Gap Filling of 3-D Microvascular Networks by Tensor Voting

Laurent Risser*, Franck Plouraboué, and Xavier Descombes

***Abstract*—We present a new algorithm which merges discontinuities in 3-D images of tubular structures presenting undesirable gaps. The application of the proposed method is mainly associated to large 3-D images of microvascular networks. In order to recover the real network topology, we need to fill the gaps between the closest discontinuous vessels. The algorithm presented in this paper aims at achieving this goal. This algorithm is based on the skeletonization of the segmented network followed by a tensor voting method. It permits to merge the most common kinds of discontinuities found in microvascular networks. It is robust, easy to use, and relatively fast. The microvascular network images were obtained using synchrotron tomography imaging at the European Synchrotron Radiation Facility. These images exhibit samples of intracortical networks. Representative results are illustrated.**

***Index Terms*—Gap filling, skeleton, tensor voting, vessel extraction, X-ray imaging.**

I. Introduction

THE extraction of vascular networks is an important problem in numerous medically oriented image analysis tasks. In this context, many methods for extracting vessel networks from noisy images have been developed using energy minimization criteria [1], type combination and multiscale techniques [2], mathematical morphology [3], and deformable models [4] to mention just a few. An important issue concerning these extraction techniques is that they "may cause the boundaries of a structure to be indistinct and disconnected" as pointed out in [4]. In an attempt to solve the problem, some authors have proposed specific methods associated with the need for topological preservation in the extraction of vascular networks. Quek *et al.* [5] propose the use of digital wave propagation within the vessels from 3-D noisy gray-scaled images to force continuity in the resulting segmented network. Nevertheless, despite the large number of vessel segmentation methods put forward, the issue of gap filling has not been addressed very thoroughly in the context of 3-D medical imaging. It should not be forgotten that vascular network segmentation methods aim generally at minimizing the influence of noise so that the gaps between segmented vessels are implicitly minimized. However, the resulting networks are often disconnected. To the best of our knowledge, only two gap filling algorithms for vascular networks have been developed [6], [7]. Discontinuities in the network limit the topological analysis, a drawback which is becoming important in an increasing number of medical applications [8]. There is indeed a growing demand for better analysis of microvascular structures [9], the quantification of which necessitates specific image processing. The interest of such quantification is to permit a better understanding of the impact of microvascular structures on the hydrodynamic interactions between arteriolar and veinular systems. Improving vessel connectivity in such networks is thus a key issue, and it is the goal of the present paper.

Our work is directed toward a quantitative analysis of the capillary network organization. High-resolution images of microvascular networks are of interest in, for example, the study of strokes, or tumor angiogenesis [16]. The microvascular network images are obtained by high-resolution X-ray tomography [15]. The contrast between the vessels and the surrounding tissue is made very strong by injecting a highly concentrated X-ray contrast agent into the vascular network. The segmentation of the resulting 3-D image is thus fast and robust. Nevertheless, the preparation leaves some discontinuities in the network due to inhomogeneities in the contrast agent injection. Parts of the microvascular network are thus disconnected from each other in the segmented image. We can distinguish three typical cases of discontinuities.

1) Information is missing in a vessel. In this case, the vessel observed in the image is divided into two distinct parts [Fig. 1(a)].
2) Several voxels remain in an area where information is missing. A small set of connected voxels is, in this case, pointed to by adjacent vessels whose extremities were previously identified as terminal segments [Fig. 1(b)].
3) Information is absent just after a bifurcation. This configuration is associated with a vessel extremity pointing at another vessel [Fig. 1(c)].

This lack of information has to be minimized if microvascular networks are to be analyzed efficiently. For this purpose, we have developed a new gap filling method suitable for large 3-D vessel networks.

The paper is organized as follows. Section II reviews the existing literature on gap filling methods or methods which can

*L. Risser is with the IMFT UMR 5502 CNRS/INPT/UPS, Avenue du Pr. Camille Soula, 31400 Toulouse, France (e-mail: laurent.risser@gmail.com).

F. Plouraboué is with the IMFT UMR 5502 CNRS/INPT/UPS, 31400 Toulouse, France (e-mail: plourab@imft.fr).

X. Descombes is with the INRIA, 06902, Sophia Antipolis Cedex, France (e-mail: xavier.descombes@inria.fr).

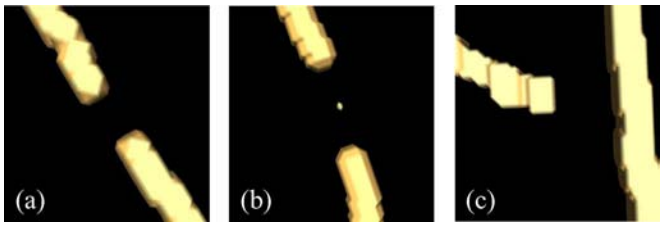

Fig. 1. (a) Two segment ends pointing at each other. (b) Small isolated set of interconnected vessel voxels between two segment ends. (c) Segment end pointing at a segment.

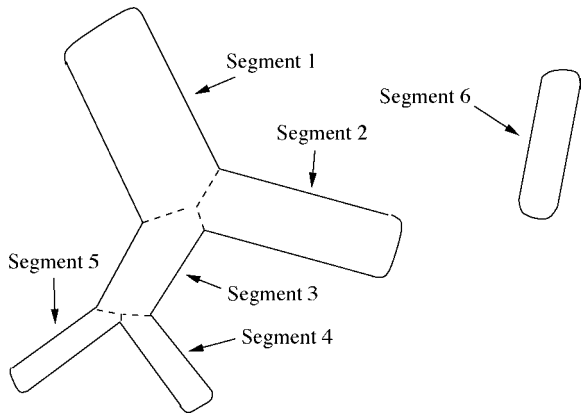

Fig. 2. Segments of a network of tubular structures.

be adapted to fill gaps between tubular structures. This review shows that most of the current gap filling methods which could have a reasonable algorithmic cost in 3-D are only designed for the first case of discontinuities listed above. Hence, none of these pointing methods enables the three typical cases of discontinuities to be filled. This is why we present, in Section III, a new efficient method based on the tensor voting formalism suited to our overall problem. The gap filling algorithm is presented in that section. In Section IV, the algorithm is tested on both real and phantom networks. Our gap filling algorithm is also compared to three other algorithms. Finally, conclusions are drawn in Section V.

II. Prior Work on Tubular Structure Gap Filling

In order to generalize the notion of imaging a 3-D vascular network, we will consider it as a network of tubular structures. In the following, the term segment refers to a part of this network. A segment may be defined between two bifurcations, between one bifurcation and one extremity of the tubular structure or between two extremities of the tubular structure (Fig. 2).

Although the problem of extracting tubular structures has not been widely addressed in 3-D, it has been the object of much attention in 2-D. For example, in the field of remote sensing, the extraction of road networks requires gap filling. Blockages, crossroads, shadows, and variations of the background or road intensity may cause gaps in the road network. In this context, Rochery *et al.* have improved a road extraction algorithm by using higher-order active contours [10]. This method is inspired by the active contours [11]. The evolution equation is modified by adding a force which makes neighboring regions attract one another. Bicego *et al.* [12] look for parallel straight lines in small subimages of streets using a Hough transform. The resulting small segments are then used to initialize a contour following algorithm, which starts in a direction perpendicular to the image gradient. In [13], road network midlines are first extracted. The midline endpoints are thus easily detected. They become the seeds of the gap filling algorithm. The nearby seeds are then compared with one another through a cost function and are merged. In [14], Lacoste *et al.* apply a global optimization solution with simulated annealing for gap filling in line networks. The cost function depends on the configuration of the overall line endpoint pairs.

Perceptual organization techniques [17] are especially interesting for the problem of gap filling in vascular networks. These techniques aim at grouping the primitives contained in a data set into perceptual structures. These primitives are usually called tokens. For example, Parent and Zucker [18] give a reliability measure on whether two isolated segment ends seem to be on the same arc of a circle. Experiments have been carried out on biomedical images to merge vessel pieces. In [19]–[21] gaps between tokens are filled with a fine to coarse process. Local junctions between tokens are processed first. Then the tokens are recursively redefined, so that larger tokens are compared for coarser junctions and so on. In [22], several features are distinguished in shape boundaries. They are thus merged locally without feedback. A good overview of perceptual organization techniques can be found in [23].

A rich expansion of perceptual organization techniques is tensor voting [24]. This technique was initially developed in order to reconstruct shapes from point clouds. It allows simultaneous communication among various types of tokens. Moreover, the information exchange between tokens uses tensor fields rather than scalar ones as in other methods. The communication is thus much richer. The initial 3-D tensor voting frameworks are extended to N-D in [25]. In [26], the curvature is added into the communication parameters. A multiscale algorithm for boundary inference is presented in [23]. Two methods adapted from tensor voting can be found in [27] and [28]. In [27], gray level images are filtered with Gabor filters. A tensor field is thus created using the directional features obtained from the filtering. The tensor field is used to fill gaps between segments. In [28], tensor voting is applied to curve endpoints in order to extrapolate and merge them. Textures are also extrapolated with N-D tensor voting. Therefore, tensor voting provides an interesting formalism for gap closure procedures in vessel networks.

