Computer Methods in Biomechanics and Biomedical Engineering.
Vol. 5, No. 5, pp. 329-341. 2002.

Skeletonization of Volumetric Angiograms for Display

Dingrong Yi and Vincent Hayward

Department of Electrical and Computer Engineering &
Center For Intelligent Machines
McGill University
3480 University Street, Montreal, Quebec, H3A 2A7 Canada
{yidingr|hayward}@cim.mcgill.ca

Abstract

The display of three-dimensional angiograms can benefit from the knowledge of quantitative shape features such as tangent and curvature of the centerline of vessels. These can be obtained from a curve-like skeleton representation. If connectivity and topology are preserved, and if geometrical constraints such as smoothness and centeredness are satisfied, it is possible to estimate length, orientation, curvature, and torsion. It is also required that no part of the original object be left unrepresented. An efficient method for the identification of such shape components is developed. First, a suitable representation is obtained using a voxel coding approach to yield connected and labeled unit-thick paths. The desired features are estimated from a smoothed version of the skeleton produced by a moving average filter. The computational cost is linear, of the order of N_{object}, the total number of object voxels contained in the binary volumetric data. The method is also shown to be robust to boundary noise. Examples are discussed.

Keywords: angiography, shape extraction, display, skeletonization, voxel coding, interpolation.

1 Introduction

Techniques including digital subtracted angiography (DSA), computed topography angiography (CTA), phase contrast magnetic resonance angiography (MRIPC), and time-of-flight magnetic resonance angiography (MRITOF), provide practitioners with angiographic images of unprecedented resolution and signal to noise ratio. The transition from raw 3-D imaging data to a representation that provides a rich set of affordances for visualization is not trivial however. Examples of need for efficient visualization include the screening and diagnostic of cerebral vascular diseases [22], as well as planning incisions to a lesion in the least invasive manner possible [20]. It is an inevitable problem to have to display large size three-dimensional data sets on 2-D pixel screens. To date, two major graphical approaches have been developed: maximum intensity projection (MIP), and surface rendering.

1.1 Maximum Intensity Projection

Parallel rays are issued from each pixel of the projection screen, and the projected pixel value is the maximum density value encountered along the projection ray. Key advantages of this method include simplicity and speed due to the absence of pre-processing and post-processing. Nevertheless, the representation may not provide for an intuitive interpretation of the underlying structures contained in the data. Information related to depth is lost owing to the fact that it is not truly a 3-D visualization technique.

1.2 Surface Rendering

Surface rendering of angiograms affords true 3-D visualization. Objects are first segmented from the background to obtain binary volumetric data with label "0" being assigned to all background voxels and "1" to object voxels, then rendered as isosurfaces using such surface-fitting algorithms as dividing cubes or marching cubes [14, 24]. Small vessels or narrow structures cause the technique to break down. They may be skipped or else generate huge amounts of triangles, which is improper for fast-paced interactive visualization. A tentative explanation for the lack of acceptance of surface-rendering in 3-D is simply because it is too slow and details are lost in the absence of high performance computer graphics hardware.

1.3 Other Alternatives

Other alternatives include the use of stereopsis [7], as well as ideas borrowed from the field of augmented reality, whereby the primary data is overlaid with specific markers designed to alleviate the perceptual ambiguities contained in graphically displayed objects [9].

1.4 Motivation

We address the 3-D visualization problem by breaking it down into several tasks, each of them concerned with the visualization of different feature components of the shape to be displayed. The display itself, to be described in a forthcoming publication, is mediated by different modalities in the graphical and haptic domains, selected for their properties in terms of indicators of shape components. Initial results indicate, for example, that the local orientation of a vessel with respect to the viewing direction is efficiently conveyed in the haptic domain, while torsion or size are more effectively conveyed graphically. Three such components are contained in the vascular tube centerline geometry: tangent, curvature, torsion, as well as the locations of topological branching. Their extraction from raw data is the primary focus of this paper. While the described method has been developed with the visualization problem in mind, its properties should make it useful for other applications, chiefly among them is the automatic extraction of quantitative data from angiographic images. For the purpose of this paper, the results will be illustrated with simple visualization techniques such as pseudo-color mapping and stereo pairs.

Computer Methods in Biomechanics and Biomedical Engineering.
Vol. 5, No. 5, pp. 329-341. 2002.

2 Skeletons

A useful representation of a natural object can be its skeleton. The skeleton of a 3-D object may be defined as the locus of the centers of all maximal spheres inscribed in the object, thus touching its boundary at more than one point.

In the case of a vascular tree, the space curves traced by the skeleton represent the paths followed by the vessels, while a radius at each point represents its size. The properties of a skeleton needed to yield proper representation of a vascular system are:

- **Connectivity**: The vascular system is a connected object, hence its skeleton should be connected too. For any two points on the skeleton, there exists a path to connect them that is completely contained in the object.
- **Completeness**: The extracted skeletons must be complete in the sense that no part of the original raw data should be left unrepresented.
- **Centeredness**: The skeleton should be located at the center of the space occupied by a vessel.
- **Smoothness**: The vascular system is smooth. Smooth curves, or piecewise smooth curves are required to represent it.
- **Explicit topology labelling**: Branching sections of the vascular system shall be explicitly labelled at the corresponding location of its skeleton.
- **Insensitivity to noise**: The resulted skeleton should not be sensitive to boundary noise, since such noise is inevitable.
- **Size labeling**: The size of the cross-section of a vessel in a plane normal to its centerline should be known. The size should be rotation invariant.

Desirable attributes of the extraction procedure include the possibility to automate it as much as possible, as well as a low the computational complexity that should grow no faster than the size of the 3-D array. This is important given the large size of an angiographic data set.

The main contribution of this work is to provide an efficient and automatic procedure yielding skeletons having these properties. For a segmented angiographic 3-D array volumetric data, this method computes a skeleton representation of the underlying vessel systems that satisfies the above mentioned constraints.

3 Previous Approaches

Since the pioneering work of Blum [3], who realized that the shape of an object, especially that of a natural object, can be properly represented by its skeleton, six basic approaches have been proposed, namely: boundary-based approaches such as Voronoi diagrams, thinning algorithms, grassfire simulation, distance transform ridge detection, the voxel coding method, which is closely related to the present work, and recently, skeletonization approaches in scale space.

3.1 Boundary based approaches

An important approach is based on the Voronoi diagram (VD) [19]. In 2-D it is a collection of Voronoi polygons, each polygon being a non-overlapping sub-region of a 2-D planar shape bounded by straight line segments. The Voronoi diagram generated by a set of boundary points is such that the points at the intersection of two (or more) polygons are equally distant to two (or more) nearest boundary points. This property was used to generate skeletons from a set of boundary point [6, 18]. The Voronoi diagram of a discrete set of boundary points is called the discrete Voronoi medial axis (DVMA) and can be regarded as the intersection of the VD with the generating shape. Once the DVMA has been constructed, a skeleton can be extracted from it by pruning away undesirable Voronoi polygon edges [18]. In 3-D the Voronoi polygons extend to Voronoi polyhedra, and edges of Voronoi polygons extend to faces of polyhedra. While this approach is suitable to mesh many types of objects into finite elements, it does not apply well to the skeletonization of vessel data. Far too many polygons are needed to approximate narrow vessels.

3.2 Thinning algorithms

Most thinning algorithms rely on the topological properties of an object rather than on the metric properties of its shape. Thinning algorithms repetitively test and subsequently delete simple points. A simple point is a point whose removal does not change the topology of the underlying shape [2]. In practice, there may be many ways to characterize a simple point. Lee et al. listed 6 different such characterizations. Authors use various working definitions of simple points, resulting in different thinning algorithms and different skeletons [12]. There exist parallel thinning algorithms [1]. In a recent work by Nystrom a two phase thinning algorithm is used to skeletonize angiographic images [17], but the result is rotation dependent and sensitive to noise.

