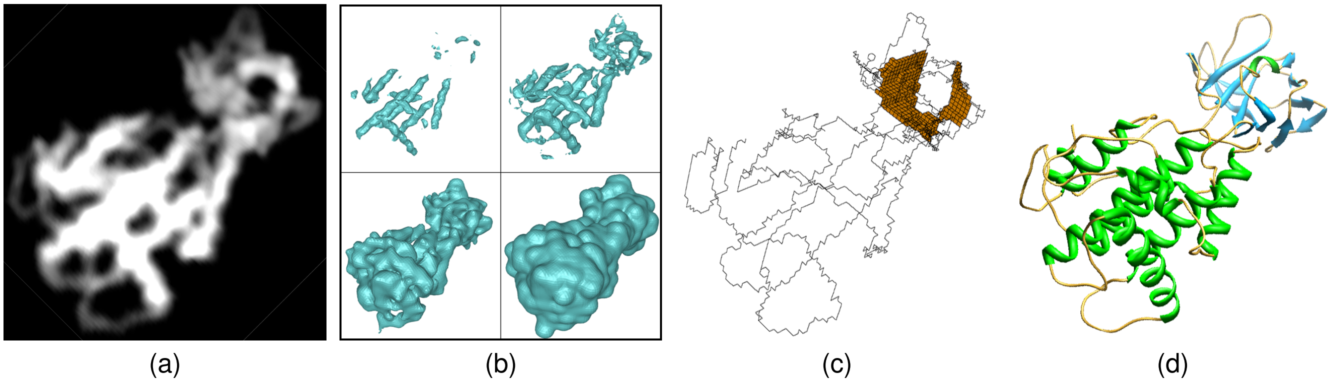

Figure 1: A grayscale volume of a protein molecule (a), the segmentation at various thresholds (b), the skeleton generated by our segmentation-free method (c), and the ground-truth structure of the protein (d).

Abstract

Medical imaging has produced a large number of volumetric images capturing biological structures in 3D. Computer-based understanding of these structures can often benefit from the knowledge of shape components, particularly rod-like and plate-like parts, in such volumes. Previously, skeletons have been a common tool for identifying these shape components in a solid object. However, obtaining skeletons of a grayscale volume poses new challenges due to the lack of a clear boundary between object and background. In this paper, we present a new skeletonization algorithm on grayscale volumes typical to medical imaging (e.g., MRI, CT and EM scans), for the purpose of identifying shape components. Our algorithm does not require an explicit segmentation of the volume into object and background, and is capable of producing skeletal curves and surfaces that lie centered at rod-shaped and plate-shaped parts in the grayscale volume. Our method is demonstrated on both synthetic and medical data.

Keywords: Grayscale Skeletonization; Structure Tensor; Thinning; Pruning

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shape

1 Introduction

Bio-medical imaging techniques, such as MRI, CT and EM, are routinely used to produce a large number of volumetric images. Computer-based understanding of the biological structures, ranging in scale from proteins to organs, from these 3D volumes has become increasingly demanded by research efforts and clinical practices.

Oftentimes such understanding can be obtained by identifying shape components, such as rod-like and plate-like parts, in the 3D volume. For example, blood vessels appear as rods in MRI scans, and the cortical shell of bones appear as curved plates in CT scans. Figure 1 (a) shows an example of a 3D volume of a protein molecule obtained using electron cryo-microscopy (cryo-EM) at sub-nanometer resolution. Here the rod-like and plate-like parts correspond to key building blocks of a protein, such as α-helices (appearing in the volume as straight rods), β-sheets (as curved plates), and loops (as curved rods). For comparison, the actual 3D structure of this protein is shown in Figure 1 (d), where the three protein building blocks are shown as green spirals (α-helices), blue arrows (β-sheets) and orange curves (loops).

For solid objects, a typical approach for identifying their rod-like and plate-like shape components is to consider the object's *skeleton* [3]. An ideal skeleton for this purpose would consist of medial curves and medial surfaces lying centered at the rod-like and plate-like parts of the object [16, 4, 9]. Unfortunately, the 3D data produced by medical imaging is usually in the form of a grayscale volume, which lacks a clear boundary between the object and the background. Although an object segmentation can be obtained by some particular threshold gray value, the segmented objects may have widely varying shapes depending on the choice of the threshold (as illustrated in Figure 1 (b)). The skeletons of these segmentations would assume very different shapes, and the skeleton at a particular threshold may not reflect all shape components intrinsic to the grayscale volume.

*e-mail: sasakthi.abeysinghe@wustl.edu
†e-mail:mlbaker@gmail.com
‡e-mail:wah@bcm.tmc.edu
§e-mail:taoju@cs.wustl.edu

IEEE International Conference on Shape
Modeling and Applications 2008
4 - 6 June, Stony Brook, New York, USA
978-1-4244-2261-6/08/$25.00 ©2008 IEEE

1.1 Problem statement

We are interested in computing the skeleton of a grayscale volume for identifying its intrinsic shape components. Instead of depending on the segmentation at some threshold, the skeleton should consist of curves and surfaces that are centered at the rod-like and plate-like parts of the full, un-segmented grayscale volume.

In contrast to the vast literature on skeletonization of solid models, computing skeletons on un-segmented 3D volumes has received much less attention (see review in Section 2). In particular, we know of no existing method capable of extracting both skeletal curves and surfaces from a grayscale volume for the purpose of shape understanding without specific domain knowledge or an object segmentation.

One aspect of this skeletonization problem that requires further clarification is what constitutes a rod-like part or a plate-like part in a grayscale volume. In these volumes, the gray values behave like a density distribution, where voxels with higher values are likely to be located closer to the center of the imaged subject. In addition, different parts of the subject may exhibit different brightness levels. Many bio-medical imaging techniques (i.e. MRI, CT, EM) produce volumes which have features that satisfy these observations. However, different imaging modalities (ex: T1-weighted and T2-weighted MRI scans) capture different features of the subject being imaged, and thus what is characterized as a rod-like part or a plate-like part will vary based on the imaging modality. We explain our observations using a synthetic Hand volume in Figure 2(a), which contains 5 rod-like parts (the fingers) and 1 plate-like part (the palm):

Observation 1: A shape component, such as a rod or a plate, in a density-like volume is usually captured by the segmented object at *some* threshold values. For example, the top four fingers in the Hand volume appear as rods in the segmentation at one threshold (b), while the palm forms a plate at a different threshold (c). However, this observation alone is not sufficient to disambiguate the different types of shape components located in the same part of the volume. For example, the top four fingers also form a plate in the segmentation (c).

Observation 2: The center of a shape component usually lies at the "ridge" of the underlying density function. In particular, the *variation* of gray values at a ridge point is usually smaller along the ridge than in other directions. For example, the grayscale variation along the center line of each finger is much smaller than along other directions (as illustrated in the close-up view in (d)). In contrast, a plate-like shape formed by the top four fingers would not have this property, as the grayscale variation along the center surface of this plate can be much greater than in some other direction, especially between two fingers (as illustrated in the cross-section view in (e)).