III. Method

A. General Presentation

The key idea of the proposed algorithm is the use of three types of tokens: segment-end tokens, segment tokens, and island tokens (Fig. 3). Those tokens provide an efficient basis for the description of disconnected networks. The segment-end tokens contain information which allow isolated segment ends to

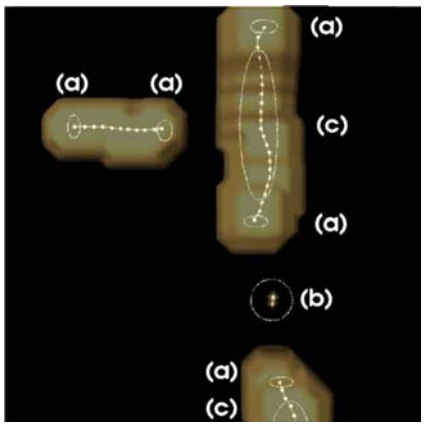

Fig. 3. Surface and skeleton of the sample of a segmented vascular network. The three kinds of tokens can be distinguished: (a) segment-end token; (b) island token; (c) segment token. A segment has to be long enough to be considered as a segment token. Hence, one of the segments here is not a segment token.

be described, while one segment token describes a segment except its extremities. The island tokens allows small sets of interconnected voxels to be specified.

The general algorithm is as follows. The initial vascular network images are considered as already segmented. We first skeletonize the networks. Skeletonization extracts the medial axis voxels from the shape described in the segmented image and associates each medial axis voxel with its distance to the nearest shape boundary. This description composes a homotopic, thin version of the shape. The shape becomes simply connected, a property which allows an easy description of the network to be made. The three types of tokens are then identified all over the skeletonized network. The tokens are thus used to fill gaps in the network by a method based on the tensor voting formalism. This efficient method is the main purpose of this paper. Once the gap filling process has been completed, the improved image is reconstructed.

B. Initial Data

As already mentioned, the images are assumed to be already segmented. The 1-voxels represent the shape and the 0-voxels the background. Any segmentation method can be processed since the segmented data represent a set of tubular structures without holes which entirely fill the tubular structure. These holes have to be filled because they change the topology of the tubular network. The method we use to fill discontinuities is presented in Section IV.

C. Skeletonization

Skeletons are widely used in vascular network study [29]–[32]. They reduce an object shape to the set of its medial points. Each medial point is associated with its coordinates and distance to the nearest object shape boundary. In 2-D, a skeleton is invertible. In 3-D, an invertible skeleton is a surface structure [33], [34]. However several skeletonization algorithms [30], [31], [34]–[36] provide a noninvertible curve structure in 3-D. Even for flattened shapes, these algorithms do not give rise to local surfaces, or shrink to nothing. Moreover, in the special case of nearly tubular shapes, the curve skeleton

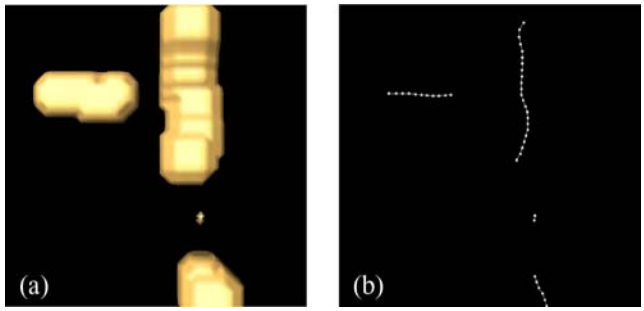

Fig. 4. (a) A segmented shape. (b) Skeleton of the shape. The points are the elements of the skeleton.

of the shape represents a negligible loss of information. Curve skeletons are thus a useful tool for describing tubular structure networks. They give an optimal representation of the data [37]. Moreover, they allow the segments ends, their orientation and their size to be easily detected. It can be seen that, for a shape, skeletonization and inverse skeletonization are the links between the segmented image representation and the network representation.

In our application, the skeletonization algorithm has to strictly preserve the homotopy properties of the original network. Indeed, it would not make sense to change the shape topology. It also has to roughly verify the medialness property. Finally, the third property of a skeleton, its thinness, has to be attained. The skeletonization algorithm class which best fits our needs is the thinning algorithms class. We thus used the algorithm proposed by Palágyi *et al.* [30], [36], which respects our needs and constraints.

The thinning algorithm is iterative. The voxels on the shape boundary are removed until they are part of a thin curve. 26-adjacency is adopted for 1-voxels (the shape), 6-adjacency for 0-voxels (the background), and 6-adjacency for the connectivity between 1-voxels and 0-voxels. The algorithm is subdivided into cycles of six subiterations. Each subiteration is characterized by a deletion direction $\mathbf{d}$ (up, down, north, south, east, and west in turn) and is as follows.

1) The 1-voxels which have a 0-voxel as direct neighbor in the direction $\mathbf{d}$ are taken into consideration.
2) The new value of those voxels will depend on their $3 \times 3 \times 3$ neighborhood. For each direction $\mathbf{d}$, a set of six masks of size $3 \times 3 \times 3$ and their rotations around $\mathbf{d}$ is defined. These masks are described in [36]. A voxel under consideration is then to be deleted if its $3 \times 3 \times 3$ neighborhood matches at least one of the given set of masks.

The six subiteration cycle is repeated until no voxel is deleted in a cycle. For further information on the thinning algorithm, refer to [30] or [36]. Note that any other thinning algorithm which respects our constraints could be used. Fig. 4 illustrates the result of the thinning algorithm.

The thinning algorithm provides a collection of disarranged skeleton voxels which correspond to the medial points of the

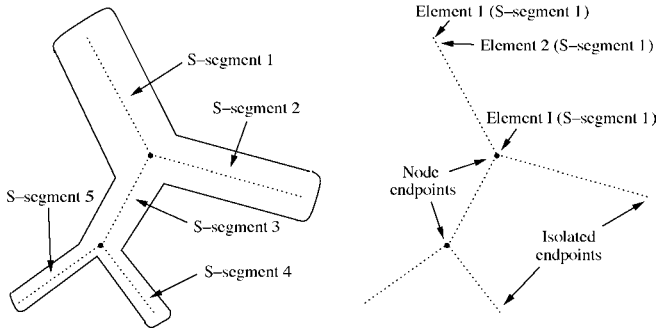

Fig. 5. Notations used for the skeleton description.

shape. They are merged and ordered in the step of skeleton pruning as follow. They are gathered in a chains set where each chain contains the skeleton voxels of a segment. In each chain, the skeleton voxels are ordered following their proximity. The following notations will be used. A skeleton voxel is an element. Each element contains its coordinates and the distance to the nearest boundary of the described shape. For a tubular shape, this distance is the radius of the shape local section. Each chain of elements is an S-segment (Fig. 5).

Before pruning, the S-segment endpoints must be characterized as either isolated endpoints or node endpoint. Let $N^1_{3\times3\times3}(\nu)$ be the 1-voxels in the $3 \times 3 \times 3$ neighborhood of ν. Since 26-adjacency was adopted for 1-voxels, an isolated endpoint ν is characterized by $\mathrm{card}(N^1_{3\times3\times3}(\nu) \setminus \{\nu\}) = 1$, a line element by $\mathrm{card}(N^1_{3\times3\times3}(\nu) \setminus \{\nu\}) = 2$ and a node endpoint by $\mathrm{card}(N^1_{3\times3\times3}(\nu) \setminus \{\nu\}) \geq 3$.

The pruning step will order the elements in each S-segment following their connectivity. The pruning algorithm is described in Appendix I. It is noteworthy that the algorithm outputs two lists N and E. For each node, the identifiers of the connected S-segments are gathered together in a sublist of N. The list E indicates the coordinates of the isolated S-segment endpoints in the pruned skeleton.

Boundary fluctuations may result in some extraneous skeleton branches. We consider as an extraneous branch a S-segment which has only one extremity connected to an other S-segment and which has a length smaller than 1.5 times the radius of the connected S-segments. Tests have shown that about 10% of the S-segments of our skeletonized networks could thus be considered as small extraneous skeleton branches. Of those 10%, about 8% are useful for the reconstruction of the network. Likewise, on a network where 134 gaps are closed if those extraneous skeleton branches are removed, none of these gap closures are lost if the branches are not removed. Finally, one can note that no fake junction is observed whether the extraneous skeleton branches are removed or not. We then consider that those small S-segments are more of help than a hindrance. Hence, they are not filtered.

Finally, in order to provide smoother skeletons, the integer coded coordinates of the skeleton elements are cast into float coded coordinates. The coordinates of the skeleton elements within the S-segments are thus smoothed by linear filtering. The isolated curve endpoint coordinates are not changed and further estimation of segment end direction is thus improved.

It should be noted that our gap filling method could be used directly on a skeletonized network since the lists N and E are known.

D. Tokens Used

The end and the size of each S-segment are immediately identified in the skeleton. The isolated S-segment endpoints are saved in the list E. The orientation of each S-segment end is easily estimated by computing the coordinates of the skeleton S-segment endpoint minus the coordinates of an element near the endpoint. Each end-segment orientation is generally different.

Let β be the threshold number of S-segment elements which differentiates an island from a segment. We take $\beta = 6$ elements. In order to define the tokens, we will distinguish three kinds of S-segments:

1) 0-segments, which have no connection to another S-segment;
2) 1-segments, which have just one end connected to another S-segment;
3) 2-segments, which have their two ends connected.

The tokens are then defined as follows.