3.3 Grassfire simulation

In this approach, an initial firefront is ignited at the object boundary. It then propagates along the inward normal at constant speed. The merging points of different parts of the firefront are retained to become symmetric axis points, which are equidistant to at least 2 boundary points. A skeletonization technique by grassfire simulation was proposed by Leymarie and Levine [13]. An active contour (snake) represents the firefront. The movement of the snake is controlled by a pre-computed Euclidean distance-transform of the shape (or potential surface, which is the negative of the distance map for an object, and zero for the background). Objects with numerous branches such as cerebral vessels cause problems with this approach, as the initial firefront is difficult to locate and the inward normal is hard to compute.

3.4 Distance transform ridge detection

A distance transform is an operation that converts a binary object/background image into a distance map. For each point in the image, the map indicates its distance to the nearest boundary point. The gradient of the distance map is a vector field and may present loci of discontinuity, or ridges, which can be used to define a skeleton.

In practice, the gradient of the distance map may not be well defined and hence, it is not trivial to detect its discontinuities. Malandain and Fernandez-Vidal approximate this gradient at an interior point by a vector originating from this point and ending at the nearest boundary [15]. Recently, Bouix and Siddiqi proposed a divergence-ordered thinning algorithm that identifies points on ridges on the basis of a stronger negative divergence of the gradient vector field of the distance map at these points than at regular points [5]. They use local divergence as a measure of priority in a thinning process by removing points until all surviving points are not simple or until the associated divergence is below a chosen value. They could obtain a centeredness and topology preserving medial surface of a vessel contained in an angiogram. This promising approach is not immediately suitable for our purpose since it does not directly produce a unit voxel-thick and complete skeleton representation of the vessels.

3.5 Voxel coding

Zhou and Toga proposed a voxel coding approach to efficiently skeletonize volumetric objects [26]. Each object point has two codes. One is the boundary seeded code (BS) which coincides with the traditional distance transform to indicate the minimum distance to a boundary point. The second code is the so-called single seeded code (SS) which indicates the length of the shortest path to a common reference object point, subject to the constraint that this path be completely contained within the object. Hence the shortest path may be piecewise linear but not necessarily a straight line segment. The key idea of voxel coding is to use the SS codes to generate a connected raw skeleton and the BS codes to guarantee the centeredness of the final skeleton. An important feature for our purpose is that the obtained skeleton is guaranteed to be exactly one unit in thickness and this makes it suitable for further processing. This approach of skeletonization is adapted to our purpose.

3.6 Centerline extraction in scale space

The aforementioned five skeletonization approaches apply to segmented binary images. For tubular structures such as vessels, Krissian et al. apply multiscale analysis to raw intensity image to detect vessel structures of various sizes [11] . They propose a 'medialness' measure $m(X_A, \sigma_A)$ at a given point and scale to quantify how much a point X_A approaches the medial axis of an object. Based on the eigenvalues and eigenvectors of the Hessian matrix of the image, an adaptive medialness measure is proposed to be the mean of first order derivative information at the circle $C(\overrightarrow{X}_A, \theta\sqrt{t})$ centered at X_A with radius $\theta\sqrt{t}$. Zoom-invariance and response optimization determine the parameter to be $\sqrt{3}$, while t is the discrete scale associated to σ_A, the standard deviation of a Gaussian.

A set of medial responses taken at different discrete scales produces a multiscale response image. Normalization compensates for the decrease of intensity and of its derivatives with increasing scale. This yields a local maxima image from which the object centerline is obtained by segmentation and then thinning (Section 3.2). The resulting skeleton is composed of pieces of curves, each of them representing a piece of vessel. The size of the vessel at a point on the skeleton is proportional to the scale at which the current point has been extracted.

Computer Methods in Biomechanics and Biomedical Engineering.
Vol. 5, No. 5, pp. 329-341. 2002.

4 Method

Three items are introduced with respect to the original voxel coding method [26]. First, a dynamic array of voxels which collects all connected object voxels is explicitly extracted from the binary volumetric data. This array is named the object voxel chain. The SS codes of the object voxels are then obtained by simple propagation of the initially known SS code, from the head of the chain down to the tail in one operation, without resorting to a recursive mechanism. This is valuable since, for large data sets, the recursion can run very deep and cause thrashing in the management of heap memory. Second, we eliminate need of an expensive intra-path gap filling step. Third, the shortest path extraction (SPE) selects the next point as the smallest SS coded 26-neighbor, instead of the smallest SS coded 6 face neighbor. Hence, the extracted path is guaranteed to be the shortest path. In addition, no self knotted segment can occur within a path.

Quantitative shape feature components such as tangent, curvature, and torsion, are estimated in a post-processing step. The two step interpolation method that preserves loyalty to the original data at the voxel level and produces good estimates of tangent, curvature and torsion of the vessel centerlines. This is achieved by running a moving average on the discrete voxel skeleton, so that these quantities can be reliably computed by finite differences.

Figure 1 here

A binary volumetric 3-D array is used as an input. Label ‘1’ is assigned to all voxels that belong to a vessel, and ‘0’ is assigned to all background points. A software package called “Display+register”, freely available through anonymous `ftp` from the Montreal Neurological Institute (MNI), has been employed to segment vessels from background tissues.[1] An automatic flood-fill-in algorithm labels all the voxels connected to a selected seed point and whose intensity values fall between a prescribed range. Figure 1 summarizes the steps and the data flow.

4.1 Distance Transform I

The Euclidean distance must be approximated to account for the discrete nature of the data such that global Euclidean distance can be computed using only local operations on integers and storage requirement can be reduced from real numbers to integers. Several metrics, and correspondingly several methods, may be used to obtain a distance map. Ordering these metrics according to increasing computation complexity and memory requirements yields: the N-neighbors, the chamfer metric, the hyper-octagonal, and the D-Euclidean [4]. The chamfer metric (d_0, d_1, d_2) assigns distance d_0 from a voxel to its face neighbors, distance d_1 to its 12 edge neighbors, and d_2 to its 8 corner neighbors, see Figure 2. The optimum values for d_0, d_1 and d_2 computed by minimizing the maximum difference with the Euclidean distance are: $d_0 = 1$, $d_1 = 1.314$, and $d_2 = 1.628$ [4]. We use the chamfer metric (3, 4, 5) which uses 1.667 to approximate the optimal value $d_2 = 1.628$ and the optimal value of $d_1 = 1.314$ by 1.33 (see Figure 3). Another commonly used integer chamfer metric is (1, 2, 3), but it is less accurate.

First initialize the map so that each object voxel has an infinite distance value and so that each background voxel has a zero distance value. Zero padding is used to obtain the minimal

[1] `ftp.bic.mni.mcgill.ca`

Figure 2 here

distance to edge voxels. A forward mask is then swept over the volume, starting at the upper left corner of the first front slice, from left to right, from top to bottom, and from front to back (Figure 3). At each location, the sum of the local distance of each voxel of the mask and the value of the voxel it covers is computed. The smallest of these sums is assigned as the new value to the current voxel—the voxel covered by the (i, j) voxel of the mask. This forward pass calculates the distance between each object voxel and its nearest boundary point at its left, top and front sides. The minimum distance is then adjusted by considering the boundary at its right, bottom and back sides. This involves moving a backward mask (Figure 3) starting from the bottom-right voxel of the last back slice in an opposite way as for the forward mask [4].

Figure 3 here

4.2 Distance Transform II

The single-seeded code SS field corresponds to a second distance map which for each object voxel indicates the minimum-in-object distance to a common reference point RP. For voxels in the background, the distances to RP are all infinite. In principle, RP can be any object voxel. However, RP preferably selected to be a prominent point on the object, for example a point with the largest BS code.

The SS codes are computed layer by layer, in a forward propagation process. To make this efficient, a dynamic object-voxel-chain that contains all object voxels connected to RP is first extracted. Point RP is first extracted and set it to be a background voxel. Then, RP's object face neighbors are joined to the first surrounding layer and, similarly, those neighbours become background voxels. An object face neighbor of a voxel **V** is not only **V**'s face neighbor, but also an object voxel. The number of object face neighbours of voxel **V** may be smaller than 6. The first surrounding layer of RP generates a second surrounding layer and so-on, until the farthest element to RP is added. The surrounding layer $(l + 1)$ contains all the object face neighbours of each member of the surrounding layer l. All object voxels are necessarily ordered into a object-voxel-chain according to their distance to RP. If there is more than one disconnected object in the scene, then each should be associated to a different RP.