1.2 Method

Our algorithm extracts the skeleton of a density-like grayscale volume guided by the above two principle observations. There are two main stages in this algorithm. In the first stage, the algorithm visits all possible threshold values and identifies the shape components on the segmented object at each threshold by their center curves and surfaces. According to Observation 1, the collection of these shape components on various segmentations is a super-set of those intrinsic to the grayscale volume, the latter of which, by Observation 2, are characterized by the ridge-like centers in the density distribution. Hence, the second stage extracts the final skeleton as the sub-set of curves and surfaces generated in the first stage that exhibit small grayscale variation.

An example result of our method is shown in Figure 1 (c). Note that the curves (black lines) and surfaces (orange faces) in the skeleton correspond well to α-helices and loops (which appear as rods in the volume) as well as β-sheets (which appear as plates in the volume) in the actual protein structure in (d).

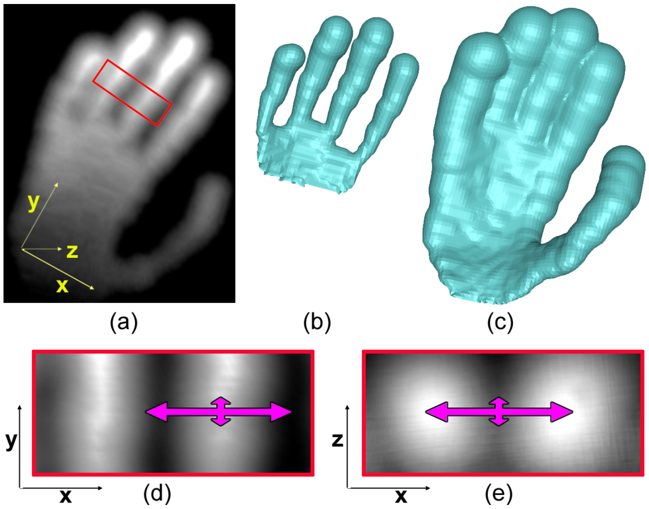

Figure 2: A grayscale volume (a), segmented surfaces at two thresholds (b,c), and close-up view of two fingers at different angles (d,e) where the pink arrows illustrate the magnitude of grayscale variation in different directions. See Section 1.1.

1.3 Contribution

The primary contribution of this paper is a novel algorithm for computing the skeleton of a density-like grayscale volume. The algorithm does not require segmentation at any particular threshold, and the resulting skeleton consists of curves and surfaces centered at rod-like and plate-like parts of the grayscale volume. The independence from a threshold makes the skeletonization process less sensitive to human bias and allows for the understanding of the intrinsic shapes in such a volume. The method is demonstrated on both synthetic and medical data.

2 Previous work

Here we briefly review skeletonization methods for 3D solid and grayscale models, with an eye towards identifying shape components.

Solid models: Computing skeletons of 3D solid models has been extensively researched in the past. A number of representative methods include morphological thinning [2, 15, 20], distance transforms [5], potential field methods [1], and Voronoi diagrams [18, 6]. For the purpose of identifying shape components, Saha *et al.* [16] and Bonnassie *et al.*[4] differentiates curves and surfaces in the skeletons generated by morphological thinning by classifying skeleton voxels based on their local neighborhood . While the result of such classification can be highly sensitive to the quality of the skeleton, the method of Ju *et al.* [9] directly extracts skeletal curves and surfaces during the thinning process without need for post-classification. These methods have been used to classify rod-like and plate-like structures in bone matrices [16, 4] and proteins [9].

Grayscale volumes: In contrast, few work has addressed skeletonization of un-segmented grayscale volumes. Although the use of morphological thinning has been well-studied in the vision community for skeletonizing 2D grayscale images (see the excellent survey by Mersal and Darwish [14]), extensions to 3D volumes have been rare. Segmentation techniques (see surveys [17, 8, 21]) have been used to build solid models of grayscale volumes; however, skeletons computed from these models are medial to the segmentation and do not align well with the high density regions of the grayscale volume. In contrast the method of Svensson *et al.* [19] generates skeletal surfaces starting from a known object segmentation, but utilizes the interior grayscale information. Similar to ours, the method of Doklada *et al.* [7] computes an initial skeleton by thinning on the full grayscale volume, but it then requires a grayscale threshold to remove insignificant skeleton parts and is designed only for skeletal curves.

In a different approach, Lopez *et al.* [12] identifies centers of a grayscale distribution using a multi-local creaseness measure, continuing a body of research on ridge and valley detection in 2D images (see the survey and evaluation in [13]). However, Lopez's method results in a collection of center points that lack any curve or surface structure necessary for identifying shape components.

Several researchers have proposed to explicitly extract both curves and surfaces in a grayscale volume based on the second-order tensor field of the volume [23, 11, 22]. However, these methods are either designed for visualizing flow anisotropy [23, 11] rather than locating shape components, or require domain-specific knowledge to find those curves or surfaces at the center of the shape components [22]. In contrast, our method relies on the same tensor field for extracting curve and surface geometry but is capable of placing such geometry at the center of grayscale shape components without the use of application-domain specific knowledge.

3 Overview

3.1 Data representation

Our algorithm takes in a volume represented as a 3D rectilinear grid, where each grid point, called a *voxel*, is associated with a grayscale value. The output skeleton of our algorithm consists of a subset of voxels on this grid. Figure 3 shows examples of grids and voxels where skeleton voxels are colored gray.

For the purpose of shape understanding, we define two types of geometry, curves and surfaces, on a set of voxels. A curve is a collection of *voxel edges*, each consisting of two voxels lying on the ends of a grid edge. A surface is a collection of *voxel faces*, each consisting of four voxels lying on the corners of a grid face (i.e. a grid face surrounded by four voxel edges). For example, the skeleton voxels on the left of Figure 3 form a voxel edge, and those on the right form four voxel edges and a voxel face. In this figure (and other figures in the paper), voxel edges are drawn as black lines and voxel faces as orange quads.

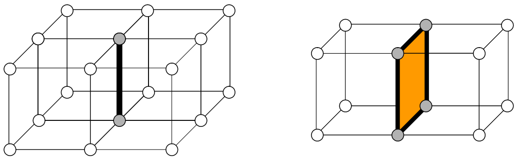

Figure 3: Examples of grids with voxels (circled dots) where skeleton voxels (gray) form edges (black lines) and a face (orange quad).

3.2 The algorithm

Given a density-like grayscale volume, our algorithm, guided by the observations in Section 1.1, extracts skeletons consisting of curves and surfaces corresponding to the rod-like and plate-like parts of the volume. Since each shape component in the volume is captured by the segmented object at some threshold value, we first identify the set of all shape components at a range of threshold values. This is done by accumulating the skeletal curves and surfaces computed from the segmented objects at each threshold value. The sub-set of this initial skeleton that represents the grayscale shape components are then identified as those curves and surfaces exhibiting small grayscale variations. This is done by comparing the directions of these geometric elements with the shape of grayscale variation in a local neighborhood of each voxel.