- *Segment-end token*: A nonconnected segment end. The S-segment size is larger than β elements. Coordinates and a direction certainty are associated with each segment-end token. It can be seen that the 1-segments of size smaller than or equal to β elements do not generate segment-end tokens. Indeed, due to their size, their directional aspect is not relevant.
- *Island token*: A 0-segment whose size is smaller than or equal to β elements. Coordinates of the element at the middle of the S-segment are associated with each island token. There is, however, neither a directional aspect, nor a volume associated with it.
- *Segment token*: A 0-segment of size larger than 2β elements or a 1-segment of size larger than β elements or a 2-segment of size larger than 3 elements. For the 1-segments and 0-segments, the segment token expression must not interact with the segment-end token expression. A margin of β elements is thus left close to the endpoints. For the 2-segments, at least one element of the segment must not be an endpoint. A chain of element tokens is associated with each segment token. These element tokens are the elements of the corresponding S-segment, plus the following internal information: coordinates, radius, and direction uncertainty around the segment tangent. The radius expresses the volume of each element.

One can remark that 0-segments which are larger than β elements and smaller than 2β elements only generate two segment-end tokens. Their directional aspect is indeed significant and a margin of β elements is required by their two segment-end tokens, so that they do not generate an island token or a segment token. On those S-segments, junctions are then only possible at their extremities.

E. Construction of Tensor and Scalar Fields

Two fields are required for the gap filling algorithm: a tensor field $\mathbf{T}$ and a scalar field S. Each field can be seen as a 3-D

image of tensors for $\mathbf{T}$ and of scalars for S. Both $\mathbf{T}$ and S are of $[I, J, K]$. This size has to be large enough to contain the whole network. The initial segmented image size plus small margins is the best choice. The tokens will express themselves in $\mathbf{T}$ and S.

Let S be the token map. The tokens express their coordinates and their identifier in this map. An identifier is given to each token. Each segment-end token or island token expresses itself in the token map by giving its identifier value to the voxels corresponding to its coordinates. The segment token identifier takes the whole segment volume except close to its endpoints. In Fig. 7(a), the three kinds of tokens are illustrated using a token map.

Let $\mathbf{T}$ be the tensor voting field. This discrete second order tensor field has the same size $[I, J, K]$ as the token map. Each tensor is represented by a 3×3 matrix. The tokens as a whole express their direction certainty in $\mathbf{T}$. Let us break $\mathbf{T}$ into several tensor fields, each expressing a token directional aspect. Let $\mathbf{TE}_n$, $n \in [1, N]$ be the tensor field created by the nth segment-end token, $\mathbf{TI}_m$, $m \in [1, M]$ the tensor field created by the mth island token and $\mathbf{TS}_q$, $q \in [1, Q]$ the tensor field created by the qth segment token. Thus, for all points $[i, j, k]$

$$\mathbf{T}(i,j,k) = \sum_{n=1}^{N} \mathbf{TE}_n(i,j,k) + \sum_{m=1}^{M} \mathbf{TI}_m(i,j,k) + \sum_{q=1}^{Q} \mathbf{TS}_q(i,j,k). \quad (1)$$

This formula represents the communication between tokens. The construction and expression of each token shows numerous similarities with the stick voting fields and ball voting fields proposed in [23] or other papers of Medioni *et al.* In those papers, the tensor fields express a direction uncertainty. Due to the token expression, we consider in this paper that a direction certainty is more natural to consider. Our tensor fields thus have an embedded direction certainty. The construction of each kind of tensor field is described in the next subsection.

F. Segment-End Tokens Expression

The segment-end tokens express an orientation certainty. Their tensorial expression thus is close to a stick voting field of [23]. We assume a point P close to the S-segment end O. The tangent to the S-segment in O is $\mathbf{V}$. First, $\mathbf{V}$ and $\mathbf{OP}$ are chosen as normalized unit vectors ($\|\mathbf{V}\| = \|\mathbf{OP}\| = 1$). Let $\mathbf{W}$ be such that

$$\mathbf{W} = 2\mathbf{OP}(\mathbf{OP}.\mathbf{V}) - \mathbf{V}. \quad (2)$$

The vector $\mathbf{W}$ is the oriented tangent at P to the circle $\mathcal{C}$ which contains O and P and which is tangent to $\mathbf{V}$ at O (Fig. 6).

As in [23], $\mathbf{W}$ is weighted as follows:

$$\mathbf{w}^* = e^{-\left(\frac{r^2 + c\varphi^2}{\sigma^2}\right)} \frac{\mathbf{W}}{\|\mathbf{W}\|} \quad (3)$$

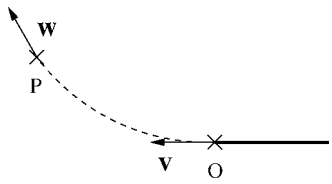

Fig. 6. Arc of circle $\mathcal{C}$ used for the construction of the tensor field expressing a segment-end token represented with dotted lines between points O and P.

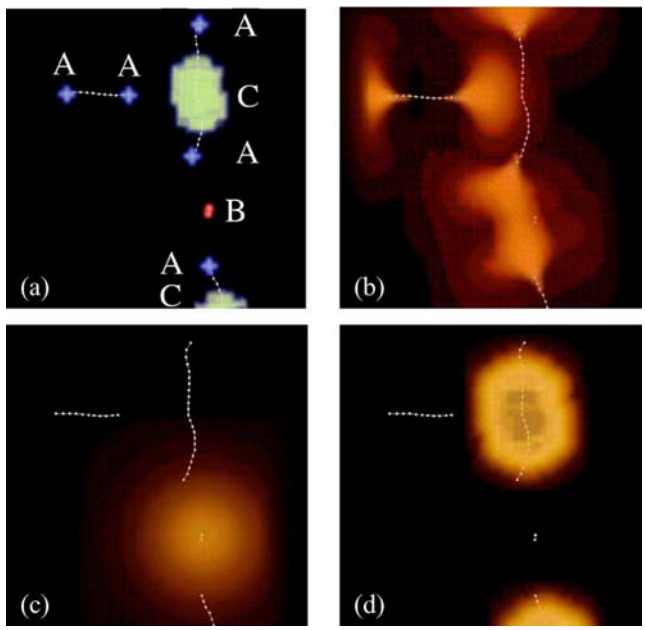

Fig. 7. Different classes of tokens and their associated tensor fields. (a) The letters refer to different token classes: A: segment-end tokens; B: island tokens; C: segment token. In the following figures, the energy $E(\mathbf{T})$ associated with each tensor $\mathbf{T}$ such that $E(T) = (\sum_{i=1}^{3}\sum_{j=1}^{3} \mathbf{T}_{i,j}^2)^{1/2}$ is represented. (b) Energy of segment-end voting fields $E(\mathbf{TE})$. (c) Energy of an island voting field $E(\mathbf{TI})$. (d) Energy of segment voting fields $E(\mathbf{TS})$. The segment token volume does not contain the field.

where r is the length of the arc $\widehat{OP}$ of the circle $\mathcal{C}$ and φ its curvature. The coefficient σ is the scale of the analysis and c the influence ratio between the proximity and the curvature. The choice of these coefficients is discussed in Section III-A. The tensor $\mathbf{TE}$ is then defined from $\mathbf{w}^*$ as $\mathbf{TE} = \mathbf{w}^* \otimes \mathbf{w}^*$, where $\otimes$ is the tensor product. At P, the tensor $\mathbf{TE}$ then corresponds to the vote of the segment-end token located at O. Fig. 7(b) shows the energy of voting fields generated by segment-end tokens to illustrate the concept. We denote by segment-end voting fields those voting fields.

G. Island Tokens Expression

The island tokens express a lack of preferential orientation. Their tensorial expression is then close to a ball voting field of [23]. Let C be the center point of the S-segment representing an island token and P a point close to C. The vector $\mathbf{CP}$ is then weighted as in previous case except that the length of the arc of

the circle is replaced by the distance between C and P and the curvature is no longer used

$$\mathbf{cp}^* = e^{-\left(\frac{\|\mathbf{CP}\|^2}{\sigma^2}\right)} \frac{\mathbf{CP}}{\|\mathbf{CP}\|}. \quad (4)$$

The tensor $\mathbf{TI}$ is then defined from vector $\mathbf{cp}^*$: $\mathbf{TI} = \mathbf{cp}^* \otimes \mathbf{cp}^*$. At P, the tensor $\mathbf{TI}$ corresponds to the vote of the island token located at C. Fig. 7(c) shows the energy of a voting field generated by an island token. The notation island voting fields is used for this kind of voting fields.

H. Segment Tokens Expression

Each element token E of a segment token has an uncertainty of orientation around an axis. This axis is the segment token tangent at E. In order to build the tensor voting field of a segment token we first create a vector field. Let P be a point near a given segment token.

At $P = [i, j, k]$, the vector field $\mathbf{V}$ is equal to $\mathbf{NP}$ with N the nearest element token to P. Its norm is then the distance to the skeleton. The algorithm for this vector field creation is described in Appendix II. In this algorithm, the scalar field A has the radius of the nearest element token N. The points P such that $\|\mathbf{V}(P)\| < A(P)$ are thus within the segment token. If P is not within the segment token, its distance to the segment boundary is $\|\mathbf{V}(P)\| - A(P)$. As for the other tokens, each vector outside the segment token is thus weighted

$$\mathbf{np}^* = e^{-\left(\frac{(\|\mathbf{V}(P)\| - A(P))^2}{\sigma^2}\right)} \frac{\mathbf{V}(P)}{\|\mathbf{V}(P)\|}. \quad (5)$$

The tensor $\mathbf{TS}$ is then defined from vector $\mathbf{np}^*$: $\mathbf{TS} = \mathbf{np}^* \otimes \mathbf{np}^*$. At P, the tensor $\mathbf{TS}$ corresponds to the vote of the segment token at the nearest point to N of its skeleton. Fig. 7(d) shows the energy of voting fields generated by segment tokens. We use the notation segment voting field for those voting fields.