The SS codes of all voxels are initialized to infinity except that of RP, which has zero distance to itself. The SS codes for each member of the first surrounding layer of RP are calculated as the smallest of the SS codes of its object face neighbors plus one. Since all voxels have infinite SS codes except RP, all members of the first layer surrounding RP get one as their SS codes based on RP directly. Similarly a layer indexed by l get l as SS code. For each member of the object-voxel-chain, the SS coding scheme can be summarized as:

$$\mathrm{SS}_{[i,j,k]} = \min \sum_{i',j',k'} \mathrm{SS}_{[i',j',k']} + 1, \qquad (1)$$
$$|i - i'| + \quad |j - j'| + \quad |k - k'| = 1$$

where (i, j, k) are the indices of a voxel **V** in the object-voxel-chain, and (i', j', k') are those of **V**'s face neighbours. The SS coding scheme given by Equation (1) is equivalent to the chamfer metric (1, 2, 3). The only initially known SS code at RP is propagated efficiently from the head of the object-voxel-chain (RP) down to the tail of the chain. The object-voxel-chain has non-decreasing SS codes.

The SS codes obtained so far convert the object with voxel as its basic element into an object with clusters of varying sizes as its basic elements. A cluster is a group of voxels that are connected and have the same SS code [26]. The cross section of the object is approximated by a cluster. The center of a cluster is a voxel that has the largest BS. Strictly speaking, the center of a cluster is not the gravity center of the cluster, but a good approximation of it. A cluster with a SS code that is strictly larger than that of all its neighboring clusters is called a local maximum (LM) cluster. In general LM clusters are found at the tips of branches or at the joining section of two branches that form a loop. A merging cluster is a cluster that has two or more leading clusters with the same SS code as its smallest SS coded 26-neighbours. A cluster is said to be branching when its SS code is smaller than that of at least two of its neighbours. These concepts are illustrated in Figure 4.

Figure 4 here

4.3 Skeleton Extraction

The object skeleton evolves from a raw skeleton, to a network of medial paths termed a Z-skeleton, to a labeled Z-skeleton, and finally to a labeled Q-skeleton which is no longer aligned on a grid. The procedures used for these transformations are: shortest path extraction, medial point replacement, medial path gap filling, a procedure to add topological information to the Z-skeleton, and finally a moving average to yield a Q-skeleton as described in the next subsections.

4.3.1 Raw skeleton

The LM clusters are of particular importance since they indicate the local group of voxels farthest to RP. These clusters are first extracted and their centers are ordered into a LM chain according to their decreasing SS codes. Each voxel in the LM chain defines a shortest in-object path to RP. Each of such shortest paths are skeleton branches. The number of skeleton branches is initially the size of the LM chain and then found by a shortest pass extraction procedure (SPE). Each skeleton branch initially has a head, a single member of the LM chain. Then the next point is the point of smallest SS code among the 26-neighbors. This process iterates until RP is reached, it must then have the smallest SS code. A path expanded in this fashion will be guaranteed to be the shortest in-object path between a member of LM chain and RP. No self-knotted segment can occur within a branch. A shortest in-object path exists for each element in the LM chain and is routed accordingly. Branches may terminate at RP. To avoid redundant calculations, any path terminates at a voxel that has been visited by another path. Hence, a terminal voxel is always shared by two skeleton branches and the cluster to which it belongs must be a branching cluster.

For complicated objects having loops, such as some vessel systems found in the brain, the concept of merging cluster and a proper way to detect them are essential. For each voxel $\mathbf{V_j}$ listed in a skeleton branch found by SPE, the cluster $\mathbf{C}$ it belongs to is first found and then the number of successive clusters $\mathbf{S_1}, \mathbf{S_2}, \cdots \mathbf{S_k}$ adjacent to $\mathbf{C}$ with the same SS code as that of its next point $\mathbf{V_{j+1}}$ on the skeleton branch are checked. If there is more than one successive cluster, then the center of those clusters that does not contain $\mathbf{V_{j+1}}$ are appended to the LM chain. Like other members of the LM chain, each of these newly appended voxels also corresponds to a skeleton branch and are routed by the SPE procedure. To guarantee intra-path connectivity, the center of a merging cluster, namely voxel $\mathbf{V_j}$, is inserted in that skeleton branch. Here, the center of a merging cluster must be shared by two (or more) skeleton branches.

Skeleton branches starting from LM clusters and those starting from merging clusters constitute the raw skeleton of the vessel system. They are complete as since any object must be decomposed into separate parts. One first category corresponds to independent branches, characterized by LM clusters. The second category corresponds to loops (joined branches), and is characterized by merging clusters. This decomposition is complete and hence the raw skeleton extracted represents all the original parts of the underling object. The raw skeleton must be connected as well due to the sharing of the centers of branching clusters and merging clusters. As far as inter-branch connectivity is concerned, it is also guaranteed since every next point added to a branch must a 26-neighbor of its predecessor.

4.3.2 Network of medial paths

One remaining issue is centeredness which may not satisfied yet. It is simple to adjust the skeleton to the physical center of the underlying object by replacing each voxel of the raw skeleton—except its first and its last—by the center of its corresponding cluster. When the center contains more than one voxels, then the closest to its predecessor is used based on its BS code field. This procedure is termed medial point replacement (MPR). The MPR procedure transforms the raw skeleton into a network of medial paths.

4.3.3 Z-skeleton

Should connectivity be lost during medial point replacement, gaps are easily detected and filled. For a voxel $\mathbf{V_j}$ in a medial path, if its next voxel $\mathbf{V_{j+1}}$ is not a 26-neighbor, then a gap exists. In general, the gap is just one voxel. For such cases, the missing voxel can be found by interpolation between both ends of the gap. If the gap is larger than one voxel, then the SPE procedure is employed to fill it. The object-voxel-chain starts from one end of the gap and terminates at the other end of the gap to avoid collecting all the object voxels. The reference point that used to generate the SS code field is simply $\mathbf{V_j}$.

The skeleton so far obtained is exactly one voxel thick, connected, located at the center of the underlying object, and each branch has genus zero. The data structure for each branch includes a branch identification number, its length, and labels to describe its head and its tail. A dynamic array collects the voxels it contains, and a second array indicates the corresponding BS codes of the voxels. A third array encodes the corresponding topological labels.

Hence, the skeleton satisfies the constraints listed in Section 2 and is called a Z-skeleton since it is a set of voxels such that:

$$\begin{aligned}\mathbf{V}: \quad & \mathbf{V}(u_{p,l}) = 1,\ u_{p,l} \in \mathbf{Z}^3, \\ & \forall p \in \{1, \ldots, P\},\ \forall l \in \{\{1, \ldots, L(1)\}, \ldots, \{1, \ldots, L(P)\}\} \qquad (2)\end{aligned}$$

where the $u_{p,l}$ are the spatial grid indices of the l^{th} voxel of the p^{th} skeleton path, with a total of P paths each having $L(p)$ voxels. The algorithm used to obtain the labeled Z-skeleton is outlined in Figure 5.

Figure 5 here

In summary, the Z-skeleton encodes the original data in the form of a list of spatially indexed voxels. In addition, each voxel carries topological properties as well as the size of the vessel at each discrete location.

5 Tangent, curvature and torsion

The coarse discretization of volumetric angiograms precludes the direct use of differentiation in order to estimate shape feature components such as tangent, curvature, and let alone torsion, as in [8, 10], even using adaptive windowing approaches similar to those of [25], or post-filtering techniques as proposed in [16].

A better approach is to minimize the effects of discretization by computing a skeleton which no longer is required to be aligned on a grid: a smoothed skeleton. One alternative is to compute a spline passing through all the points of the smoothed skeleton and differentiate it three times to obtain tangent, curvature, and torsion everywhere. Such procedure is not advisable since the spline would have to be a quartic at least. This would be computationally cumbersome and the interpolant could oscillate. A more effective approach is to compute a cubic spline for position and obtain the components by finite differences, which can then be interpolated linearly.