In order to extract two types of skeletal geometry, namely curves and surfaces, we device a four-step flow that first extracts the skeletal surfaces followed by the skeletal curves. The generation of each type of skeletal geometry follows the same stages of initial skeleton generation and skeleton pruning, as illustrated in Figure 4:

- **Step 1: Initial skeletonization.** Accumulate the skeletal surfaces of each segmentation of the grayscale volume at a range of thresholds.
- **Step 2: Pruning.** Identifies those surfaces in the result of Step 1 that exhibit small grayscale variations.
- **Step 3: Initial skeletonization.** Accumulate the skeletal curves of each segmentation of the *original* grayscale volume at the same range of thresholds.
- **Step 4: Pruning.** Identifies those curves in the result of Step 3 that exhibit small grayscale variations.

To ensure accurate identification of surface and curve features, surface skeletonization and pruning has to be done before curve skeletonization and pruning. This restriction arises from the fact that a curve can be geometrically defined as a subset of a surface, and performing curve skeletonization (and pruning) on a feature which is actually a surface (but not yet classified as a surface) will result in that feature being incorrectly classified as a curve.

In the following sections, we will describe the skeletonization and pruning steps in details.

4 Initial skeletonization

Given a segmented object at a particular threshold, which is made up of voxels on a grid, a classical approach of obtaining its skeleton is morphological thinning. To be able to identify shape components such as rods and plates, we consider the iterative thinning approach of [9], which selectively generates curves or surfaces (as defined in Section 3.1) centered at these parts. Briefly, this method shrinks the object to its medial structure by iteratively removing its border voxels. Skeletal curves or surfaces are identified by preserving voxels at the ends of either curves or surfaces during thinning.

To accumulate the skeletal surfaces or curves computed at multiple thresholds, we make a small modification to the method of [9] to utilize skeletons generated at different thresholds. Specifically, we segment the volume with decreasing threshold values. At each threshold, we compute the skeletal surfaces (in Step 1) or curves (in Step 3) by thinning the segmented object while additionally preserving, at each thinning iteration, the voxels belonging to the skeletons generated at previous thresholds. This incremental approach, combined with iterative thinning, ensures that skeletons computed at lower thresholds are aligned with skeletons at higher thresholds, and hence to regions with high gray values, which are likely to be centers of the shape components in the grayscale volume. The results of this incremental thinning for the Hand example in Figure 4 (a) are shown in (b,d).

In our implementation, the range of threshold values is taken as the full range of gray values in the volume, unless the user specifies a maximum and/or minimum gray value of interest. As enumerating each gray value present in the volume within the range can be time-consuming, we may also use values at discrete intervals. In all our examples, we discretize an input threshold range into 256 levels and visit each level in descending order.

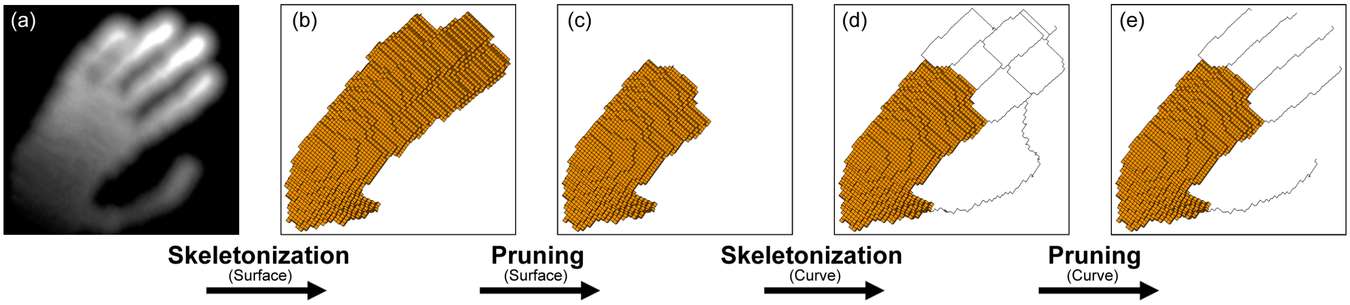

Figure 4: The four steps in our algorithm. See Section 3.2.

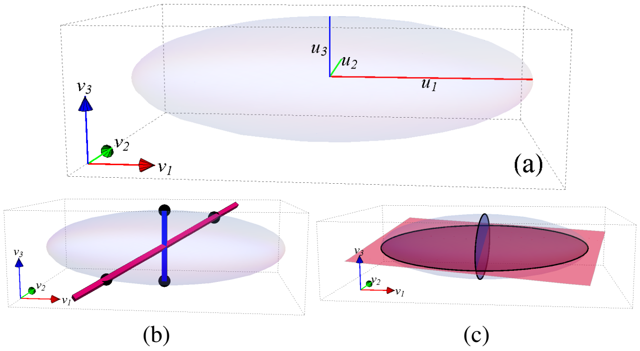

Figure 5: (a): Grayscale variation visualized as an ellipsoid. (b): Scoring of a line (red) as ratio between length of the shortest axis (blue) over that of the line segment in the ellipsoid. (c): Scoring of a plane (red) as ratio between area of the smallest axial ellipse (blue) over that of the cross-sectional ellipse.

5 Pruning

The initial skeleton generated by the previous step (e.g., Figure 4 (b,d)) contains a super-set of skeletal surfaces or curves that represent the actual shape components of the grayscale volume. Based on our earlier observation, the desired sub-set of surfaces or curves are those along which the grayscale variation is smaller than along other directions. We will identify this sub-set in two phases. First, we will compute a score at each skeleton voxel based on the direction of the skeletal surface or curve at that voxel with respect to the shape of the local grayscale variation. Next, we will extract well-formed surfaces or curves consisting of high-score voxels.

5.1 Scoring

Structure tensor: While it is possible to measure the grayscale variation at each voxel by its local gradient, such measurement easily becomes unreliable in the presence of noise, which is typical in medical volumes. Instead, the structure tensor offers an average measurement of such variation within a neighborhood of each voxel, which is much more robust under noisy conditions. Specifically, we first compose a tensor T' at a voxel p as a 3×3 matrix:

$$T'_p = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} \times \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix}^T = \begin{bmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & I_z^2 \end{bmatrix} \qquad (1)$$

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_z = \frac{\partial I}{\partial z}$ are the partial derivatives in the x, y and z directions of the grayscale volume I at p. Performing spatial averaging of these tensors in the neighborhood of p using a Gaussian convolution mask g_σ with standard deviation σ gives the *structure tensor* T_p:

$$T_p = g_\sigma * T'_p. \qquad (2)$$

The key piece of information offered by the structure tensor T_p is its Eigen-structure, which reveals the principal directions and magnitudes of grayscale variation in the neighborhood of p. As shown in Figure 5 (a), the "shape" of this variation can be visualized as an ellipsoid whose axes are along the eigenvectors of T_p with the magnitude of the corresponding eigenvalues. Intuitively, the gray values around the voxel p vary more dramatically along directions closer to the major axis of the ellipsoid (i.e., the eigenvector with the largest eigenvalue), and less along directions closer to the minor axis (i.e., the eigenvector with the smallest eigenvalue).