I. Scale and Curvature Expression

The coefficients c and σ of (3)–(5) express the curvature and the scale, respectively, in the weights of the tensor fields. Evaluating the parameters c and σ is not convenient since they are dependent coefficients. In order to find a more intuitive way of fixing c and σ, we deduce them from a characteristic length L and a characteristic angle θ. To illustrate those parameters, let us consider the creation of a segment-end voting field. Let O be the origin of this voting field and $\mathbf{V}$ the normal vector which represents the direction certainty at O. The point A is such that $\mathbf{OA} = L\mathbf{V}$ and the point P such that $\|\mathbf{OP}\| = \|\mathbf{OA}\| = L$ and the angle $(\mathbf{OA}, \mathbf{OP}) = \theta$. Note that θ represents the angular wideness of the stick voting field and not the relative angle between two close S-segment ends. The curvature of the circle which passes through points O and P is $2\sin\theta/L$, as illustrated with dotted points in Fig. 8. The tangent to this circle at O is $\mathbf{V}$.

We consider that the weight of this voting field at the point A is e times smaller than at O. Likewise, the weight at P is e^2 times smaller than at O. Following these lines, the coefficients c and σ are given by

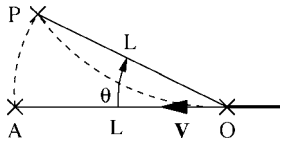

Fig. 8. Notations used for the calculation of c and σ from L and θ.

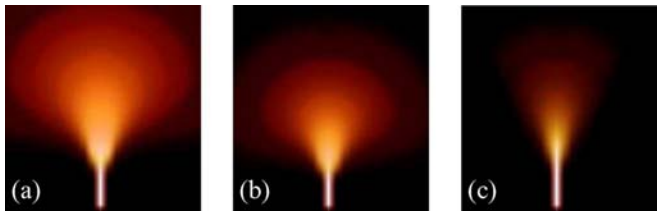

Fig. 9. Segment-end voting fields for different values of L and θ. (a) $L = 28\ \mu\mathrm{m}$ and $\theta = 45°$. (b) $L = 21\ \mu\mathrm{m}$ and $\theta = 45°$. (c) $L = 21\ \mu\mathrm{m}$ and $\theta = 25°$.

$$\begin{cases} c = \frac{L^3}{4(\sin\theta)^2} \\ \sigma = L \end{cases}. \quad (6)$$

Fig. 9 illustrates three segment-end voting fields for different values of L and θ.

Notice that, as explained in [24] or [38], for points which are farther than 90° along a spherical surface, a circular arc is not the most likely continuation. In practice, we then only consider the points P' such as $(\mathbf{OA}, \mathbf{OP'}) < 45°$ for a segment-end voting field construction.

For an island voting field or a segment voting field we only consider a distance L for which the weight is e times smaller than at the origin. The parameter σ is equal to L.

J. Saliency Map

As already proposed in [23] or [24] we compute the saliency map S to a curve through the tensor voting field $\mathbf{T}$ [Fig. 12(a)]. Let $\lambda_1(i,j,k)$, $\lambda_2(i,j,k)$, and $\lambda_3(i,j,k)$ be the eigenvalues of the tensor $T(i,j,k)$ with $|\lambda_1(i,j,k)| > |\lambda_2(i,j,k)| > |\lambda_3(i,j,k)|$. We compute the eigenvalues with the Jacobi algorithm of [39]. Since our choice for the tensor construction differs from those of [23] and [24], we self-consistently define the saliency to a curve at the point $[i,j,k]$ as

$$S(i,j,k) = \lambda_1(i,j,k) - \lambda_2(i,j,k). \quad (7)$$

We illustrate this choice in Fig. 10 where the vector field associated with the left segment-end token is drawn in the vicinity of its extremity. On the other hand, the vector field associated with the right segment token is also drawn. One can see that both vector fields are closely collinear in the region in between the segments. This collinearity induces that the resulting tensor map will have a large energy, leading to a high saliency map value,

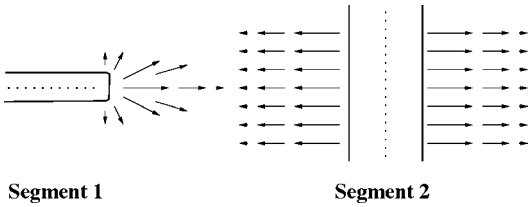

Fig. 10. Vector fields generated by a segment-end token (segment 1 end) and a segment token (segment 2). These fields are tensorized in order to permit the tokens communication.

which is precisely what is needed. Equation (7) then leads to high values for points which provide continuity of segments.

K. Token Junction

The tokens are merged with paths following the watersheds of the saliency map S [Fig. 12(b) and (c)]. Paths are generated iteratively. Each segment-end token is the seed of a path. Let P_i be a point of a path and $\mathbf{D}_i$ a direction associated with P_i. The point P_{i+1} is estimated as follows:

$$P_{i+1} = \arg\max \quad S(p) \quad \begin{cases} p \in N_{3\times3\times3}(P_i) \\ \mathbf{pP}_i.\mathbf{D}_i > 0 \end{cases}. \qquad (8)$$

The direction $\mathbf{D}_{i+1}$ is thus

$$\mathbf{D}_{i+1} = \mathbf{P}_{i+1}\mathbf{P}_i \qquad (9)$$

The path is stopped according to four criteria.

1) It leads to another segment-end token. The two S-segments are then joined by the path [Fig. 11(a)].
2) It leads to a segment token. The segment token is divided at the nearest element token to the junction point. The segment-end token joins the segment token at this point [Fig. 11(b)].
3) It leads to an island token. The segment-end token is then joined to the island token by the path [Fig. 11(c)]. The island token and its contribution to the tensor voting field are deleted. The expression of the newly created segment-end token is added to the tensor voting field [Fig. 11(d)]. Another watershed path thus starts from the end of the new segment to try to reach another junction.
4) It is stopped if the value of the saliency map is lower than a given threshold, if it reaches the boundary of the image or if its length is larger than a fixed threshold.

A junction list is created in order to avoid joining two tokens twice. This list also avoids the creation of artifact cycles between three segment-end tokens. The radii of the S-segments created depend on the radii in the segment-end tokens. When two segment-end tokens are joined, the radii are interpolated. If a segment-end token is joined to a segment token or an island token, the radius of the segment-end token is the same throughout the junction S-segment.

At each iteration, the junction test between the path and a token is computed using the token map S. When a path joins a token identifier, the token is then immediately recognized. The whole algorithm is then of order $O(E)$ with E the number of segment-end tokens. It is thus especially suitable for large networks.

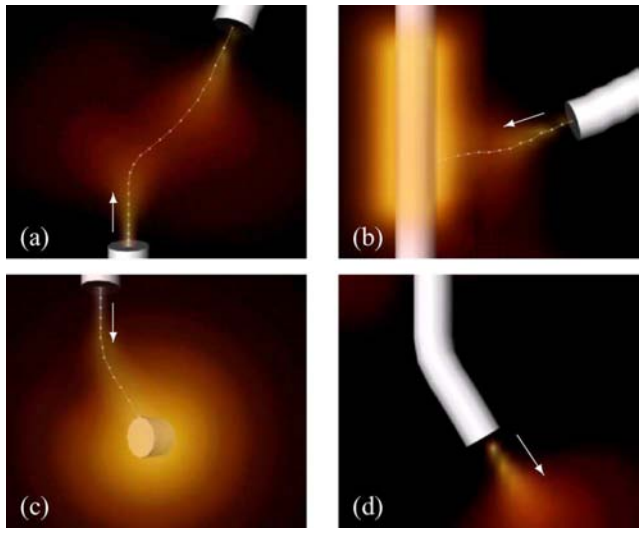

Fig. 11. Different cases of paths. (a) Segment-end token to segment-end token. (b) Segment-end token to segment token. (c) Segment-end token to island token (part 1). (d) Segment-end token to island token (part 2).

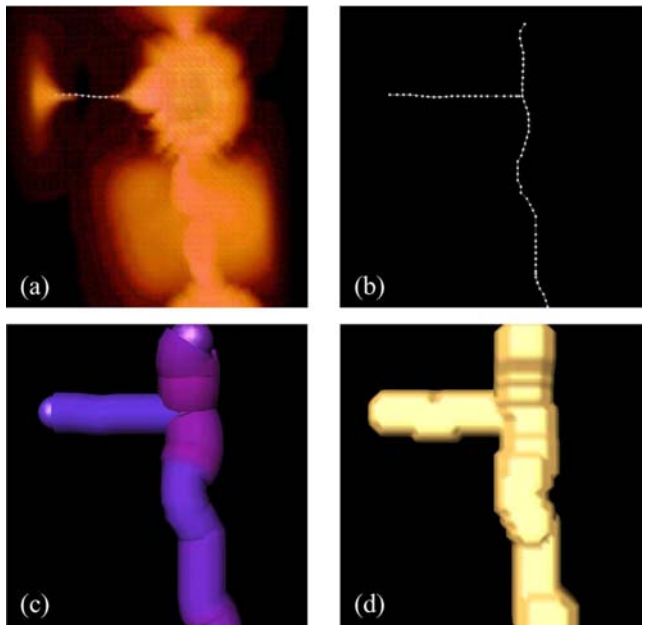

Fig. 12. (a) Saliency map. (b) Filled skeleton. (c) Filled skeleton with volume. (d) Dense Volume reconstruction.