5.1 Q-skeleton

Smoothing is achieved by replacing each voxel $u_{p,l}$ by a point $v_{p,l}$ produced by a weighted average with its neighbors. An essential necessity is to guarantee that each new point $v_{p,l}$ be located inside the original voxel. Thus the original geometrical information contained in the segmented data is preserved.

$$\begin{array}{ll} v_{p,l}: & |v_{p,l} - u_{p,l}|_\infty < 0.5,\ v_{p,l} \in \mathbf{Q}^3, \\ & \forall p \in \{1, \ldots, P\},\ \forall l \in \{\{1, \ldots, L(1)\}, \ldots, \{1, \ldots, L(P)\}\} \end{array} \tag{3}$$

While many filters are possible, a simple moving average filter with the mask $(1, 2, 1)$ has been found to be quite effective. The index p is omitted since the procedure is repeated for each path.

$$\begin{aligned} v_1 &= u_1 \\ v_2 &= \frac{1}{4}(v_1 + 2u_2 + u_3) \\ &\cdots \\ v_l &= \frac{1}{4}(v_{l-1} + 2u_l + u_{l+1}), \quad l = 2, \ldots, L(p) - 1. \end{aligned} \tag{4}$$

The maximal shift may be conveniently evaluated using the infinite norm and the triangular inequality. This mask is nearly optimal in the sense that the maximal amount of shift along each of the x, y, and z directions is bounded by 0.5, thus preventing the resulting smoothed skeleton from escaping voxel boundaries in any of the three directions. All calculations done:

$$\begin{aligned}
\|v_l - u_l\|_\infty &\le \frac{1}{4^{l-1}}\|u_1 - u_l\|_\infty + \frac{2}{4^{l-1}}\|u_2 - u_l\|_\infty + \frac{9}{4^{l-1}}\|u_3 - u_l\|_\infty + \frac{9}{4^{l-2}}\|u_4 - u_l\|_\infty \\
&\quad +... + \frac{9}{4^3}\|u_{l-1} - u_l\|_\infty + \frac{9}{4^2}\|u_l - u_l\|_\infty + \frac{1}{4}\|u_{l+1} - u_l\|_\infty \\
&\le \frac{1}{4} + \frac{9}{4^2}\sum_{k=1}^{l-1}\frac{k}{4^k} - \frac{l-2}{4^l} - \frac{l-1}{4^{l+1}} \\
&\le 0.5 \qquad \forall l > 1.
\end{aligned} \tag{5}$$

More careful mask selections will have rather marginal effects on the result. For example, a local averaging filter with mask $(1, 2, 1)$ would produce a maximum shift of just 0.5. A Q-skeleton has the same number of elements as the Z-skeleton, but they are no longer aligned on a regular grid. Please see Figure 6 for an illustration of the effect of smoothing on a two dimensional example.

Figure 6 here

5.2 Finite Differences

An estimate of the tangent at l can be taken to be the average of the two half tangents estimates:

$$\hat{T}_l = \frac{1}{2}(\hat{T}_l^+ + \hat{T}_l^-), \quad \text{with} \quad \hat{T}_l^+ = \frac{v_{l+1} - v_l}{|v_{l+1} - v_l|}, \quad \text{and} \quad \hat{T}_l^- = \frac{v_l - v_{l-1}}{|v_l - v_{l-1}|} \tag{6}$$

The angle between two successive tangents can be found using a dot product and a first order approximation of the arc-length from v_{l-1} to v_{l+1}, the curvature estimate per unit of length is:

$$\hat{C}_l = \frac{\cos^{-1}(\hat{T}_l^+ \cdot \hat{T}_l^-)}{|v_{l-1} - v_l| + |v_l - v_{l+1}|} \tag{7}$$

While curvature reflects bending in the osculating plane, torsion reflects bending out of it. Torsion measures the rate of change of direction of the osculating plane. The direction normal to the osculating plane can be found as a cross product:

$$\hat{N}_l = \begin{cases} \frac{\hat{T}_l^- \times \hat{T}_l^+}{|\hat{T}_l^- \times \hat{T}_l^+|} & \text{if } |\hat{T}_l^- \times \hat{T}_l^+| > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Then, the torsion estimate per unit of length can be found from the angle between two consecutive osculating planes, from l to $l+1$ is:

$$\hat{S}_{l\to(l+1)} = \frac{\cos^{-1}(\hat{N}_l \cdot \hat{N}_{l+1})}{|v_l - v_{l+1}|} \tag{9}$$

In practice, for display, piece-wise cubic interpolation (cardinal splines) is only needed for position. Linear interpolation for the rest is adequate.

6 Results

6.1 Actual Example

The method is applied to a real 3-D computed rotational angiography (CRA) image. The size of the angiography is 360 × 330 × 420, with a grid size of 0.54 mm. The binary segmented image is 356 × 148 × 216 with 44,936 vessel voxels. There are two embedded separate vessel systems, the major has 44,758 voxels and the smaller has 178 voxels. Figure 7 is the MIP display of the image, in which small vessel branches are embedded by the background and invisible to the viewer.

Figure 7 here

The smoothed-skeleton superimposed on the surface display of the segmented binary image as well as the pseudo-colored display of orientation, absolute curvature and torsion can be viewed in Figure 8.

Figure 8 here

In Figure 9, four more views of the smoothed-skeleton are shown but this time in the form of stereo pairs obtained with viewing angles that differ by 7^0. Figure 10 shows an enlarged small section of a skeleton snaking through ten voxels.

Figure 9 here

Figure 10 here

6.2 Synthetic Example

The same method is now applied to synthetic objects. A 128 × 128 × 128 binary volumetric image contains two such objects, synthesized by sweeping a sphere along known centerlines. One object is made of straight line segments meeting at 90^o and 45^o, and of an arc of a circle. The other is an helix of constant torsion. This results in 8652 object voxels displayed in Figure 11 as a surface rendered 3D image. Cases such as sharp angle branching, right angle branching, a loop, and independent objects are contained in this example.

Figure 12 shows the corresponding smoothed-skeleton superimposed on the surface display of the reconstructed object along with the same pseudo-colored display of estimated orientation, absolute curvature, and torsion. The reconstruction results from the union of spheres centered at all the voxels of the Z-skeleton and of radii estimated by the algorithm. The objects are reasonably well reconstructed including the branches and the loop, but with imperfections at some clusters when they cannot approximate the cross section of the original object. Also notice that the skeleton extends to the very end of the original object where the reconstructed sphere radii vanish, except for the reference point situated at the bottom the left object.

Figure 11 here

Figure 12 here

6.3 Sensitivity to boundary noise

The sensitivity to boundary noise is low owing to the fact that the basic component of an object is a cluster rather than a single voxel. This has an integration effect which reduces the sensitivity to small perturbations. The length of each skeleton path is readily available, so the resulted skeleton can be pruned from unwanted minor branches without any effect on the other portions of the skeleton.

To test the noise sensitivity a noised image obtained by randomly adding noise to the CRA binary image of Figure 8. This noised image has 113,392 additional object voxels, see Figure 13. Visual inspection indicates that the sensitivity to the boundary noise is indeed low.

Figure 13 here

6.4 Computational complexity

N_{total} is the total number of data, the sum of the number of object voxels N_{object} and of the number of background voxels $N_{\text{background}}$.

The Distance Transform I scans the 3-D binary array twice to assign each object voxel a minimum distance to its nearest boundary points, yielding $2N_{\text{total}}$ integer operations.

The Distance Transform II involves two steps to obtain the SS code field. The connectivity chain involves N_{object} bookkeeping operations. Propagating the SS code for RP from the head of the connectivity chain down to its tail, also requires N_{object} integer operations. Here each operation involves only integer summation and comparison. The overall cost is linear, on the order of N_{object}.