Scoring surfaces and curves The ellipsoidal representation of the grayscale variation offers an intuitive way of measuring the variation in any given direction. For example, the grayscale variation along a given line can be measured as the length of the line segment within the ellipsoid (Figure 5 (b)). Since we are more interested in whether such variation is *smaller* than variations in other directions, we can score a voxel on a skeletal curve by the *ratio* of the minimum length of such line segments (i.e., the shortest axis of the ellipsoid) over the actual length along the tangent line of the curve. Likewise, we can score a voxel on a skeletal surface by the ratio of the minimum area of a cross-section in the ellipsoid (i.e. formed by the two shortest axes) over the actual area of the cross-section along the tangent plane of the surface (Figure 5 (c)).

Specifically, denote the eigenvectors and eigenvalues of the structure tensor T_p by $\{v_1, v_2, v_3\}$ and $\{u_1, u_2, u_3\}$. As T_p is a positive semi-definite matrix $u_1 \geq u_2 \geq u_3 \geq 0$. Given a line passing through the origin in the unit direction of $c = \{c_x, c_y, c_z\}$, the length of the line segment within the ellipsoid is

$$L(c) = \frac{u_1 u_2 u_3}{\sqrt{u_2^2 u_3^2 c_x^2 + u_1^2 u_3^2 c_y^2 + u_1^2 u_2^2 c_z^2}}. \qquad (3)$$

Given a plane passing through the origin defined by two orthogonal unit vectors $n_1 = \{n_{1x}, n_{1y}, n_{1z}\}$ and $n_2 = \{n_{2x}, n_{2y}, n_{2z}\}$ on the plane, the area of the cross-section of the ellipsoid (which is an ellipse) is computed as:

$$A(n_1, n_2) = \frac{\pi}{|m_1 \cos\theta + m_2 \sin\theta| \times |m_1 \sin\theta - m_2 \cos\theta|} \qquad (4)$$

where,

$$m_1 = \left\{ \frac{n_{1x}}{u_1}, \frac{n_{1y}}{u_2}, \frac{n_{1z}}{u_3} \right\},$$

$$m_2 = \left\{ \frac{n_{2x}}{u_1}, \frac{n_{2y}}{u_2}, \frac{n_{2z}}{u_3} \right\},$$

$$\theta = \frac{\arctan \frac{2 m_1 \cdot m_2}{m_1 \cdot m_1 - m_2 \cdot m_2}}{2}.$$

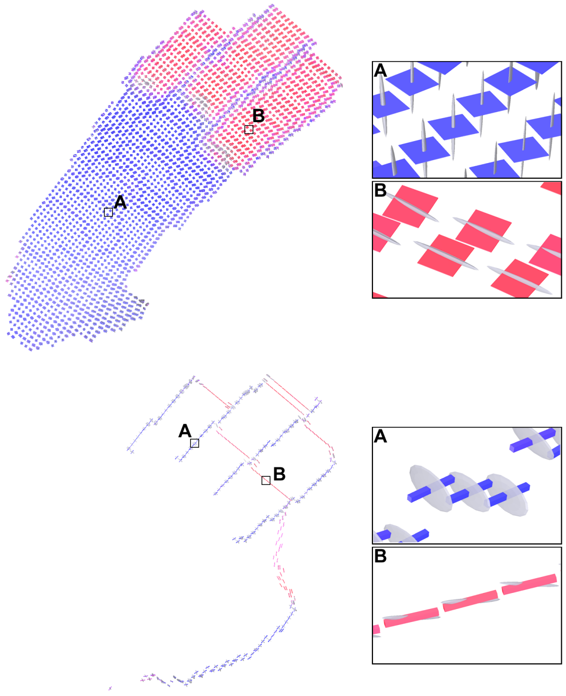

Figure 6: Voxel scores on the skeletal surfaces (top) and curves (bottom) in Figure 4 (b,d), showing the ellipsoidal representation of grayscale variations and the tangent orientation of surfaces and curves. Blue and red indicate high and low scores.

The score of a skeletal curve with tangent vector c is therefore the ratio $L(v_3)/L(c)$, and the score of a skeletal surface with tangent vectors n_1, n_2 is the ratio $A(v_2, v_3)/A(n_1, n_2)$. Note that the score is bounded between $[0,1]$, where 1 corresponds to the direction of minimum grayscale variation locally at p. To avoid possible numerical instability when evaluating scoring functions for very small values of u_1, u_2 and u_3, we note the limit of these functions are well defined when one or more of the eigenvalues approach zero, which is detailed in Appendix A. In practice, we treat any eigenvalue smaller than a threshold (such as 0.00001) as zero and directly apply the limit formula.

Due to the use of a rectilinear grid, the tangent orientation of the skeletal surface or curve at a voxel, if computed locally, will assume a limited number of directions restricted by the axes-aligned voxel faces and edges (i.e. six if using N6 connectivity). To overcome this limitation, we obtain these orientations by computing a best-fitting line or plane to all voxel faces or edges in a neighborhood of the voxel p.

In Figure 6, we show the scores computed for the surfaces and curves resulting from the initial skeletonization steps in Figure 4 (b,d). Note that lower scores (colored red) effectively indicate skeletal geometry (e.g., surfaces and curves between the fingers) that do not correspond well to actual shape components in the grayscale volume.

5.2 Feature extraction

Given the scoring of skeleton voxels (e.g., Figure 6), we next need to identify pieces of surfaces (in Step 2) or curves (in Step 4) consisting of high-scored voxels. Ideally, the final skeleton should consist of clean, recognizable surfaces and curves that are free of extraneous features such as small branches and islands. To this end, we first remove all voxels that score below a threshold. We find that the threshold of $1/\sqrt{3}$ (the ratio of the edge length over the diagonal length of a unit cube) works well for both surfaces and curves. Next, we utilize the morphological opening operator in [9] designed for skeletal curves and surfaces to remove extraneous skeleton features. Given user-specified size parameters $\varepsilon_s, \varepsilon_c$, this operator removes surface branches with radius smaller than ε_s and curve branches shorter than ε_c. The final results of skeleton pruning for the hand example are shown in Figure 4 (c,e).