L. Dense Volume Reconstruction

The volume is reconstructed within an image of the same size as the initial image. For each repaired skeleton element, the voxels having a distance to the element center smaller than the element radius are 1-voxels. The other voxels are 0-voxels [Fig. 12(d)].

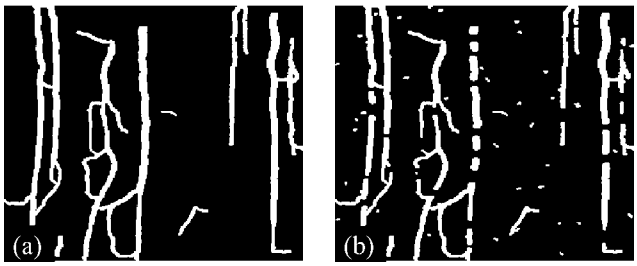

Fig. 13. Deterioration of a reference network. (a) Reference network. (b) Deteriorated network. 30 gaps have been created. Their mean size is 12 pixels. 50 islands have been injected for a size of 300×340 pixels.

IV. Application: Gap Filling in Microvascular Networks

A. Presentation

Our gap filling method was tested both on phantom networks and on real intracortical microvascular networks. The phantom networks were used in order to validate the method. They were obtained from clean reference networks which were artificially damaged. The comparison of their repair with the reference networks was then possible. Our algorithm was also compared to other algorithms. After this validation, the algorithm was tested on real intracortical microvascular networks. We also discuss how the parameters were chosen. All calculations were carried out on a Linux PC with an Intel Xeon 2,13-GHz processor and 4 GB of RAM. Each volume rendering has been obtained using Amira software.

B. Phantom Networks

1) Presentation: The validation of our gap filling method requires the comparison of reconstructed networks with an ideal result. It is also interesting to compare the efficiency of our gap filling method with other approaches. Hence, we compared our new method with a Bayesian Ising model, with mathematical morphology tools and with an active contour method.

2) Phantom Networks Creation: In order to simplify the comparison, the phantom networks are planar. A set of three small, thin networks was extracted from real intracortical networks. These samples are especially clean. They were then projected onto a plane. The crossings between segments due to the projection were erased. Then the networks were repaired with our tensor voting algorithm so that it could not improve them any more. These networks are the reference networks.

Damaged networks are obtained using the reference networks. Their deterioration is controlled. It is thus possible to measure the efficiency of the damaged network reconstruction. The deterioration is processed as follows. A given proportion of elements of the network is the center of gaps. The size of each of these gaps follows a Gaussian law. Once the network is disconnected, noise is added to the background. Finally, a given density of islands is homogeneously distributed around the disconnected network. Fig. 13 illustrates the deterioration of a reference network.

3) Methods Used for Comparison: Our gap filling method was compared to two well-known methods and a more sophisticated one. The first method is based on the Ising model and the second on mathematical morphology. Furthermore, we test a standard phase field model as a third choice.

The Ising model [40] is a Bayesian model which evaluates the probability of repairs from a deteriorated image. In order to be highly probable, the repaired image must share a proper balance between the deteriorated image and an homogeneous image. One parameter governs the relative importance of the initial image to the homogeneous one. In order to generate and test a wide range of combinations of possible repaired images and avoid falling into a local probability maximum, simulated annealing was used. It adds three other parameters to the method, an initial temperature, a decaying factor and the total number of iterations. The simulated annealing combined with the Ising model allowed the deteriorated image to be homogenized. Hence, the gaps between vessel like structures tended to be closed, as illustrated in Fig. 16(a), (d), and (g).

The method based on mathematical morphology [41], [43] has the following features. The deteriorated image is closed with a set of structuring elements. Each closure produces a partially repaired image. The repaired image is the union of all partially repaired images. The structuring elements used are thin segments of same length with several orientations. For example, a structuring elements set can be a set of thin segments of 15 voxels in length, with orientations of 0°, 30°, 60°, 90°, 120°, and 150° with the axis $\vec{x}$ of the 2-D image.

A brief introduction to phase field models is given in [42]. Since the contrast of our images is very high, a common active contours method would only find well injected vessel boundaries. By forbiding any change in the contour topology, a single contour which contains the whole vascular network could also arbitrarily connect the different connected components and fill the gaps. However, a phase field model might be suited to our problem. This approach consists in defining an energy associated to the contour of the image described by a phase field. This energy contains an internal term which tends to minimize the contour length and the surface defined by the internal part of the contour. It also contains an external term which tends to split the image into two regions in which the pixel values are modelled by two different Gaussian distributions. Following the gradient of this energy, the contour converges to a local minimum of the energy. In our case, applying this scheme leads to a contour which almost only delineates the different connected components obtained by thresholding.

Higher order active contours (HOAC) could have also been used. Nevertheless, HOACs introduce some long range interactions which define some stable shapes different from the circle. Especially, some particular models, leading to bars as stable shapes, have been applied to the problem of road detection [10]. However, these models require the definition of an interaction function dedicated to each specific application. They involve a double integral leading to time consuming algorithms. Besides, the extension of this approach to 3-D is still an open issue. Hence, the application of such a model to our problem is out of the scope of this paper. Besides, at the current time, these methods do not reach our time constraints to deal with 3-D volumes.

4) Parameter Influence: The parameters of the three methods have been varied so as to obtain the best possible results. We

L/θ. (voxels)/ (degrees)	Good Junctions (percents)	False Junctions (percents)	Time required (sec.)
5/45	85.7	0	39
10/45	85.7	0	53
15/45	95.2	0	91
20/45	95.2	0	271
30/45	85.7	4.7	717
15/15	70.8	0	94
15/30	70.8	0	93
15/45	95.2	0	91
15/60	12.5	4.7	90

Fig. 14. Gap filling algorithm efficiency on a 2-D damaged sample in function of the characteristic length (L) and the characteristic angle (θ). The percentage of good junctions is 100 times the number of good junctions over the number of gaps to fill. Likewise, the percentage of false junctions is 100 times the number of false junctions over the number of gaps to fill.

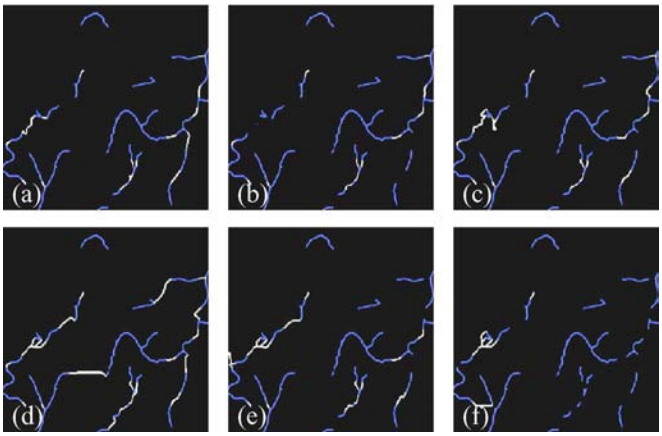

Fig. 15. Illustration of the results presented Fig. 14. Deteriorated networks are blue. Gaps to fill [image (a)] or filled (other images) are white. (a) Reference network. (b) $L = 5$ voxels and $\theta = 45°$. (c) $L = 15$ voxels and $\theta = 45°$. (d) $L = 30$ voxels and $\theta = 45°$. (e) $L = 15$ voxels and $\theta = 15°$. (f) $L = 15$ voxels and $\theta = 60°$.

first focus on the characteristic parameter influence of the tensor voting procedure. Results are shown in Figs. 14 and 15. A characteristic length of 15 voxels and a characteristic angle of 45° provide the best results for reasonable computation times. One can see that computation times clearly depend on the characteristic length but not on the characteristic angle. The results are relatively stable between small to optimal parameter values. For high parameter values, the number of good junctions tends to diminish and false junctions are observed. Indeed, excessive characteristic lengths allow the junction between segment-ends and nonneighbor segments. One can also note that characteristic angles higher than 60° should not be used since the path in the token junction step (Section III-K) tends to loop. The path then leaves the cone having an angular width of 45° where the segment-end voting fields are designed.

For information, the best results for the Ising method are obtained when the simulated annealing initial temperature is 10, the decaying factor equal to 10 and the total number of iteration set to 5000. Furthermore, the optimal structuring elements orientations set of mathematical morphology is 0°, 15°, 45°, ..., 165° for a 10 voxels length. Finally, the best results for the phase field models are given when the diffusion coefficient of the internal energy term is set to 1000 while the parameter associated with the external term is chosen equal to 3270.

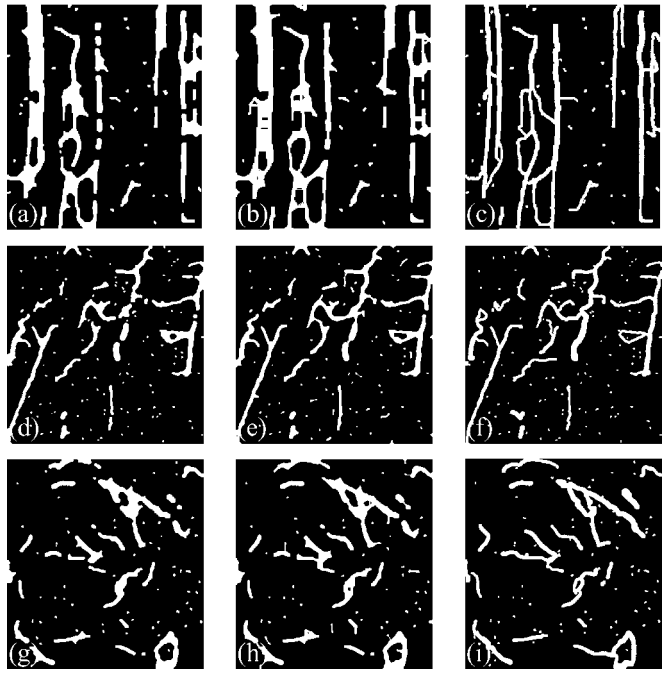

Fig. 16. Comparison of three gap closure methods with optimal parameters efficiency on three samples. (a), (d), (g) Ising method. (b), (e), (h) Mathematical morphology method. (c), (f), (i) Tensor voting method.