The skeleton extraction procedure has 3 steps: SPE MCD and CPR. Each of these steps consumes N_{object} operations, one operation per voxel, integer comparison and dynamic appending, checking, and integer operations, respectively. The overall complexity of skeleton extraction is linear, on the order of N_{object}.

The estimations of tangent, curvature and torsion are all linear in the total length of the skeleton paths. The cubic-spline interpolation solves $(N-2)$ linear equations with a tridiagonal coefficient matrix, where N is the number of voxels that a path contains. Hence the cost for parameterization is linear, of the order of N_{object} [21].

The overall operation linearly depends on the number of object voxels, N_{object}, except for the Distance Transform I which depends on N_{total}. In general for angiographic images, $N_{\text{object}} \ll N_{\text{total}}$. For example, for the image shown in Figure 8, $N_{\text{object}} = 0.39\% N_{\text{total}}$.

7 Conclusion and Future work

This paper has described an efficient method to extract one voxel-thick skeletons from angiographic data which have a number of desired properties for quantitative features analysis and display. The skeletons are complete, connected, labeled, and have low sensitivity to noisy data. A method was introduced to reduce the effects of coarse discretization. Good estimates of tangent, curvature and torsion are easily calculated from finite differences.

We are presently developing a multimodal visualization station using graphics and haptics. For example, a well-designed haptic force feedback may be used to convey to a user the sensation of depth in the projecting direction, even with planar devices [23]. As initially discussed, such depth display is cumbersome to implement graphically. Similarly, curvature and relative distance between two overlapped objects can also be displayed effectively.

Acknowledgements

The angiographic data set used was kindly provided by Dr. Hua Qian from the Robert Research Institute (RRI) in London, Ontario, Canada.

This research is funded by project "Intelligent Tools for Diagnosis, Surgery and Measurement of Resulting Patient Outcomes" supported by IRIS-III, the Institute for Robotics and Intelligent Systems which is part of Canada's Network of Centers of Excellence program (NCE). Additional funding is provided in the form of a McGill Wong Fellowship to the first author, and by NSERC, the Natural Sciences and Engineering Council of Canada, in the form of an operating grant for the second author.

References

[1] G. Bertrand (1995), A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters*, 16:97–986.

[2] G. Bertrand and G. Malandain (1994), A new characterization of three-dimensional simple points. *Pattern Recognition Letters*, 15:169–175.

[3] H. Blum (1973), Biological shape and visual science (Part I). *Journal of Theoretical Biology*, 28:205–287.

[4] G. Borgefors (1984), Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345.

[5] S. Bouix and K. Siddiqi (2000), Divergence-based medial surfaces. *European Conference on Computer Vision (Dublin, Ireland)*, June, 2000.

[6] J. W. Brandt and V. R. Algazi (1992), Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3): 329–337.

[7] P. Charland and T. Peters (1996), Optimal display conditions for quantitative analysis of stereoscopic cerebral angiograms. *IEEE Transactions on Medical Imaging*, 15(5):648–656.

[8] I. Debled-Rennesson, J.-P. Reveilles (1996), A linear algorithm for segmentation of digital curves. In *Parallel Image Analysis: Theory and Applications*, Davis, L. S., *et al.*, Singapour: World Scientific, pp. 73.

[9] D. Drascic and P. Milgram (1996), Perceptual issues in augmented reality. In *Stereoscopic displays and virtual reality systems III*, Bolas, Fisher, Merritt, (Eds.). SPIE Volume 2653. pp. 123–134.

[10] F. Feschet and L. Tougne (1999), Optimal time computation of the tangent of a discrete curve: application to the curvature. In *8th International Conferences on Discrete Geometry for Computer Imagery, , CGCI'99, Lecture Notes in Computer Science, No. 1568*, Bertrand, G., Couprie, M., and Perroton, L. (Eds.), Springer-Verlag. pp. 31–40.

[11] K. Krissian, G. Malandain, N. Ayache, R. Vaillant and Y. Trousset (1999), Model based detection of tubular structures in 3-D images. Research Report No. 3736, Institut National de Recherche en Informatique et en Automatique.

[12] T. Lee, R. Kashyap and C. Chu (1994), Building skeleton models via 3-D medial surface/axis thinning algorithms. *Computer Vision Graphics Image Processing (CVGIP): Graphical Models and Image Processing*, 56(6):462–478.

[13] F. Leymarie and M. D. Levine (1992), Simulating the grassfire transform using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):56–75.

[14] W. E. Lorensen and H. E. Cline (1987), Marching Cubes: A high resolution 3-D surface construction algorithm. *Computer Graphics.* 21(3):163–169.

[15] G. Malandain and S. Fernandez-Vidal (1998), Euclidean skeletons. *Image and Vision Computing*, 16:317–327.

[16] J. Matas, Z. Shao, Z. and J. Kittler (1995), Estimation of curvature and tangent direction by median filtered differencing. In *8th International Conference, ICIAP'95, Image Analysis and Processing, Lecture Notes in Computer Science*, No. 974, pp. 83–88. Braccii, C., DeFloriani, L. and Vernazza, G. (Eds.), Springer-Verlag.

[17] I. Nystrom (1998), Skeletonization applied to magnetic resonance angiography images. *Medical Imaging 1998: Image Processing, Proc. SPIE 3338*, pp. 693–701.

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

[19] A. Okabe, B. Boots and K. Sugihara (1992), Spatial Tesselations: Concepts and Applications of Voronoi Diagrams, John Wiley and Sons.

[20] T. M. Peters and B. G. Pike (1999), Magnetic resonance angiography in image-Guided neurosurgery. In *Advanced Neurosurgical Navigation*, Alexander III E., and Maciunas, R. J. (Eds.), pp. 61–69. Thieme Medical Publishers, Inc., New York.

[21] W. H., Press, S. A. Teukolsky, W. T. Veterling, and B. P. Flannery (1992), *Numerical recipes in Fortran, The art of scientific computing, 2nd Edition*, Cambridge university press, London.

Computer Methods in Biomechanics and Biomedical Engineering.
Vol. 5, No. 5, pp. 329-341. 2002.

[22] A. Puig, D. Tost and I. Navazo (1997), An interactive cerebral blood vessel exploration system. In *1997 IEEE Visualization Conference*, pp. 443–446.

[23] G. Robles-De-La-Torre and V. Hayward (2000), Illusory surfaces and haptic shape perception. Proc. *2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems,* Proc. ASME Vol. DSC-69-2, pp. 1081–1087.

[24] W. Schroeder, K. Martin and B. Lorensen (1997), The Visualization Toolkit Prentice Hall PTR, second edition.

[25] F. Janabi-Sharifi, V. Hayward and C.-S. J. Chen (2000), Discrete-time adaptive windowing for velocity estimation. *IEEE T. on Control Systems Technology.* Vol. 8, No. 6, pp. 1003–1009.

[26] Y. Zhou and W. Toga (1999), Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):196–209.

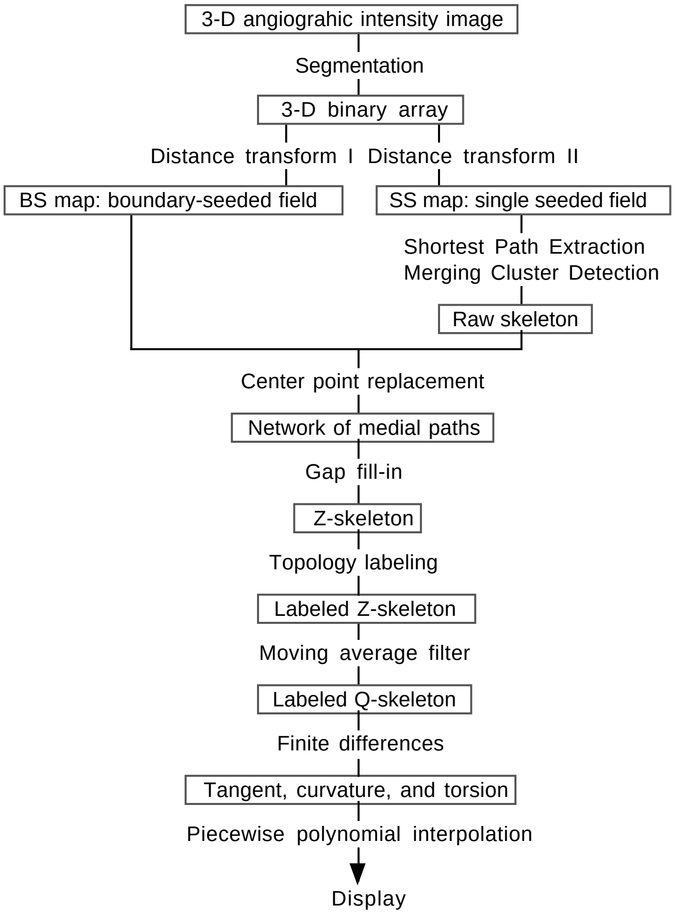


Figure 1: Data processing diagram.