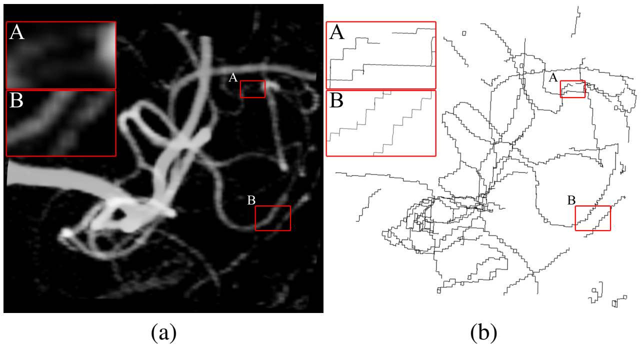

Figure 7: An MRI scan of blood vessels in the human head (a), the skeleton generated by our algorithm (b) that captures thin blood vessels which are barely visible to the naked eye (A, B).

As shown in [9], the choice of $\varepsilon_s, \varepsilon_c$ controls the minimum size of the surface or curve feature in the final skeleton. This number typically only depends on the grid resolution and the type of subject being imaged. In our experiments, we use $\varepsilon_s = \varepsilon_c = 5$ except for imaged subjects made up of only rod-like parts (e.g., blood vessels), where we set $\varepsilon_s = \infty$, and subjects made up of only plate-like parts (e.g., cortical bones), where we set $\varepsilon_c = \infty$.

6 Results

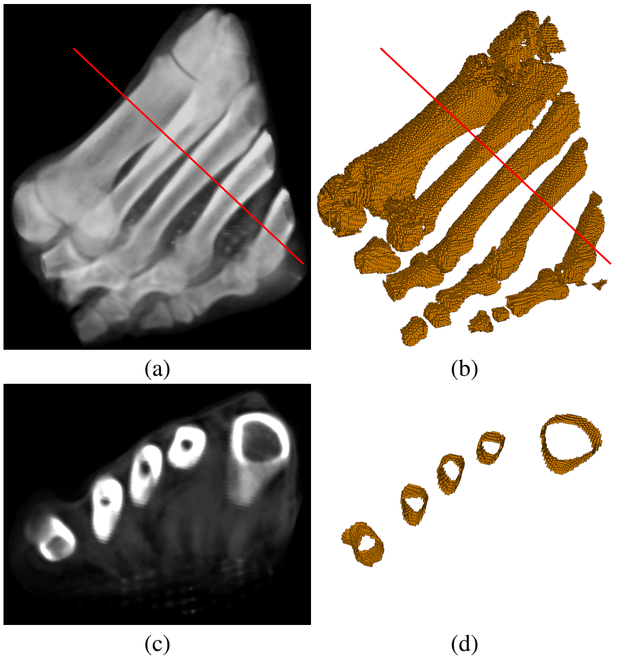

Figure 9: A CT scan of a human foot (a), the skeleton generated by our algorithm (b) that captures the cortical bones as surfaces, and a cross-section view (c,d).

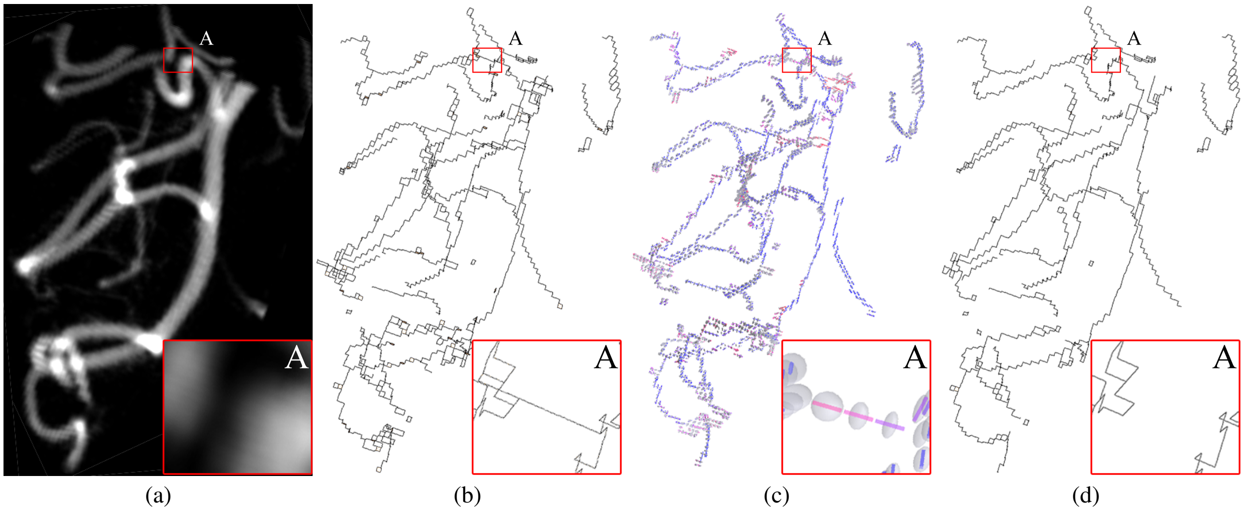

Figure 8: A CT scan of blood vessels in the human head (a), the skeletal curves obtained after initial skeletonization (Step 3) (b) and pruning (Step 4) (d), and the voxel scoring during pruning (c).

Data Set	Dimensions	Step 1		Step 2		Step 3		Step 4		Total
	(voxels)	Time	Voxels	Time	Voxels	Time	Voxels	Time	Voxels	Time
Hand	$129 \times 129 \times 129$	4.12	3640	4.40	1981	3.85	2331	1.65	2202	14.04
Protein 2ITG	$64 \times 64 \times 64$	6.76	4736	3.18	142	6.79	1198	0.73	659	17.48
Protein 1TIM	$96 \times 96 \times 96$	16.87	8954	5.93	978	17.04	3067	1.75	1735	41.61
Protein 1BTV	$128 \times 128 \times 128$	34.29	12232	8.48	747	34.76	3777	2.73	1910	80.27
Blood Vessels (CT)	$121 \times 71 \times 66$	11.67	6608	0.59	0	11.78	4737	1.85	1757	25.90
Blood Vessels (MRI)	$101 \times 82 \times 111$	11.34	10313	0.71	0	11.41	7753	2.75	2662	26.23
Bones	$150 \times 128 \times 128$	33.50	143617	76.29	78178	29.67	105708	9.98	78178	149.44

Table 1: Time taken (in seconds) for each step of the algorithm (see Section 3.2) and the number of skeleton voxels after each step.

We demonstrate our algorithm on a set of medical data produced by MRI, CT and cryo-EM imaging, where the biological structure of interest consists of rod-like and/or plate-like components.

Figure 7 shows the results of our method on an MRI scan of blood vessels in the human head. Observe from the close-up views, that, without relying on a particular threshold value, our technique was able to capture vessels at a wide range of gray levels and thicknesses, some of which are not even visible to the naked eye.

Figure 8 shows both the intermediate and final results of our method on a CT scan of blood vessels. The usefulness of pruning based on grayscale variations is illustrated in the close-up view between two vessels, where a skeletal curve is generated during initial skeletonization (as the segmented surface is connected at a low gray level), but receives a low score (b) as the curve exhibits a large grayscale variation, and finally gets removed (d).