5) Tests Results: Fig. 16 illustrates the gap closure, for optimal parameters, with three methods on the deteriorated network presented in Fig. 13 and two other networks. In terms of computations times, the Ising model and the mathematical morphology method are faster than the tensor voting method. For example, result of image Fig. 16(a) requires 116 s when results of images Fig. 16(b) and (c), respectively, require 4 and 74 s. However, it can be seen that the tensor voting method gives much better results than the other two methods. One key point here is to realize that the Ising and mathematical morphology methods do not use any directional information about segments. The whole shape topology can be modified. For example, it can be seen on the left side of Fig. 16(a) and (b) that two parallel segments may become a single segment. On the other hand, small nodes may be entirely filled. This comparison shows that these two standard methods do not yield satisfactory results.

Fig. 17 illustrates the comparison between the phase field method and the tensor voting method. Phase field method is faster than tensor voting. For example, the results of the image Fig. 17(c) require 68 s whilst the results of the image Fig. 17(d) take 23 s. The whole shape topology can also be modified by phase field models. For example, one can see at the top of images Fig. 17(d) and (f) the merging of two closed segments. Nevertheless, such merging is not possible anymore when other parameters are used for the phase field model, as illustrated in image Fig. 17(e). The results that are exhibited by this standard method are however much better than those obtained with the other ones.

C. Real Networks

1) Description of Images: The real network images were obtained using synchrotron tomography imaging at the European Synchrotron Radiation Facility (ESRF) [15]. The resolution is

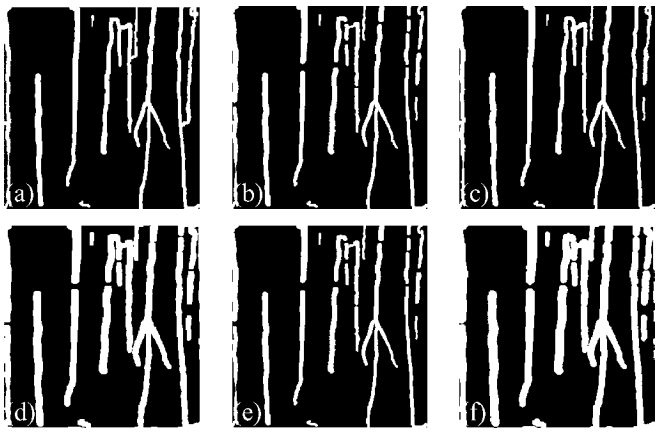

Fig. 17. Comparison of two gap closure methods efficiency. (a) Reference network. (b) Deteriorated network. 14 gaps have been created. (c) Gap closure with Tensor voting. (d)–(f) Results obtained with the phase field model/standard active contour.

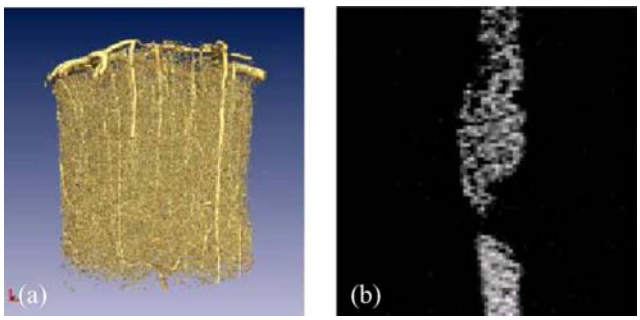

Fig. 18. (a) Isosurface of a sample. Scale bar is 100 μm. (b) Slice representing a badly injected vessel in gray levels.

about one micron, so that a 3-D image of the entire vascular network can be obtained on volumes as large as $8\ \mathrm{mm}^3$ [Fig. 18(a)]. The image contrast between the vessels and the surrounding tissues is very good. The vascular network was then binarized with hysteresis thresholding, leading to a very satisfactory result. However, the observed vascular network contains some discontinuities [Fig. 18(b)].

Small discontinuities within the vessels were filled in a simple and efficient way. We used a morphological closure operator with a $7 \times 7 \times 7$ cube structuring element. The size of this structuring element was set to the maximum typical size of discontinuities within the data. In order to fill the last discontinuities of background voxels remaining within the vessels, all connected sets of 0-voxels having sizes under a given threshold became 1-voxels. The segmented images were thus clean enough for the skeletonization.

Twenty networks were tested with our gap filling method. They contain from 5000 to 100 000 segments. Their mean diameter is 9.24 μm for a standard deviation of 5.41 μm.

2) Gaps Characterization: Our gap filling method requires two parameters, a characteristic length and a characteristic angle (Section III-A). In order to define them, we have studied the mean size of the gaps that we would have filled manually. We also characterized the tortuosity of the segments and took into account the segment diameters.

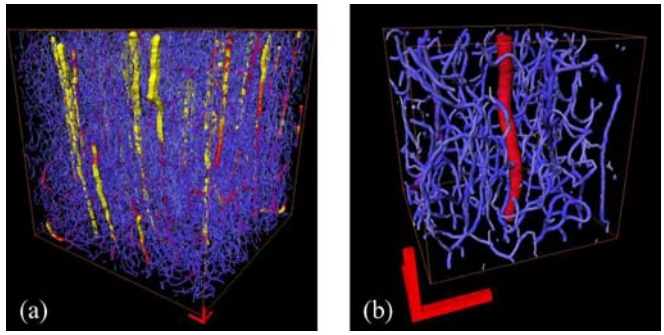

Fig. 19. Extraction of a sample of the network. The segments are colored according to their radius. Scale bar is 100 μm (73 voxels). (a) Full network. (b) Sample of the network.

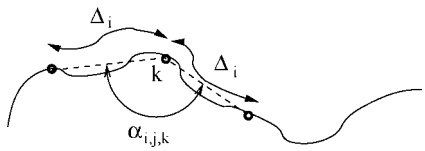

Fig. 20. Illustration of tortuosity estimation.

A typical sample of size $200 \times 200 \times 200$ voxels was extracted from a network (Fig. 19). The gaps of this sample were then detected and their sizes measured. This detection was supervised. In this sample, 31 gaps were detected. In those gaps, 10 are segment end to segment end gaps, 10 are segment end to island gaps, and 11 are segment end to segment gaps. A student test shows that, depending on the gap type, the mean gap size is the same in a confidence interval of 95%. The mean size of these gaps is 13.6 voxels (19.0 μm) for a standard deviation of 4.27 voxels (5.99 μm). One can note that, in the typical sample, the mean vessel diameter is 8.44 μm for a standard deviation of 3.61 μm. These values are slightly smaller than those found for all studied networks because the typical sample does not contain very large vessels. This is not penalizing for the tests because almost all the gaps were found on segments having a diameter in the range of 4–8 voxels (5.6–11.2 μm). Those segments are capillary vessels.

In order to estimate the segment tortuosity according to their diameter, we used the following method. The segments are broken down into several classes γ_j according to their mean diameter. Likewise several distances Δ_i are considered. Let us consider now the angle $\alpha_{i,j,k}$ between the line which passes through a point k of a segment of γ_j and the point at the curvilinear distance Δ_i on the same segment and the line which passes through the same point k and the point at the curvilinear distance Δ_i in the other direction (Fig. 20). For each class of segment γ_j and each distance Δ_i the mean angle $\overline{\alpha_{i,j}}$ is finally computed. Fig. 21 gives the results for two networks. This figure shows that thin vessels are more tortuous than wide ones, especially for distances Δ_i greater than 14 μm.

3) Parameter Definition: From the statistical analysis presented in the previous section the gap size does not depend on the vessel diameter in our samples. However, the vessel tortuosity at the gaps scale is higher for thin vessels than for wide

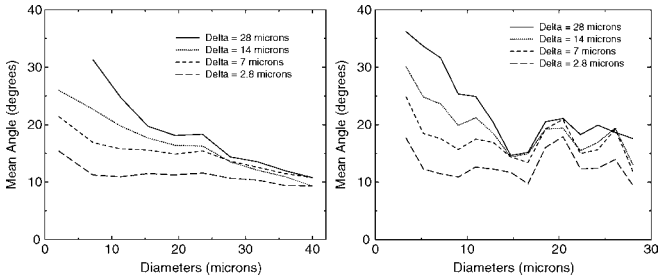

Fig. 21. Tortuosity estimation on two real 3-D networks. (a) First network. (b) Second network.

ones. It thus seemed interesting to adapt the curvature parameter of the tensor fields to segment diameter. The scale parameter only depends on the mean gap size.

We used the characteristic length L and the characteristic angle θ developed in Section III-A. The mean gap size (13.6 voxels) was used for length L. For the angle θ we used a coarse model. Its value was set to 40° for diameters smaller than or equal to 15 μm (10,7 voxels) and to 20° for wider diameters. We could also have used a finer model. The key idea here is that the parameters of the tensor fields can be easily adapted to the shape of the network.