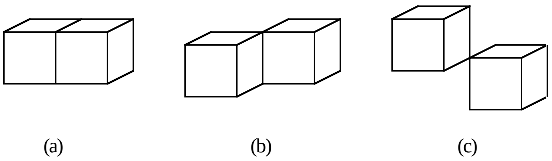
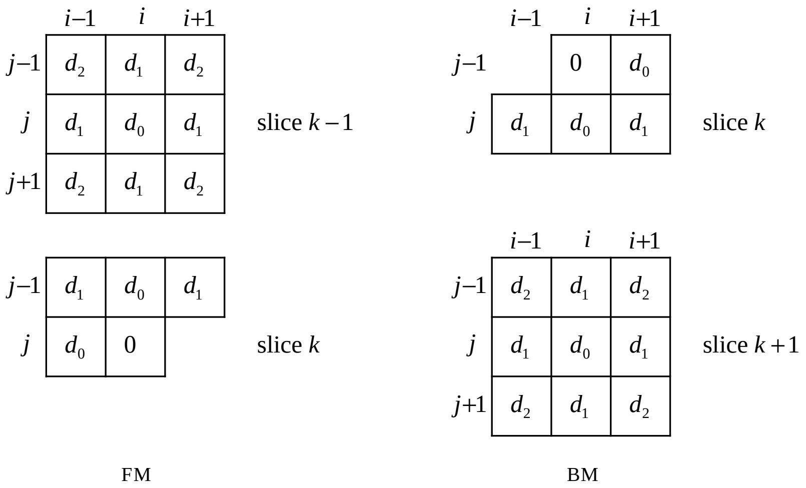


Figure 2: (**a**) 6 Face neighbors. (**b**) 12 edge neighbors. (**c**) 8 corner neighbors.



FM, slice $k-1$

	$i-1$	i	$i+1$
$j-1$	d_2	d_1	d_2
j	d_1	d_0	d_1
$j+1$	d_2	d_1	d_2

FM, slice k

	$i-1$	i	$i+1$
$j-1$	d_1	d_0	d_1
j	d_0	0	

BM, slice k

	$i-1$	i	$i+1$
$j-1$		0	d_0
j	d_1	d_0	d_1

BM, slice $k+1$

	$i-1$	i	$i+1$
$j-1$	d_2	d_1	d_2
j	d_1	d_0	d_1
$j+1$	d_2	d_1	d_2

Figure 3: Forward mask: FM. Here $d = (d_0, d_1, d_2) = (3, 4, 5)$. For each mask entry, FM$_{[i',j',k']}$, the value d_i indicates the local distance to the 0 voxel, i.e., FM$_{[i,j,k]}$. Backward mask: BM.

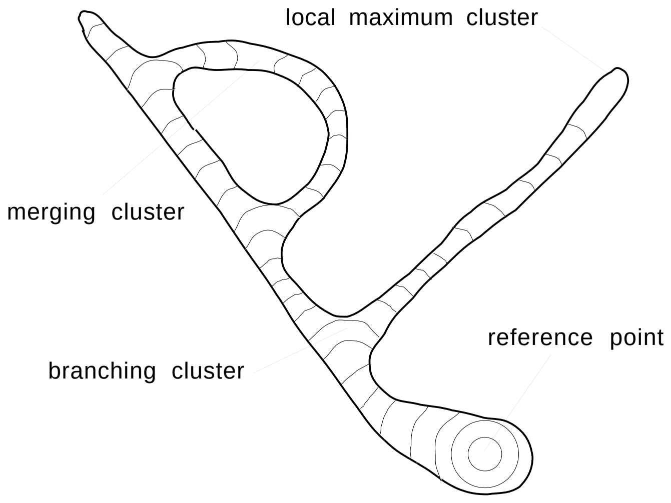


Figure 4: A 2-D illustration of the concept of cluster.

```
// The segmented data set is assumed to be known.
// ‘LMchain’ is a set of voxels.
// ‘Path’ is a set of sets of voxels. It also has an identifier Path[].id
proc Z-Skeleton
    LMchain := GetLocalMaximumChain(SS field, RP)
            // LMchain is in decreasing SS code order
    $n$ := Cardinal(LMchain)
    for $i$ := 1 to $n$ do
        ShortestPathExtraction(Path[$i$])                    // expand Path[$i$]
        for $j$ := 1 to Cardinal(Path[$i$]) do
            c := GetCluster(Path[$i$].Voxel[$j$])
            s[1...$K$] = Successors(c)
            for $k$ := 1 to $K$ do
                if s[$k$] $\not\ni$ Path[$i$].Voxel[$j+1$] then
                    LMchain := append(LMchain, Center(s[$k$]))
                    Path[$n$].id := i
                    $n$ := $n + 1$
                fi
            od
        od
        MedialPointReplacement(Path[$i$])
        MedialPathGapFill(Path[$i$])
        SetTopologicalLabels(Path[$i$])
    od
```

Figure 5: Algorithm outline.

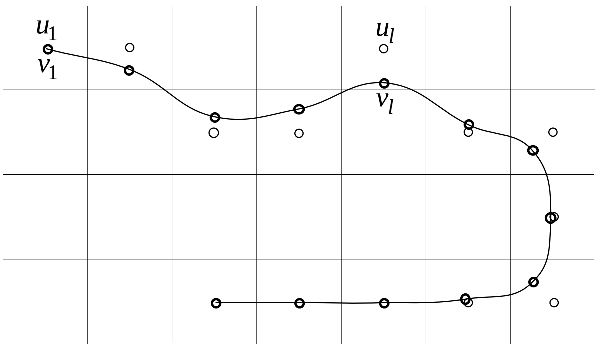


Figure 6: Illustration of the effect of the smoothing filter in two dimensions.

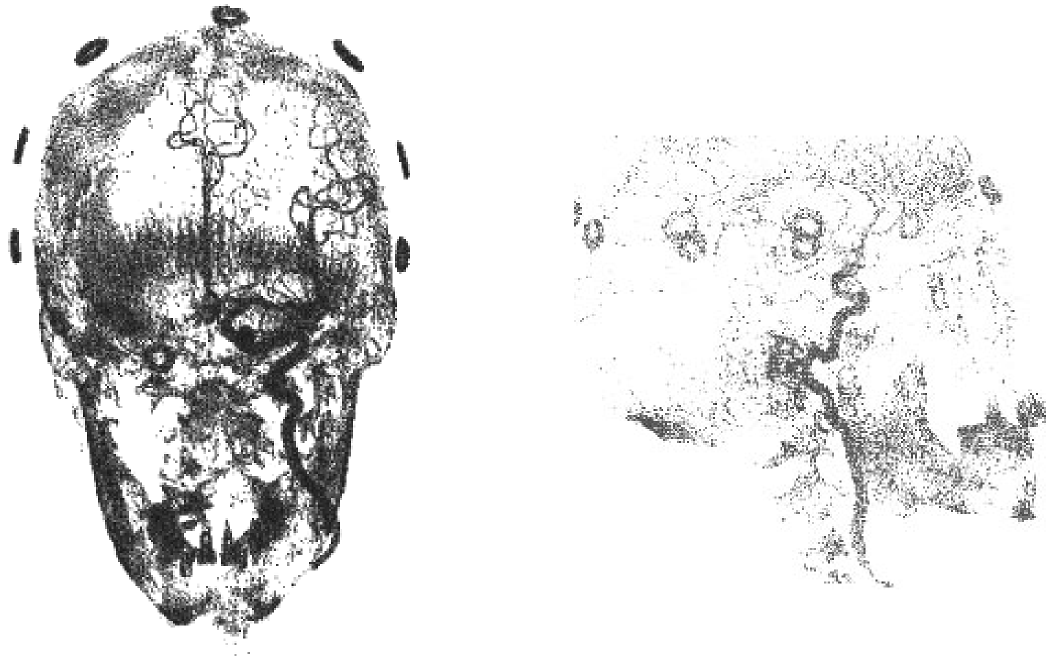


Figure 7: MIP display of an angiogram from different views and with different parameter settings.