Figure 9 shows another example where our method computes a surface skeleton of cortical bones in a CT scan of the human foot. As seen in the cross-sections (c,d), our technique accurately captures the shell-shape of the cortical bones and preserves their hollow nature. Note that the skeleton is computed independent of any thresholds, and hence is capable of capturing both bright and dark potions of the cortical shell well.

Finally, we present two more examples of skeletonization of protein volumes imaged by cryo-EM in Figure 10. As mentioned earlier, the rod-like and plate-like parts of these volumes correspond well to key building blocks of the protein, including α-helices (rod-like), β-sheets (plate-like) and loops (rod-like). Observe from Figure 10(c) that our method is capable of capturing shape components that correlate well with the actual protein structures shown in (d).

We additionally compare our method with a previous method by Yu *et al.* [22] for computing skeletal curves and surfaces specifically in cryo-EM data. Yu's method also relies on the structure tensor for extracting the skeletal geometry, but requires explicit knowledge about the typical thickness of α-helices and β-sheets as well as their brightness level in order to locate the corresponding rod-like and plate-like parts in the volume. In comparison with the result of Yu's method in Figure 10 (b), our method, without any domain-specific knowledge (i.e. the typical thickness of α-helices and β-sheets, or their brightness levels), additionally extracts skeletal curves that correspond to loop structures in the protein. The difficulty Yu's method faces when identifying loops arises from the fact that loops lack a uniform thickness and are often at low gray levels in a cryo-EM volume.

All experiments were performed on a PC with a 3GHz Pentium-D CPU and 4GB of memory (our implementation runs on a single thread, thus utilizes only one of the cores of the CPU). Table 1 shows the breakdown of the time for each step in our algorithm. The time complexity of the initial skeletonization process is $O(ng)$, where n is the number of voxels, and g is the number of distinct gray-levels in the grayscale volume. The pruning process has a time complexity of $O(\varepsilon s)$ where s is the number of voxels in the initial skeleton, and ε is the minimum size of the curve or surface feature in the final skeleton.

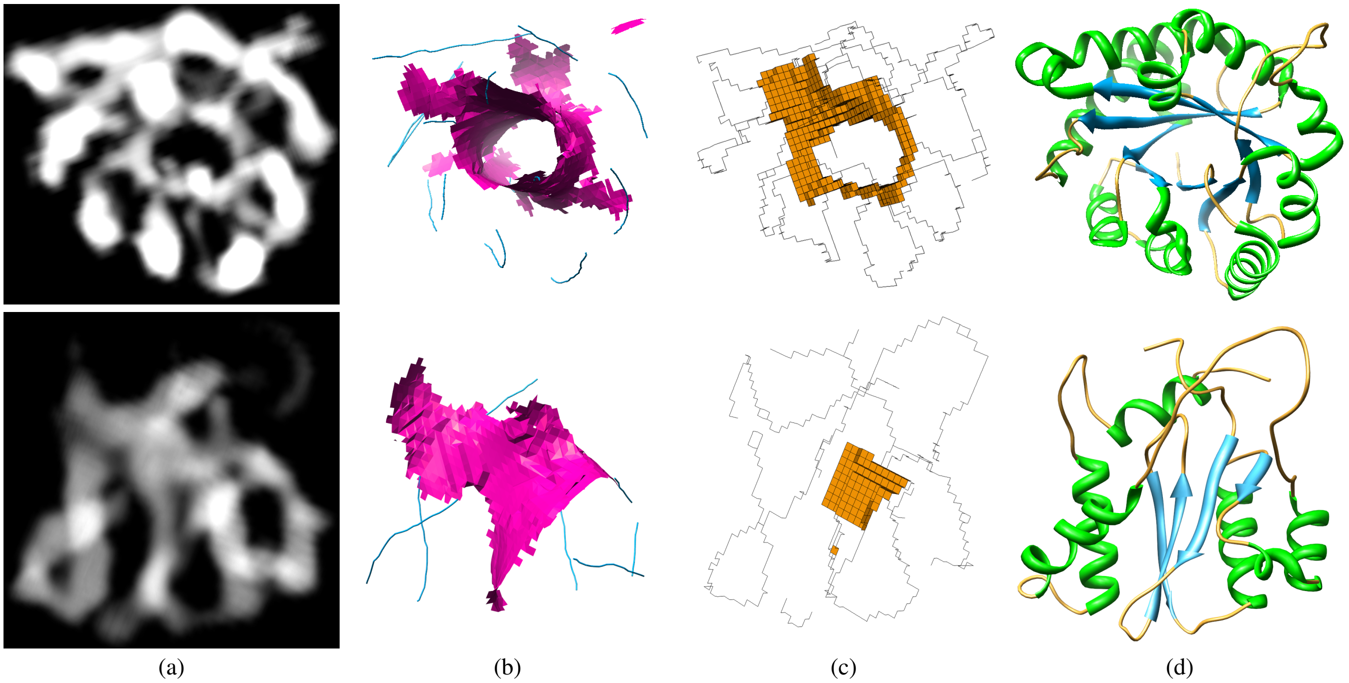

Figure 10: Cryo-EM volumes of proteins (1TIM and 2ITG) at 8Å resolution (a), skeletons computed by the method of [22] (b) and our method (c), and the ground-truth structure of these proteins (d).

7 Conclusion and Discussion

In this paper we proposed an innovative approach for skeletonization of density-like grayscale volumes, for the purpose of shape understanding. Our method does not require any explicit segmentation of the volume, is robust under the presence of noise, and is capable of extracting skeletal surfaces and curves corresponding to plate-like and rod-like grayscale shape components. We tested our technique on synthesized and medical data-sets to demonstrate its behavior in different application domains.

Limitations While emphasizing shape representation, the resulting skeleton of our method may not exhibit the desired topology of the imaged subject. For example, as can be observed in the top left curve of Figure 10 (c,d), the skeleton contains extraneous loops and broken curves. This topological noise is mainly due to the sensitivity of morphological thinning to image noise in the initial skeletonization stage. Unlike a solid model, the topology of the imaged subject in the grayscale volume is not well defined, and a correct topology often needs to be defined by a human expert. In the future we would like to investigate the incorporation of recently-developed topology-repair methods [10] that would guide the skeletonization process using user-specified topology.

The assumptions based on the observations of Section 1.1 limit the applicability of our technique to density-like grayscale volumes where features of interest are in high-density areas. Although typical in medical imaging, they may not apply to other data (e.g., a photograph of a scene). We would like to explore the extension of our algorithm to a more general set of data. One possible solution is to explore mapping functions that convert grayscale volumes in a different form to those satisfying our assumptions.