4) Results: The results look perceptually natural. About 90% of the junctions which would have been connected by an expert are filled. This point is developed in Section IV-D. For a network of 318 segments embedded in an image of 200^3 voxels, the procedure takes 5 min and 35 s and requires 372 Mo. For a network of 2745 segments embedded in an image of 400^3 voxels, the procedure takes 49 min and 29 s and requires 2839 Mo. The average computation times and the amount of memory required are proportional to the image size. In order to limit the memory required in large images, the gap filling algorithm is applied in a sliding window. We use a sliding window of size 300^3 voxels for our tests. With this sliding window the treatment of a network of 57197 segments embedded in an image of 1024^3 voxels [Fig. 19(a)] takes 19 h and 12 min and requires only 1215 Mo. Almost all the computational time is spent on tensor map construction. Each tensor map is computed one point after another in a window surrounding the expressed token. Computational time could then be significantly improved with a more sophisticated tensor map construction.

D. Noise Influence

The influence of noise on our gap filling algorithm is now evaluated. Reference networks described in Sections IV-B and IV-C are now considered whilst creating artificial gaps in those networks. Several densities of noise islands are homogeneously injected into the networks as well as gaps. The noisy networks are then treated with our gap filling method. Figs. 22 and 23 represent noisy and gap filled networks.

The proportion of gaps filled versus noise is then evaluated. The number of fake junctions over the expected number of junctions is also computed. Fake junctions are due to two segment-end tokens joining the same noise island. They are uncommon and are only found on very noisy networks.

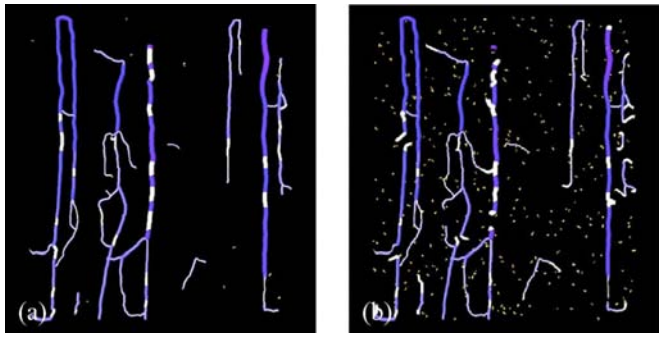

Fig. 22. Noise influence on a planar phantom network. The deteriorated network is dark blue. The gaps filled are white. Yellow noise islands surrounding the vessels. We consider a thickness of 20 voxels. (a) 5.94 noise islands for 10^6 voxels. (b) 198 noise islands for 10^6 voxels.

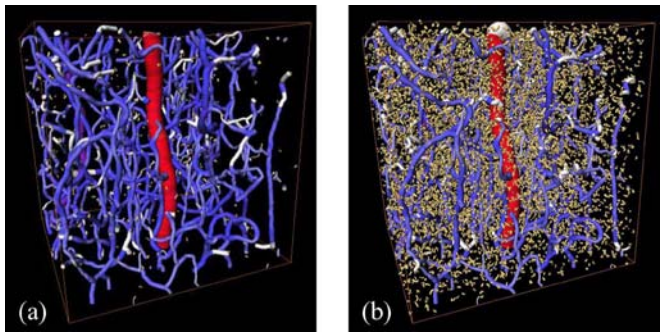

Fig. 23. Noise influence on a sample of 3-D network. The deteriorated network is blue and red. The gaps filled are white. Yellow noise islands surrounding the vessels. (a) 40 noise islands for 10^6 voxels. (b) 1280 noise islands for 10^6 voxels.

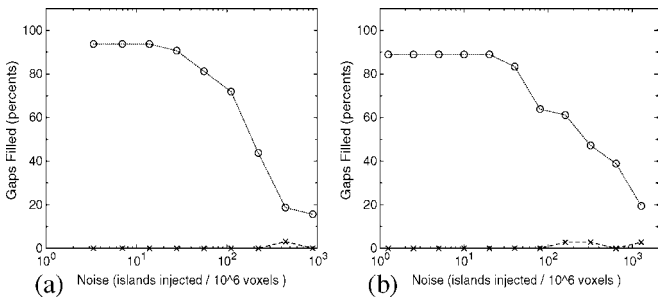

Fig. 24. Noise influence. Circles represent the percentage of good junctions. Stars represent the number of fake junctions over the expected number of junctions multiplied by 100. (a) Three-dimensional network of Fig. 22. (b) Three-dimensional network of Fig. 23.

As presented in Fig. 24, the number of good junctions is high and stable for moderately noisy networks. One can note that the voting field encodes proximity as well as continuity, so that two closely-packed vessels will not be merged into one. Then some gaps are not filled. For highly noisy networks, the number of good gaps filled decreases almost linearly in the semi-logarithmic representations of Figs. 24, indicating a noticeable influence on the noise ratio. At this stage, the gaps that are not filled anymore are mainly associated with segment-end tokens useful for the network reconstruction which join noisy islands.

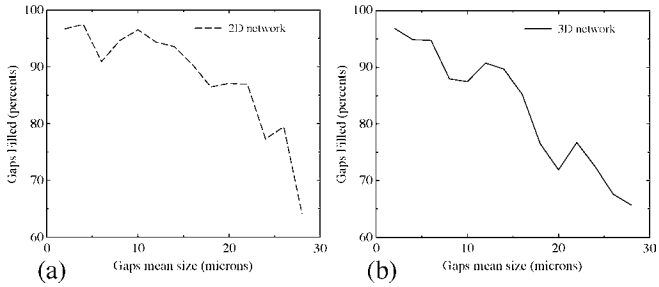

Fig. 25. Gap filling algorithm efficiency vs mean gap length and with given a characteristic length (15 μm) and angle (30°). The 2-D and 3-D reference networks are the same as those of Figs. 22 and 23.

E. Gap Size Influence

Finally, we test the influence of gap size on our gap filling algorithm. As in the previous subsection, reference networks described in Sections IV-B and IV-C are considered whilst creating artificial gaps in those networks. Gaps are homogeneously distributed in the networks. For each damaged network, their lengths follow a Gaussian law. Each reference network is then damaged according to several mean lengths. Standard values of the lengths are 0.3 times the mean lengths. Fig. 25 presents our gap filling algorithm's efficiency with fixed characteristic lengths and angles. The algorithm is very efficient for mean lengths lower than or equal to the characteristic length. The percentage of junctions decreases linearly for mean lengths slightly higher than the characteristic length. Then for higher mean lengths the algorithm is absolutely inefficient. Likewise, no fake junctions have been observed for our networks. This is due to the fact that the vessels are far enough from each other compared to the characteristic length of the chosen voting fields. A proper reparametrization of the voting field method might partially overcome this substandard performance.

V. Conclusion

We have proposed a new method for filling gaps in large 3-D images of tubular structures. The algorithm is designed so as to be robust to the typical configurations of gaps in such networks. The use of skeletons allows a robust and fast algorithm to be used. The information required for the description of tubular structure shapes is indeed minimized using curve skeletons. This tensor voting algorithm also avoids the comparison of token pairs. The whole algorithm is then of order $O(E)$ with E the number of segment-end tokens. It is thus especially suited to large networks. Tensor fields however require a large amount of memory. Nevertheless, this memory can be minimized using a sliding window. The method has been tested on high-resolution images of intracortical vascular networks. The network was improved by the joining of tokens that would have been perceptually attributed to the same vessel. The procedure was also robust to moderate noise compared to other methods. An interesting characteristic of our gap filling procedure is that it does not require numerous parameters or manual interventions. Moreover, its parameters can be intuitively fixed. The proposed method should prove very useful in the reconstruction of the topology of microvascular networks. We hope that it will help the analysis of network structures, so that the vascular territory of arteriolar vessels may now be investigated. Finally, it is interesting to note that the proposed method can be used for posttreating medical images, in angiography or scanners for example. The formalism can also be easily generalized to any other dimension, including 2-D, with a large number of applications.

Appendix I
Skeleton Pruning Algorithm

This algorithm takes as input the thinned image I_t. The image I_t is traversed. When a 1-voxel of I_t is found, its corresponding S-segment is extracted from the image and added to the skeleton.

{1: Initialization}

Create a new list N {$N(D)$ will contain the list of the S-segment identifiers connected to the node D}

Create a new list E {E will contain the list of the isolated S-segment end identifiers}

Create a new list S {S will contain the list of the S-segments}

{2: Segment Pruning}

for all 1-voxel P of I_t **do**

{I is traversed pixel after pixel}

if P is on a S-segment extremity **then**

The S-segment which contains P is traversed

This S-segment is added to S

This S-segment is removed from I_t

if the first S-segment extremity is a node **then**

$D \Leftarrow$ node identifier

The S-segment identifier is added to $N(D)$

else

The S-segment end identifier is added to E

end if

if the second S-segment extremity is a node **then**

$D \Leftarrow$ node identifier

The S-segment identifier is added to $N(D)$

else

The S-segment end identifier is added to E

end if

end if

end for

Appendix II
Vector Map Creation Around a S-Segment S

The algorithm takes as input a binarized image I and a S-segment S. The S-segment S comes from I. A null scalar field A

and a null vector field $\mathbf{V}$ are also taken into consideration. Both A and $\mathbf{V}$ have the same size as I. A vector map is created around the S-segment S. Each point of the vector field $\mathbf{V}$ indicates the closest element of S. Each point of the scalar field A which is not in the shape of I has for value the distance to the closest element of S. This value is 0 otherwise. The map is propagated. A list L_1 indicates the points already processed.