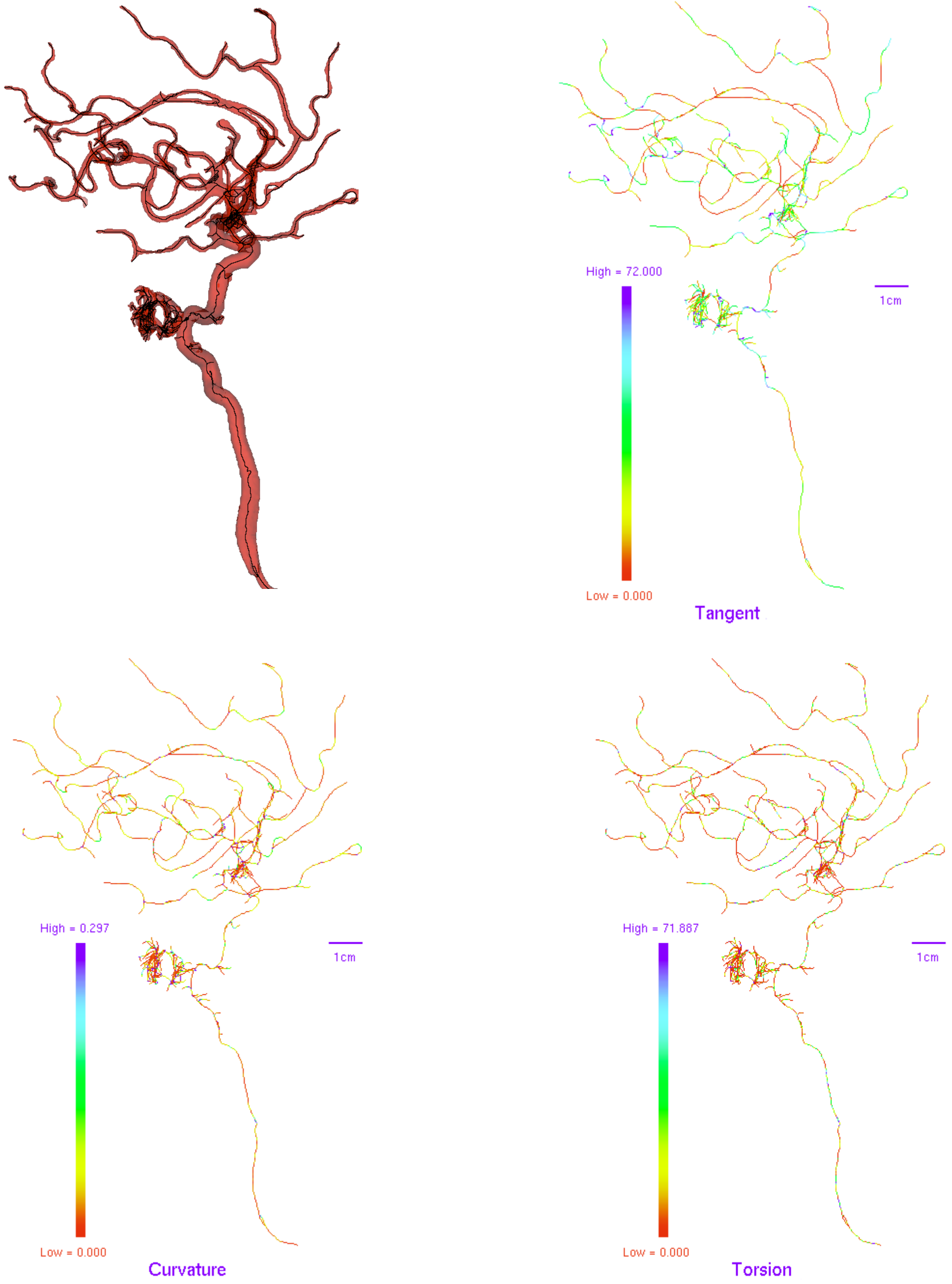


Figure 8: The left top panel shows the surface-rendered view of the skeleton. The smoothed-skeleton of the angiogram of Figure 7 is superimposed on a surface rendered display (red). Skeleton paths with length less than 2 voxels are cut off. Other panels show pseudo-color coded display of orientation, absolute curvature and torsion.

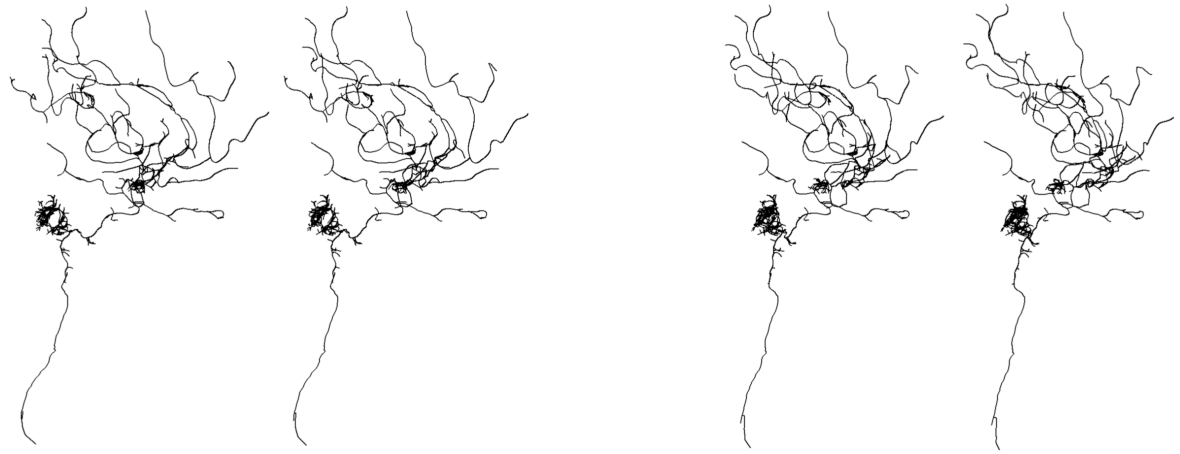


Figure 9: Two pairs of stereoscopic views. They may be fused into a 3-Dimage by uncrossing the eyes so that the left eye sees the left panel and the right eye sees the right panel. It is possible to realize that, despite the apparent complexity of the projection, the vessel system is actually divided into two subsystems quite far from each other, one in the back and one in the front.

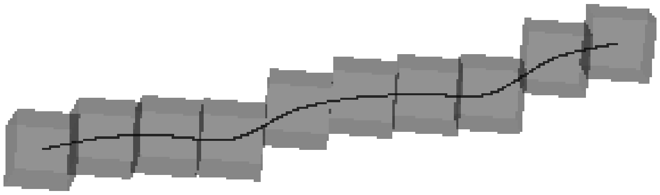


Figure 10: Close up view of a three dimensional skeleton segment (actual data).

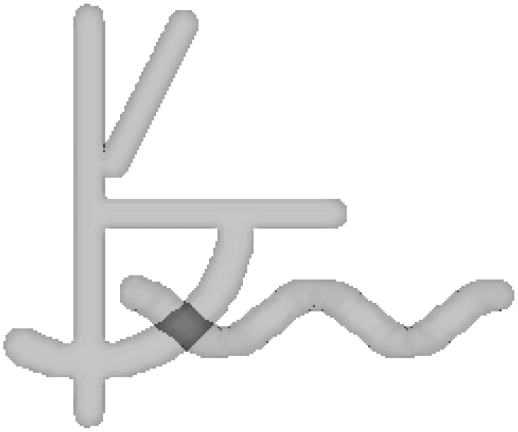


Figure 11: Surface-rendered view of the synthesized objects.

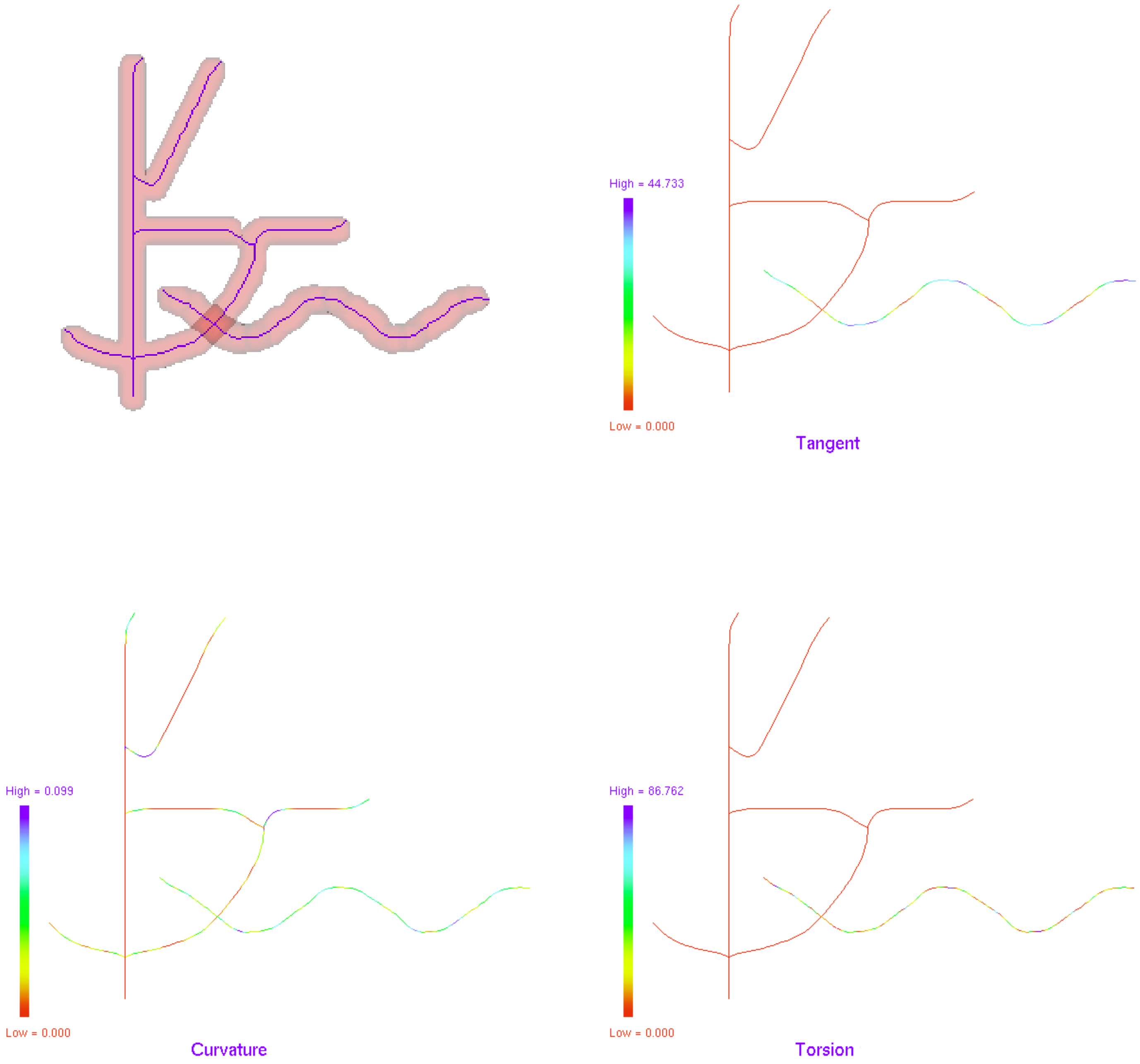


Figure 12: The left top panel shows the reconstructed object, surface-rendered, with its skeleton displayed as curves. Other panels show pseudo-color coded display of orientation, absolute curvature and torsion.

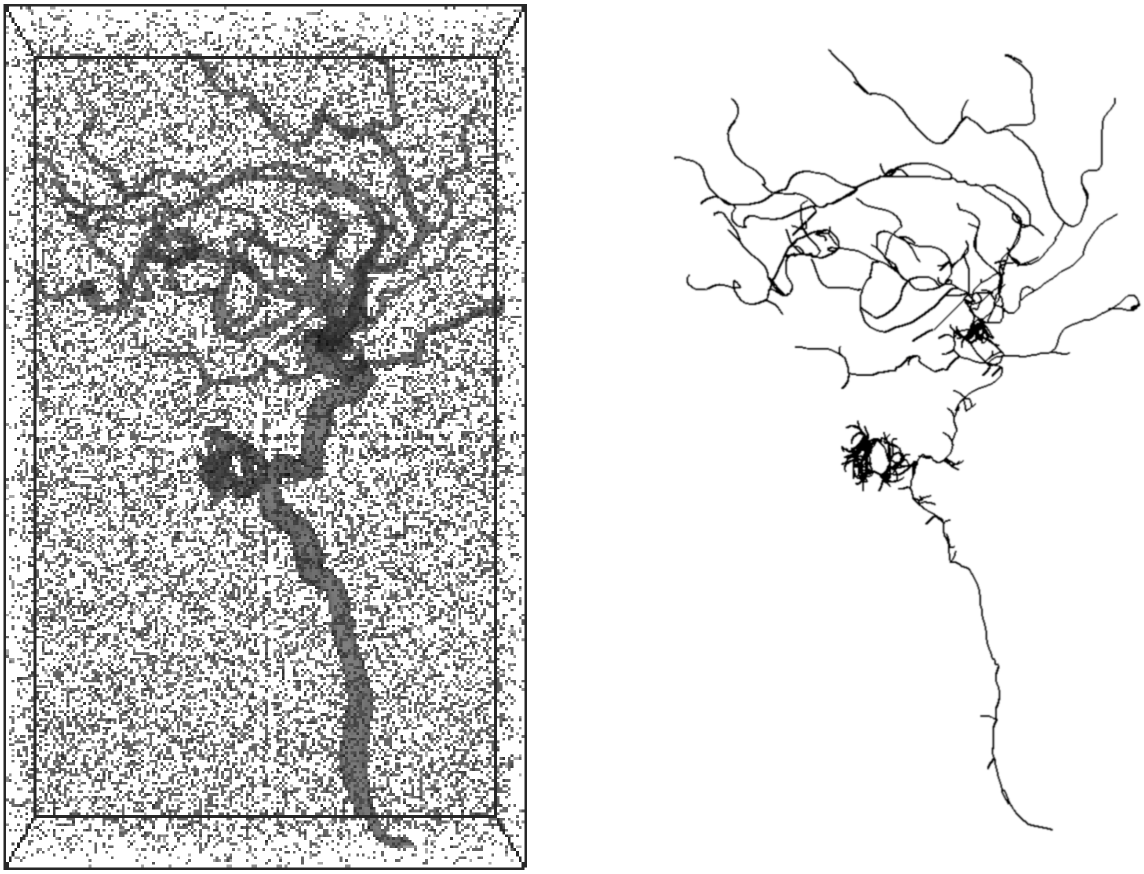


Figure 13: Surface rendering of image shown in Figure 8 with random noises added. The noises are added in such a way that one out of every 100 of the background voxels are noised to be an object voxel. The skeleton of the noised image shown in the right panel. Skeletons with length less than five voxels are cut off. There are 164 skeleton paths left for the major vessel system to be displayed and 5 for the minor vessel system.