As described in the earlier section, the performance of the initial skeletonization step is dependant on the number of distinct gray-levels of the volume. While the performance can be improved by discretizing the range of gray-levels, performing iterative thinning for each single gray-level is still time-consuming. To make the process more efficient, a possible alternative that we will explore in the future is to perform only one iterative thinning step from low-density regions to high-density regions, while adjusting the shape and topology preservation criteria in binary thinning to the grayscale data.

8 Acknowledgements

We would like to thank Zeyun Yu for the implementation of the method in [22], Philips Research for the “Bones” and “Blood Vessels (CT)” data sets and Özlem Gürvit of the Institute for Neuroradiology, Frankfurt, for the “Blood Vessels (MRI)” data set. This work is supported in part by NSF grant (IIS-0705538).

References

[1] N. Ahuja and J. Chuang. Shape representation using a generalized potential field model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):169–176, 1997. 2

[2] G. Bertrand. A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters*, 16(9):979–986, 1995. 2

[3] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, pages 362–380, 1967. 1

[4] A. Bonnassie, F. Peyrin, and D. Attali. Shape description of three-dimensional images based on medial axis. In *ICIP*, volume 3, pages 931–934, 2001. 1, 2

[5] G. Borgefors. Distance transformations in digital images. *Comput. Vision Graph. Image Process.*, 34(3):344–371, 1986. 2

[6] T. K. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. In *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 356–366, New York, NY, USA, 2002. ACM Press. 2

[7] P. Dokladal, C. Lohou, L. Perroton, and G. Bertrand. A new thinning algorithm and its application to extraction of blood vessels. *Proc. of Biomedsim*, pages 32–37, 1999. 3

[8] K. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3–16, 1981. 2

[9] T. Ju, M. Baker, and W. Chiu. Computing a family of skeletons of volumetric models for shape description. *Computer-Aided Design*, 39(5):352–360, 2007. 1, 2, 3, 5

[10] T. Ju, Q.-Y. Zhou, and S.-M. Hu. Editing the topology of 3d models by sketching. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 42, New York, NY, USA, 2007. ACM. 7

[11] G. Kindlmann, X. Tricoche, and C.-F. Westin. Anisotropy creases delineate white matter structure in diffusion tensor MRI. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4190, pages 126–133, Copenhagen, Denmark, October 2006. 3

[12] A. López, D. Lloret, J. Serrat, and J. Villanueva. Multilocal creaseness based on the level-set extrinsic curvature. *Computer Vision and Image Understanding*, 77(2):111–144, 2000. 3

[13] A. López, F. Lumbreras, and J. Serrat. Evaluation of methods for ridge and valley detection. *Evaluation*, 21(4):327–335, 1999. 3

[14] S. Mersa and A. Darwish. A new parallel thinning algorithm for gray scale images. *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, pages 409–413, 1999. 2

[15] K. Palágyi and A. Kuba. A parallel 3d 12-subiteration thinning algorithm. *Graphical Models and Image Processing*, 61(4):199–221, 1999. 2

[16] P. Saha, B. Gomberg, and F. Wehrli. Three-dimensional digital topological characterization of cancellous bone architecture. *IJIST*, 11(1):81–90, 2000. 1, 2

[17] P. K. Sahoo, S. Soltani, A. K. Wong, and Y. C. Chen. A survey of thresholding techniques. *Comput. Vision Graph. Image Process.*, 41(2):233–260, 1988. 2

[18] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Shape description by medial surface construction. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):62–72, 1996. 2

[19] S. Svensson, I. Nystrom, C. Arcelli, and G. S. di Baja. Using grey-level and distance information for medial surface representation of volume images. *icpr*, 02:20324, 2002. 3

[20] S. Svensson, I. Nyström, and G. Sanniti di Baja. Curve skeletonization of surface-like objects in 3d images guided by voxel classification. *Pattern Recognition Letters*, 23(12):1419–1426, October 2002. 2

[21] J. Weszka. A survey of threshold selection techniques. *Pattern Recogn*, 7:259–265, 1978. 2

[22] Z. Yu and C. Bajaj. A structure tensor approach for 3d image skeletonization: Applications in protein secondary structure analysis. *Image Processing, 2006 IEEE International Conference on*, pages 2513–2516, 8-11 Oct. 2006. 3, 6, 7

[23] S. Zhang, Çagatay Demiralp, and D. H. Laidlaw. Visualizing diffusion tensor mr images using streamtubes and streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, 2003. 3

A Scoring functions in limit cases

The ellipsoidal representation of the structure tensor has a clear geometric shape in the limit cases when the eigenvalues approach zero, and can be used to explicitly derive the limit of scoring functions in these conditions.

When u_1 approaches zero so does u_2 and u_3 (as $u_1 \geq u_2 \geq u_3$). In this case we treat the ellipsoid as a sphere, where all embedded line segments have the same length, and all embedded cross sections have the same area. Therefore, the value of the scoring functions (the ratio between the minimal and actual length / area) will always evaluate to one.

$$\lim_{u_1 \to 0} \frac{L(v_3)}{L(c)} = 1, \tag{5}$$

$$\lim_{u_1 \to 0} \frac{A(v_2, v_3)}{A(n_1, n_2)} = 1. \tag{6}$$

In the case where u_2 approaches zero, so does u_3, reducing the ellipsoid to a needle with an infinitesimally small circular cross section (also can be interpreted as an infinitely long cylinder). Here, the minimal line (or surface) is the projection of the actual line (or surface) onto the plane defined by the surface normal v_1, reducing the scoring functions to the following vector dot products:

$$\lim_{u_2 \to 0} \frac{L(v_3)}{L(c)} = c \cdot \frac{(v_1 \times (c \times v_1))}{\|v_1 \times (c \times v_1)\|}, \tag{7}$$

$$\lim_{u_2 \to 0} \frac{A(v_2, v_3)}{A(n_1, n_2)} = (n_1 \times n_2) \cdot v_1. \tag{8}$$

When u_3 approaches zero, the ellipsoid reduces to a cylinder with an infinitesimally small height and an ellipse shaped cross section where u_1 and u_2 are the length of the axes. The minimal line is the projection of the actual line onto the v_3 vector, reducing the curve scoring function to the following vector dot product:

$$\lim_{u_3 \to 0} \frac{L(v_3)}{L(c)} = c \cdot v_3. \tag{9}$$

The surface score on the other hand, reduces to the product of two ratios; the first being the ratio between u_2 and the length of the curve within the ellipsoid in the l_1 direction, and the second being the ratio between l_2 and its projection onto the v_3 vector. l_1 and l_2 are orthogonal unit vectors which both lie on the actual surface. Additionally, l_1 also lies in the plane defined by the normal v_3.

$$\lim_{u_3 \to 0} \frac{A(v_2, v_3)}{A(n_1, n_2)} = \frac{\sqrt{u_1^2 \cos^2(\theta) + u_2^2 \sin^2(\theta)}}{u_1} \times (l_2 \cdot v_3), \tag{10}$$

where,

$$l_1 = \frac{v_3 \times n}{\|v_3 \times n\|}, \qquad l_2 = n \times l_1,$$
$$\theta = \arccos(v_2 \cdot l_1), \qquad n = (n_1 \times n_2).$$