{1: Vector Map Initialization}

D the considered maximum distance to the S-segment.

Create the list L_1

for all Element N in S **do**

  $\mathrm{Coord}(N)$ the coordinates associated to N.

  for all Point $P \in N^1_{3\times3\times3}(\mathrm{Coord}(N))$ **do**

    $\mathbf{V}(P) = \mathbf{PN}$

    Put P in L_1

  end for

end for

{2: Vector Map Creation.}

$Changes \Leftarrow 1$

while $Changes = 1$ **do**

  $Changes \Leftarrow 0$

  for all P in the 26-neighbouroud of each point N of L_1 **do**

    if $\|\mathbf{NP} + \mathbf{V}(N)\| < D$ and ($\|\mathbf{NP} + \mathbf{V}(N)\| < \|\mathbf{V}(P)\|$ or $\mathbf{V}(P) = 0$) **then**

      $\mathbf{V}(P) \Leftarrow \mathbf{NP} + \mathbf{V}(N)$

      Put P in L_1

      $Changes \Leftarrow 1$

      if $I(\mathrm{Coord}(P)) = 1$ **then**

        $\mathbf{A}(P) \Leftarrow \|\mathbf{NP} + \mathbf{V}(N)\|$

      end if

    end if

  end for

end while

ACKNOWLEDGMENT

The authors would like to thank V. Gratsac (IMFT) for technical support, I. Jermyn (INRIA) for the tests of the phase field/ standard active contour model and proof reading, and C. Fonta (CERCO), P. Cloetens (ESRF) for data preparation and acquisition. The authors would also like to thank the reviewers for their constructive comments which helped to improve the presentation of the paper.

REFERENCES

[1] L. Lorigo, O. Faugeras, W. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C. Westin, "CURVES: Curve evolution for vessel segmentation," *Med. Image Anal.*, vol. 5, no. 3, pp. 195–206, 2001.

[2] K. Krissian, G. Malandain, and N. Ayache, "Model-based detection of tubular structures in 3-D images," *Comput. Vis. Image Understand.*, vol. 80, no. 2, pp. 130–171, 2000.

[3] F. Zana and J. C. Klein, "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1010–1019, Jul. 2001.

[4] T. McInerney and D. Terzopoulos, "Deformable models in medical images analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.

[5] F. K. H. Quek and C. Kirbas, "Vessel extraction in medical images by wave-propagation and traceback," *IEEE Trans. Med. Imag.*, vol. 20, no. 2, pp. 117–131, Feb. 2001.

[6] A. Szymczak, A. Tannenbaum, and K. Mischaikow, "Coronary vessel cores from 3-D imagery: A topological approach," in *Proc. SPIE—Med. Imag. 2005: Image Process.*, 2005, vol. 5747, pp. 505–513.

[7] T. Pock, C. Janko, R. Beichel, and H. Bischof, "Multiscale medialness for robust segmentation of 3-D tubular structures," in *Proc. Comput. Vision Winter Workshop*, Zell an der Pram, Austria, Feb. 2005, pp. 93–102.

[8] E. Bullitt and S. Aylward, "Three-dimensional blood vessel trees: Clinical needs and applications," in *RSNA Syllabus on Diagnostic Imaging Physics*. Oak Brook, IL: RSNA, 2005.

[9] A. Popel, A. Pries, and D. Slaaf, "Developments in the microcirculation physiome project," *J. Vascular Biol.*, vol. 36, pp. 253–255, 1999.

[10] M. Rochery, I. Jermyn, and J. Zerubia, "Gap closure in (road) networks using higher-order active contours," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Singapore, 2004.

[11] M. Kass, A. Witkin, and D. Terzopolous, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1987.

[12] M. Bicego, S. Dalfini, and V. Murino, "Automatic road extraction from aerial images by probabilistic contour tracking," in *Proc. Int. Conf. Image Process.*, 2003, pp. 585–588.

[13] H. Tesser and T. Pavlidis, "Roadfinder front end: An automated road extraction system," in *Proc. 15th Int. Conf. Pattern Recognit.*, Barcelona, Spain, 2000, pp. 338–341.

[14] C. Lacoste, X. Descombes, and J. Zerubia, "Point processes for unsupervised line network extraction in remote sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1568–1579, Oct. 2005.

[15] F. Plouraboué, P. Cloetens, C. Fonta, A. Steyer, F. Lauwers, and J. Marc-Vergnes, "High resolution X-ray imaging of vascular networks," *J. Microscopy*, vol. 215, no. 2, pp. 139–148, 2004.

[16] L. Risser, F. Plouraboué, A. Steyer, P. Cloetens, G. Le Duc, and C. Fonta, "From homogeneous to fractal normal and tumorous micro-vascular networks in the brain," *J. Cereb. Blood Flow Metabolism*, vol. 27, pp. 293–302, 2007.

[17] D. Marr, *Vision*. New York: W.H. Freeman, 1982.

[18] P. Parent and S. Zucker, "Trace inference, curvature consistency, and curve detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 8, pp. 823–839, Aug. 1989.

[19] A. Sha'ashua and S. Ullman, "Structural saliency: The detection of flobally salient structures using a locally connected network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1988, pp. 321–327.

[20] J. Dolan and E. Riseman, "Computing curvilinear structure by token-based grouping," in *Proc. 1992 IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 1992, pp. 264–270.

[21] E. Saund, "Labelling of curvilinear structure across scales by token grouping," in *Proc. 1992 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Champaign, IL, 1992, pp. 257–263.

[22] F. Heitger and R. Von der Heydt, "A computational model of neural contour processing: Figure-Ground segregation and illusory contours," in *4th Int. Conf. Comput. Vis.*, Berlin, Germany, 1993, pp. 32–40.

[23] W. Tong, C. Tang, P. Mordohai, and G. Medioni, "First order augmentation to tensor voting for boundary inference and multiscale analysis in 3-D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 594–611, May 2004.

[24] G. Guy and G. Medioni, "Inference of surfaces, 3-D curves, and junctions from sparse, noisy, 3-D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1265–1277, Nov. 1997.

[25] C.-K. Tang, G. Medioni, and M.-S. Lee, "N-dimensional tensor voting and application to epipolar geometry estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 829–844, Aug. 2001.

[26] C.-K. Tang and G. Medioni, "Curvature-augmented tensor voting for shape inference from noisy 3-D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 858–864, Jun. 2002.
[27] A. Massad, M. Babós, and B. Mertsching, "Perceptual grouping in grey-level images by combination of gabor filtering and tensor voting," presented at the 16th Int. Conf. Pattern Recognit. (ICPR02), Quebec City, QC, Canada, 2002.
[28] J. Jia and C.-K. Tang, "Image repairing: Robust image synthesis by adaptative nd tensor voting," in *Proc. IEEE CVPR*, Jun. 2003, pp. 643–650.
[29] C. Fouard, "Extraction de paramètres morphométriques pour l'étude du réseau micro-vasculaire cérébral," Ph.D. dissertation, Univ. Sophia Antipolis, Sophia Antipolis, France, 2005.
[30] Z. Chen and S. Molloi, "Automatic 3-D vascular tree construction in CT angiography," *Comput. Med. Imag. Graph.*, vol. 27, pp. 469–479, 2003.
[31] K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, C. Halmai, B. Erdőhelyi, and K. Hausegger, "A sequential 3Dd thinning algorithm and its medical applications," *Lecture Notes Comput. Sci.*, vol. 2082, pp. 409–415, 2001.
[32] N. Maglaveras, K. Haris, S. Efstratiadis, J. Gourassas, and G. Louridas, "Artery skeleton extraction using topographic and connected component labelin," *IEEE Comput. Cardiol.*, vol. 28, pp. 17–20, 2001.
[33] P. Saha and B. Chaudhuri, "3-D digital topology under binary transformation with applications," *Comp. Vis. Image Understand.*, vol. 63, no. 3, pp. 418–429, 1996.
[34] G. Borgefors, I. Nyström, and G. S. Di Baja, "Computing skeletons in three dimensions," *Pattern Recognit.*, vol. 32, pp. 1225–1236, 1999.
[35] A. Imiya and M. Saito, "Thinning by curvature flow," *J. Vis. Commun. Image Rep.*, vol. 17, pp. 27–41, 2006.
[36] K. Palágyi and A. Kuba, "A 3-D 6-subiteration thinning algorithm for extracting medial lines," *Pattern Recogn. Lett.*, vol. 19, pp. 613–627, 1998.
[37] K. Büler, P. Felkel, and A. La Cruz, "Geometric methods for vessel visualization and quantification—A survey," in *Geometric Modelling For Scientific Visualization*, G. Brunnett, B. Hamann, and H. Muller, Eds. New York: Springer, 2003, pp. 399–421.
[38] G. Guy and G. Medioni, "Inferring global perceptual contours from local features," *Int. J. Comput. Vis.*, vol. 20, no. 1, pp. 113–133, 1996.
[39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, 2nd ed. Cambridge and New York, U.K.: Cambridge University Press, 1992.
[40] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. Providence, RI: Amer. Math. Soc., 1980.
[41] J. Serra, *Image Analysis and Mathematical Morphology*, 2nd ed. New York: Academic, 1982, vol. 1.
[42] M. Rochery, I. Jermyn, and J. Zerubia, "Phase fields models and higher order active contours," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Beijing, China, 2005, vol. 2, pp. 970–976.
[43] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1988, vol. 2.