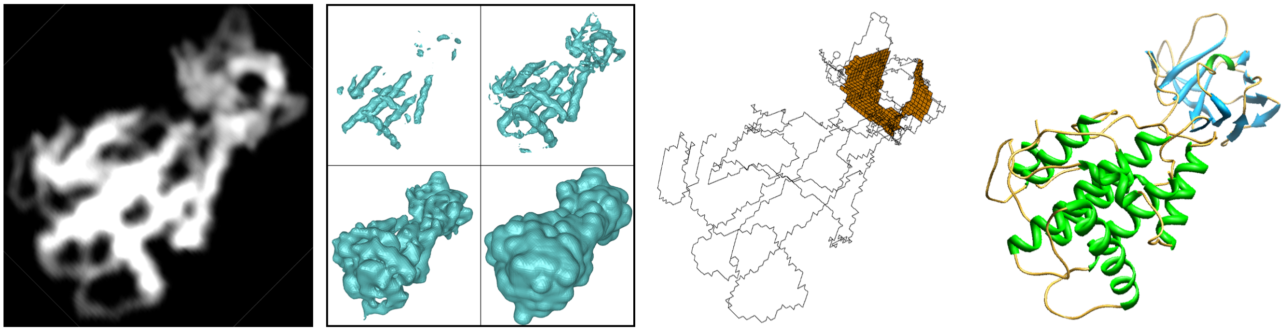

Figure 1: (a) (b) (c) (d)

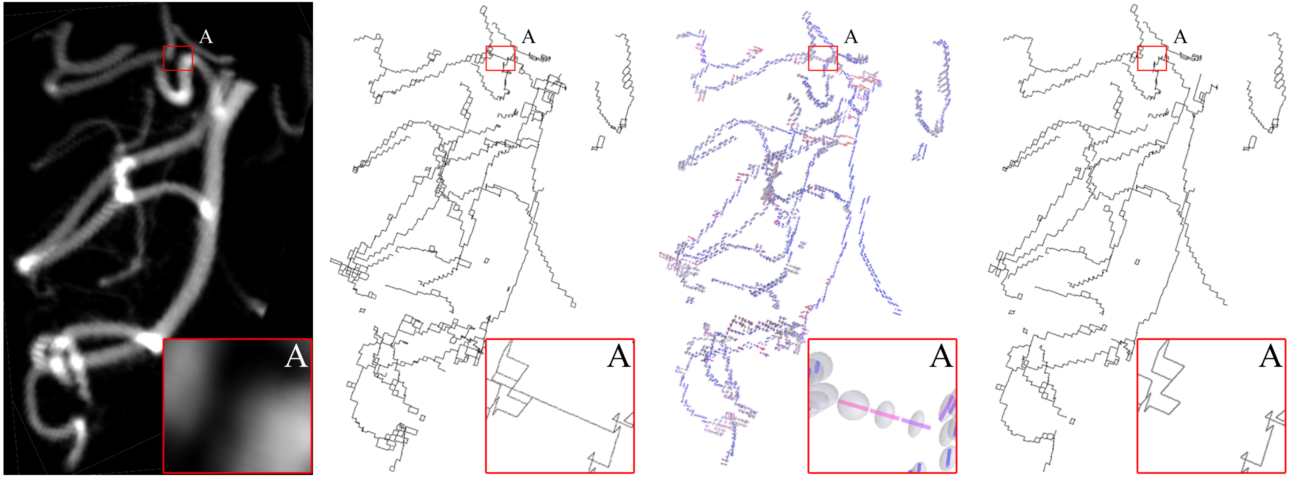

Figure 8: (a) (b) (c) (d)

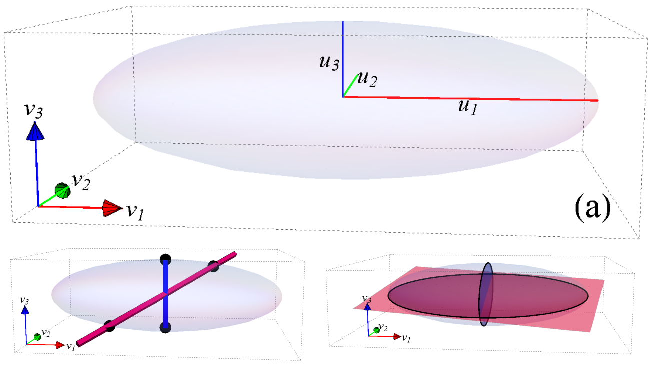

Figure 5: (b) (c)

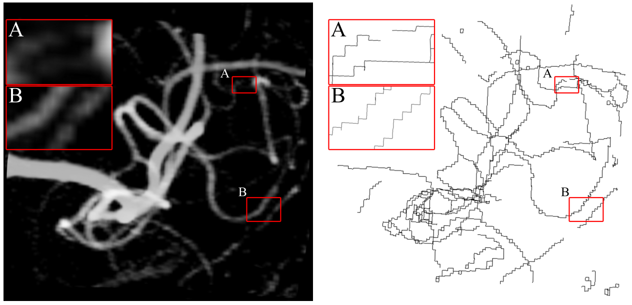

Figure 7: (a) (b)

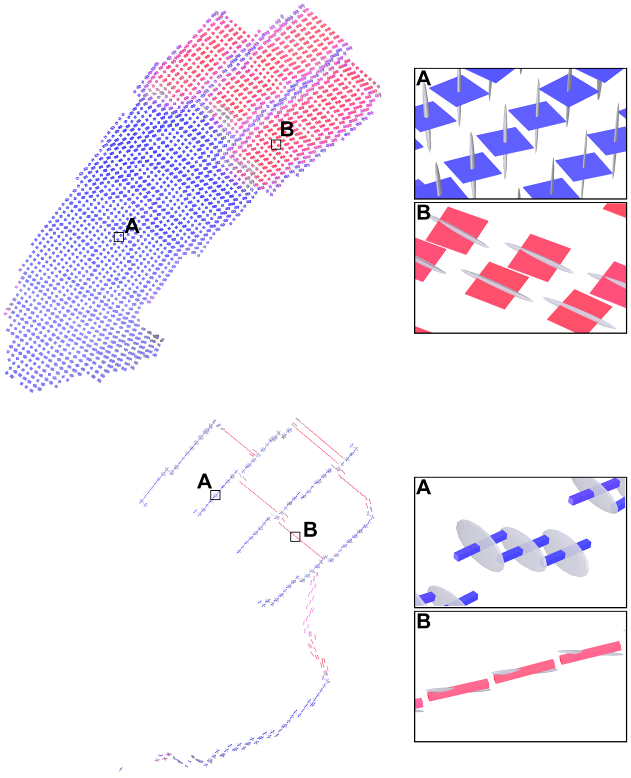

Figure 6: