Available online at www.sciencedirect.com
ScienceDirect
Medical Image Analysis 11 (2007) 567–587
MEDICAL IMAGE ANALYSIS
www.elsevier.com/locate/media

Probabilistic vessel axis tracing and its application to vessel segmentation with stream surfaces and minimum cost paths

Wilbur C.K. Wong *, Albert C.S. Chung

Lo Kwee-Seong Medical Image Analysis Laboratory, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Received 2 September 2006; received in revised form 17 January 2007; accepted 10 May 2007
Available online 2 June 2007

Abstract

We propose a novel framework to segment vessels on their cross-sections. It starts with a probabilistic vessel axis tracing in a gray-scale three-dimensional angiogram, followed by vessel boundary delineation on cross-sections derived from the extracted axis. It promotes a more intuitive delineation of vessel boundaries which are mostly round on the cross-sections. The prior probability density function of the axis tracer's formulation permits seamless integration of user guidance to produce continuous traces through regions that contain furcations, diseased portions, kissing vessels (vessels in close proximity to each other) and thin vessels. The contour that outlines the vessel boundary in a 3-D space is determined as the minimum cost path on a weighted directed acyclic graph derived from each cross-section. The user can place anchor points to force the contour to pass through. The contours obtained are tiled to approximate the vessel boundary surface. Since we use stream surfaces generated w.r.t. the traced axis as cross-sections, non-intersecting adjacent cross-sections are guaranteed. Therefore, the tiling can be achieved by joining vertices of adjacent contours. The vessel boundary surface is then deformed under constrained movements on the cross-sections and is voxelized to produce the final vascular segmentation. Experimental results on synthetic and clinical data have shown that the vessel axes extracted by our tracer are continuous and less jittered as compared with the other two trace-based algorithms. Furthermore, the segmentation algorithm with cross-sections are robust to noise and can delineate vessel boundaries that have level of variability similar to those obtained manually.

Keywords: Vessel segmentation; Probabilistic framework; Axis tracing; Stream surface; Minimum cost path

1. Introduction

Vascular disease is one of the major causes of death and disability worldwide and thus one of the leading causes of nursing home admissions and hospitalization in many countries. Taking the population of the United States of America in census year 2000 (≈281 million) as an example, there were 10–16 million people suffering from intracranial aneurysms, a kind of cerebrovascular disease that causes stroke upon rupture of the aneurysms (Rinkel et al., 1998); not to mention other types of cerebrovascular diseases such as carotid stenoses and arteriovenous malformations (AVM).

Vascular imaging is particularly essential to the diagnosis and prognosis of vascular diseases in a clinical environment—interventionists acquire three-dimensional (3-D) angiograms of a patient to comprehend the pathology and the peripheral vasculature; a treatment can then be planned with reference to those images; a new set of 3-D angiograms is often acquired after the treatment to assess the effectiveness of the operation. Hence, segmentation of those 3-D angiograms is invaluable. It provides patient-specific 3-D vascular models that can facilitate an effective and efficient diagnostic review of the vasculature. It also helps the radiologists/interventionists to further characterize vascular diseases by measuring clinical parameters of interest on these models, e.g. stenosis severity, the neck

* Corresponding author. Tel.: +852 2358 7000; fax: +852 2358 1477.
E-mail addresses: cswilbur@cse.ust.hk (W.C.K. Wong), achung@cse.ust.hk (A.C.S. Chung).

1361-8415/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.media.2007.05.003

width and the dome height of a cerebral aneurysm. In addition, segmentation is a common pre-processing step for other routines or analyses, such as visualization, volumetric measurement, quantitative comparison and image-guided surgery (Suri et al., 2002).

Nevertheless, vascular segmentation alone can be insufficient for: (a) studying quantitatively the vessel attributes over a particular vessel population (Bullitt et al., 2004); (b) generating fly-throughs in virtual endoscopy (Farag et al., 2004); or (c) performing real-time registration of 3-D vessels to X-ray angiograms (Kita et al., 1998). In such cases, extraction of the vessel axes is required instead. This usually demands further post-processing on the segmentations of the 3-D vessels. Topological thinning (Palágyi et al., 2001), voxel encoding technique (Zhou and Toga, 1999; Farag et al., 2004) and Voronoi diagram (Ogniewicz and Kubler, 1995) had been investigated for the construction of vessel axes from the segmentations. Hybrid approaches were also proposed (Flasque et al., 2001; Hassouna et al., 2005). In order to obtain a satisfactory vessel axis extraction with the above algorithms, a topologically and morphologically correct vascular segmentation (i.e., with no holes and cavities) is necessary, which however may be difficult to obtain from clinical data sets.

Axis-based vascular segmentation algorithms (Hernández-Hoyos et al., 2000; Wink et al., 2000; Aylward and Bullitt, 2002; Shim et al., 2005; McIntosh et al., 2006) provide an alternative means to kill two birds with one stone. In these algorithms, vessel axes are extracted segment-by-segment while the vessel boundary associated with each segment is delineated. Generally speaking, they consist of two core alternate steps:

(i) the generation of an axial tangent; and
(ii) the computation of the next axial point.

Segmentation is performed locally w.r.t. the axial point and its tangent to help determine the position of the next axial point. An axis-trace is initiated by a manually selected seed point. The axial tangent of the seed is then generated. This tangent helps to define a cross-sectional plane (normal plane) by shifting forward the current one along the tangential direction, on which the vessel is segmented and the next axial point is calculated as the centroid of the planar segmentation. This process keeps iterating until a termination criterion is met or the user preempts the tracing.

However, the trace obtained with the aforementioned methods may go astray and make sudden jumps in regions that contain furcations, diseased portions and kissing vessels (Wink et al., 2000). Re-initialization of the axis-trace at the proximal end of the side branch and the immediate disease-free and kissing-free portions is therefore inevitable. As a result, a disconnected vessel axis may be obtained and, more importantly, these methods may only produce an incomplete segmentation of the vessels of interest.

1.1. Our approach to vessel axis tracing and segmentation

In this paper, to alleviate the two following difficulties of:

(i) composing a topologically and morphologically correct vascular segmentation then extracting vessel axis from it; and
(ii) continuously tracing the vessel axis through regions that contain furcations, diseased portions and kissing vessels (aka problematic regions), in the meantime producing vascular segmentation,

we take a very different approach to tracing the vessel axes and segmenting the vessels of interest. First, we extract vessel axes from a gray-scale 3-D angiogram without any form of segmentation (e.g. edge detection (Hernández-Hoyos et al., 2000; Wink et al., 2000; Shim et al., 2005; McIntosh et al., 2006) or ellipse-fitting (Shim et al., 2005) on the cross-sectional planes). Flasque et al. (2001) and Aylward and Bullitt (2002) suggested an intensity-weighted mass center traversal algorithm and an intensity ridge traversal method, respectively, to extract tubular vessel axis. Our method, however, can produce continuous vessel axes in the problematic regions which may not be tubular in shape. This is achieved by allowing users to give guidance on axis tracing. With the extracted axis, we then segment the vessel of interest on the cross-sections that are derived from the axis. Such decoupling of segmentation and vessel-axis-tracing avoids incomplete segmentation of the vessel of interest due to axis-trace re-initialization and disconnected vessel axis in the problematic regions as in the aforementioned axis-based segmentation algorithms. Moreover, it is preferable to trace a vessel axis directly rather than compute it in a roundabout way from a segmentation that may be subject to topological or morphological incorrectness. Such incorrectness may lead to an undesired vessel axis which needs further rectification.

We observed that the vessel boundaries to be segmented on the cross-sections are mostly round, compared with those on either axial, coronal or sagittal image planes. This is especially true for the intracranial vessels, which are curved and thus rarely oriented normal to those image planes. Therefore, by segmenting vessels on the cross-sections, we can reduce the comparatively complex 3-D segmentation problem into a series of simpler 2-D vessel boundary delineation problems. As the final procedure, a 3-D vessel boundary surface model is constructed from the contours extracted on the cross-sections. The surface model is then deformed with the mesh vertices moving on their corresponding cross-sections rather than freely in a 3-D space. This helps avoid shrinkage of the model and bunching of the mesh vertices at regions that with high intensity gradient magnitude.

Our axis tracer works along the research lines of Aylward and Bullitt (2002), in which the vessels are analyzed on a scale-space. Nonetheless, we do not perform Hessian

matrix analysis as in the work. Instead, we embed the scale-space analysis in a probabilistic framework. In this framework (cf. Section 2), the axial point and its associated tangent are found in *a single step* as a maximum *a posteriori* (MAP) estimation, contrary to other algorithms (Hernández-Hoyos et al., 2000; Wink et al., 2000; Aylward and Bullitt, 2002; Shim et al., 2005; McIntosh et al., 2006) which take alternate steps to search those two quantities. Our method guarantees that the traced points are distanced from the preceding and the succeeding ones by a user-defined distance. This effectively helps avoid sudden jumps of the trace and large spatial discontinuities between the axial points. User-given guidance (2-D curves drawn on the screen by the user to guide the axis tracer through the problematic regions) can also be integrated seamlessly into the probabilistic framework as a prior probability density function (pdf).

In addition, the cross-sections derived from the traced axis are not necessarily flat planes as in Hernández-Hoyos et al., 2000; Wink et al., 2000; Shim et al., 2005. As pointed out by Yim et al. (2001), cross-sections of a curved vessel are prone to intersection problems if the density of the axial points is not low enough. In order to relax this planar constraint, we generate cross-sections with a commonly used flow visualization technique, namely stream surfaces (cf. Section 3). This technique guarantees that there is no intersection between the cross-sections. It offers a nice property that the vessel boundaries extracted on those surfaces are *ordered* in a 3-D space. In other words, a 3-D vessel boundary surface model with no self-intersection can be constructed by joining the vertices of adjacent contours. Also, extra cross-sections (and the extracted vessel boundaries) can be inserted between the existing ones without the need of re-computation, contrary to the method proposed in (Yim et al., 2001), where re-calculation of cross-sections is a must. In our framework, delineation of the round vessel boundary on a cross-section is accomplished by finding the shortest, globally optimal, circular path on a weighted directed acyclic graph (DAG) which is carefully constructed from the cross-section. We also allow the user to place anchor point(s) to guide the boundary extraction. This is achieved by offsetting some of the edge weights of the DAG.

Experimental results on synthetic and clinical data sets show that our axis tracer can extract less jittering vessel axis in regions that contain furcations, kissing vessels and thin vessels (in diameter <2 voxels) compared with the other two related works, (Aylward and Bullitt, 2002; Shim et al., 2005). With user guidance, our method can also produce continuous traces to side branches at furcations and bypass diseased portions. Moreover, we have evaluated our cross-sections built for the vascular segmentation. In contrast to the other methods (naïve method and Yim et al., 2001), our results have shown no intersection, surface artifact and severe uneven surface sampling amongst the cross-sections. It is also found that the proposed framework is robust to noise and can produce vascular segmentations that have level of variability similar to those segmentations obtained from human raters. We believe that our method is particularly good at segmenting the (diseased) vasculature of a small region of interest, in which users want to control the vessel axis extraction and segmentation process. Such application is essential to endovascular treatment planning, diagnosis and prognosis of vascular diseases in the clinical environment.

2. Probabilistic framework for axis tracing

2.1. Problem formulation

We formulate the vessel axis tracing as a problem that can be solved by iteratively and simultaneously finding (1) the next axial point $\boldsymbol{x}_{i+1}$, (2) the associated tangent $\hat{t}_{i+1}$ and (3) the scale r_{i+1} ($\propto$ width) of the vessel, given the current ones (indexed with i). These three quantities form the *solution vector* $\boldsymbol{p}_{i+1}$ of the axis tracing problem. Without loss of generality, one can express $\boldsymbol{x}_{i+1}$ and $\hat{t}_{i+1}$ in spherical coordinates $(\theta_x^{i+1}, \phi_x^{i+1})$ and $(\theta_t^{i+1}, \phi_t^{i+1})$ with respect to the reference frames whose Z-axes are parallel to $\hat{t}_i$ and $\hat{b}_i = (\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)/|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|$, respectively. We adopt the "double-cross" successive method (Bloomenthal et al., 1996) to compute the reference frame of $\boldsymbol{x}_{i+1}$ from that of $\boldsymbol{x}_i$ (the previous axial point). This method tries to minimize the rotation amongst successive reference frames. The x-axis of the reference frame to be calculated is expressed as $\hat{t}_i^{\perp} = \hat{t}_i \times \hat{t}_{i-1}^{\perp} \times \hat{t}_i$, where $\hat{t}_{i-1}^{\perp}$ is the x-axis of $\boldsymbol{x}_i$'s reference frame. The x-axis of the $\hat{t}_{i+1}$'s reference frame is given by, $\hat{b}_i^{\perp} = \hat{b}_i \times \hat{t}_i^{\perp} \times \hat{b}_i$. Therefore, the solution vector $\boldsymbol{p}_{i+1}$ is expressed as $[\theta_x^{i+1} \phi_x^{i+1} \theta_t^{i+1} \phi_t^{i+1} r_{i+1}]^{\mathrm{T}}$, given the vector $\boldsymbol{q}_i = [\boldsymbol{x}_i \hat{t}_i \hat{t}_i^{\perp}]^{\mathrm{T}}$ from which the reference frames of $\boldsymbol{x}_{i+1}$ and $\hat{t}_{i+1}$ can be derived. Fig. 1a shows a 2-D version for better illustration. The spherical coordinates ϕ_x^{i+1} and $\phi_t^{i+1} \in [0, \pi]$ denote the polar angles, whereas θ_x^{i+1} and $\theta_t^{i+1} \in [0, 2\pi]$ denote the azimuthal angles. Mathematically,

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + [\hat{t}_i^{\perp}(\hat{t}_i^{\perp} \times \hat{t}_i)\hat{t}_i] \begin{bmatrix} d \cos\theta_x^{i+1} \sin\phi_x^{i+1} \\ d \sin\theta_x^{i+1} \sin\phi_x^{i+1} \\ d \cos\phi_x^{i+1} \end{bmatrix} \tag{1}$$

and

$$\hat{t}_{i+1} = [\hat{b}_i^{\perp}(\hat{b}_i^{\perp} \times \hat{b}_i)\hat{b}_i] \begin{bmatrix} \cos\theta_t^{i+1} \sin\phi_t^{i+1} \\ \sin\theta_t^{i+1} \sin\phi_t^{i+1} \\ \cos\phi_t^{i+1}, \end{bmatrix} \tag{2}$$

where $d = \delta V$ is a user-defined step size of the trace and δV is the voxel size of the angiogram.

On a probabilistic framework, the solution vector is estimated by a maximum *a posteriori* (MAP) estimation, and is given as

$$\boldsymbol{p}_{i+1} = \arg\max_{\boldsymbol{p} \in \Omega} \overbrace{f(\boldsymbol{p}|\boldsymbol{q}_i, \hat{\mathscr{I}})}^{\text{posterior pdf}} = \arg\max_{\boldsymbol{p} \in \Omega} \left(\overbrace{f(\hat{\mathscr{I}}|\boldsymbol{q}_i, \boldsymbol{p})}^{\text{likelihood pdf}} \overbrace{f(\boldsymbol{p}|\boldsymbol{q}_i)}^{\text{prior pdf}} \right), \tag{3}$$

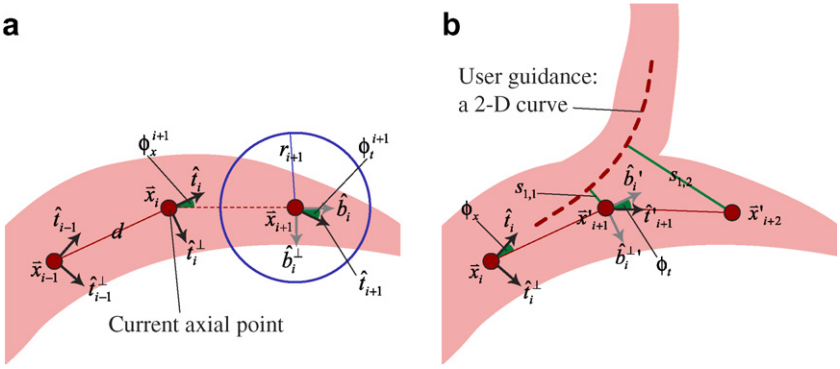

Fig. 1. Illustrations of (a) the probabilistic framework for axis tracing and (b) the tracing with user guidance on the probabilistic framework.

where Ω is the solution space, $f(\cdot)$ denotes a pdf and $\hat{\mathscr{I}}$ represents an appearance model of the subject (vessel axis) to be traced. Further discussion of the appearance model that we use in this work is presented in Section 2.2. Suppose the pdf can be expressed as $f(\cdot) = \frac{1}{Z}\exp(-U(\cdot))$, where Z is a normalization constant and $U(\cdot)$ denotes the corresponding energy function. Eq. 3 can be re-formulated as

$$\boldsymbol{p}_{i+1} = \arg\min_{\boldsymbol{p}\in\Omega}\left(\overbrace{U(\hat{\mathscr{I}}|\boldsymbol{q}_i,\boldsymbol{p})}^{\text{likelihood energy}} + \overbrace{U(\boldsymbol{p}|\boldsymbol{q}_i)}^{\text{prior energy}}\right). \tag{4}$$

$U(\hat{\mathscr{I}}|\boldsymbol{q}_i,\boldsymbol{p})$ and $U(\boldsymbol{p}|\boldsymbol{q}_i)$ are the corresponding likelihood and prior energies, respectively. Thier formulations are given in Sections 2.2 and 2.3. In order to avoid backward tracing, we set the domain of the polar angles to $[0, \pi/3]$. In addition, we force $r_{i+1} \in [(1-q)r_i, (1+q)r_i]$, where $q = 0.5$, such that we can control the allowable scale change in percentage of r_i to prevent sudden changes in r when the trace flies through bifurcations and diseased portions. Therefore, the solution space Ω of $\boldsymbol{p}_{i+1}$ is $[0, 2\pi)\times[0, \pi/3]\times[0, 2\pi)\times[0, \pi/3]\times[(1-q)r_i, (1+q)r_i]$.

2.2. Likelihood energy function

The likelihood energy function is built upon an appearance model. We employ a bounded spherical projection (BSP) image, generated with respect to $\boldsymbol{q}_i$, as the basis of the appearance model. The BSP image is created in a similar fashion as the spherical maximum intensity projection (MIP) image described in (van der Weide et al., 1998), except for the following: (1) our projection is bounded, i.e., the spherically casted rays are not extended infinitely, (2) the pixel intensities of the BSP image are the normalized cumulative sums of the voxels along the casted rays (i.e., the maximum and minimum intensities of the BSP image is 0 and 1, respectively) and (3) the polar coordinate is originated from the z-axis of a reference frame.

Figs. 2a–d show some of the BSP images generated on an image volume with a solid straight tube (intensities inside are higher than the outside) at different 3-D points and reference frames as depicted in Figs. 2e–h. It is observed that there is a special pattern (three horizontal strips) on the BSP image if (1) the length of the projection bound (later in this work, we bind this to the scale variable r of the solution vector $\boldsymbol{p}$) is greater than the tube radius (we set r equal to 1.5× the tube radius), (2) the 3-D point is on the tube's axis and (3) the z-axis of the reference frame is aligned with the tube's orientation, see Figs. 2a and e (note that there is no horizontal strip for the other three cases as illustrated in Figs. 2b–d). As such, we use this special BSP image as our appearance model $\hat{\mathscr{I}}$ in Eq. 4. Inspired by a commonly used similarity measure, sum of squared differences (SSD), the likelihood energy function is given as

$$U(\hat{\mathscr{I}}|\boldsymbol{q}_i,\boldsymbol{p}) = \frac{\sum_{\theta,\phi}\left(\mathscr{I}_{\theta,\phi}(\boldsymbol{q}_i,\boldsymbol{p}) - \hat{\mathscr{I}}_{\theta,\phi}\right)^2}{2\sigma_{\mathscr{I}}^2}, \tag{5}$$

where $\hat{\mathscr{I}}_{\theta,\phi}$ denotes the BSP intensity of the appearance model at the coordinates (θ,ϕ), $\mathscr{I}(\boldsymbol{q}_i,\boldsymbol{p})$ is the BSP image generated at $\boldsymbol{x}_i$ with the reference frame defined by $\hat{t}_i$ as the z-axis and $\hat{t}_i^{\perp}$ as the x-axis (recall $\boldsymbol{q}_i = [\boldsymbol{x}_i\hat{t}_i\hat{t}_i^{\perp}]^{\mathrm{T}}$) and the projective bound length r (the scale variable given in $\boldsymbol{p}$). The variable $\sigma_{\mathscr{I}} = 1/\sqrt{2}$ controls the tolerance to the pixel-wise differences between the BSP image $\mathscr{I}(\boldsymbol{q}_i,\boldsymbol{p})$ and the appearance model $\hat{\mathscr{I}}$.

2.3. Prior energy function

User guidance is expressed as a set of user-drawn 2-D curves on the screen. They help steer the axis tracer in 3-D by giving reference paths on which the perspective projected trace should follow on the screen. This idea is illustrated in Fig. 1b. Suppose the trace follows the main vessel from left to right without any guidance. If the user wants to guide the trace towards the side branch, a 2-D curve should be drawn on the screen as given. This is useful to make the tracer to fly-through the problematic regions in the angiogram.

We model such guidance in the prior pdf of our probabilistic framework as

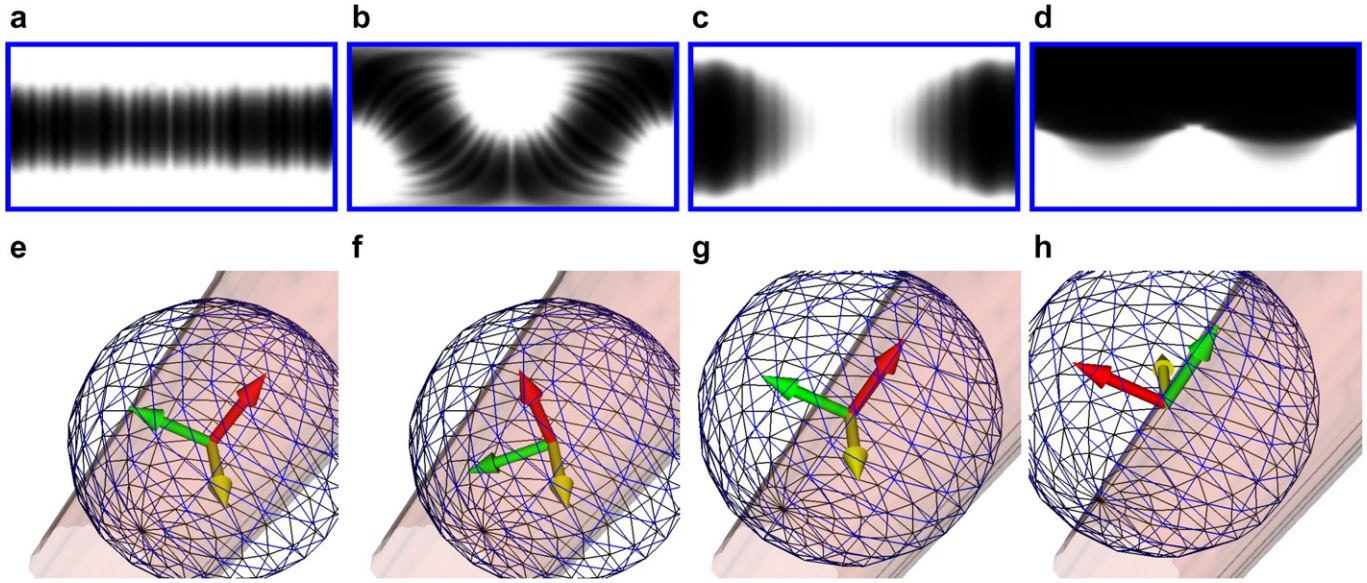

Fig. 2. (a–d) The bounded spherical projection (BSP) images (azimuthal and polar angles are enumerated in the horizontal and vertical axes, respectively) generated on an image volume with a solid straight tube (the surfaces are for visualization only) at different 3-D points and the associated reference frames (the red arrows give the z-axis directions) as depicted in (e–h). The blue frames in (a–d) are not parts of the BSP images, they are for better presentation of the images on white papers. The wireframe-spheres in (e–h) define the scope of the bounded spherical projection (i.e., their radius equals the length of the projection bound).

$$U_g(\boldsymbol{p}|\boldsymbol{q}_i) = \frac{\sum_{j=1}^{N}(s_{j,1}^2 + s_{j,2}^2)}{2\sigma_g^2}, \tag{6}$$

where N is the number of 2-D curves drawn by the user. The variables $s_{j,1}$ and $s_{j,2}$ are the shortest Euclidean distances (on the screen) from the perspective projected points of the succeeding two *candidate* axial points $\boldsymbol{x}'_{i+1}$ and $\boldsymbol{x}'_{i+2}$ to the j-th user-drawn 2-D curves, respectively. The term "candidate" refers to the quantities, annotated by the symbol prime ($'$), that are calculated on the basis of a solution vector $\boldsymbol{p} = [\theta_x \phi_x \theta_t \phi_t r]^{\mathrm{T}}$. The candidate axial point $\boldsymbol{x}'_{i+1}$ is obtained according to Eq. 1 with the spherical coordinates (θ_x, ϕ_x). $\boldsymbol{x}'_{i+2}$ is estimated from the candidate axial tangent $\hat{t}'_{i+1}$ as follows: $\boldsymbol{x}'_{i+2} = \boldsymbol{x}'_{i+1} + d\hat{t}'_{i+1}$, where $\hat{t}'_{i+1}$ is calculated from $\hat{b}'_i = (\boldsymbol{x}'_{i+1} - \boldsymbol{x}_i)/|\boldsymbol{x}'_{i+1} - \boldsymbol{x}_i|$, $\hat{b}_i^{\perp\prime} = \hat{b}'_i \times \hat{t}_i^{\perp} \times \hat{b}'_i$, θ_t and ϕ_t, according to Eq. 2. (See Fig. 1b for a pictorial illustration.) The variable $\sigma_g = 5$ screen pixels controls the guiding influence over the candidate axial points. A smaller energy value is obtained if $\boldsymbol{x}'_{i+1}$ and $\boldsymbol{x}'_{i+2}$ are closer to the curve on the screen, i.e. $\forall j,\ s_{j,1} \to 0$ and $s_{j,2} \to 0$. User guidance with more than one 2-D curve is demonstrated in Section 4.

Axis smoothness is modeled in the same fashion. The prior that favors smooth vessel axis is defined as

$$U_s(\boldsymbol{p}|\boldsymbol{q}_i) = \frac{(1 - |\bar{\hat{t}} \cdot \hat{b}'_i|)^2}{2\sigma_s^2}, \tag{7}$$

where $\bar{\hat{t}}$ is the mean vector of $\hat{t}_i$ and $\hat{t}'_{i+1}$, and $\sigma_s = 0.5$ controls the degree of the axis smoothness (the smaller σ_s is the smoother the axis). Therefore, the prior energy function in Eq. (4) is given as

$$U(\boldsymbol{p}|\boldsymbol{q}_i) = U_g(\boldsymbol{p}|\boldsymbol{q}_i) + U_s(\boldsymbol{p}|\boldsymbol{q}_i). \tag{8}$$

2.4. Initialization

The manually selected seed point may only be picked in close proximity to the vessel axis of interest. To initialize a trace, we need to find the first axial point $\boldsymbol{x}_0$ within the neighborhood of the seed point. This is done by locating the local maxima in a Gaussian smoothed version of the image volume. For the estimation of the initial projection bound length (i.e., the initial scale r_0) and the initial reference frame (defined by $\hat{t}_0$ and $\hat{t}_0^{\perp}$), we perform the analysis of the Hessian matrix and compute the ridge direction (analogous to $\hat{t}$) and the direction of maximum curvature (analogous to $\hat{t}^{\perp}$) at each scale (analogous to r) in a discrete scale-space (Aylward and Bullitt, 2002). Then, we set the values of r_0, $\hat{t}_0$ and $\hat{t}_0^{\perp}$ to the triple $(r, \hat{t}, \hat{t}^{\perp})$ that minimizes $U(\hat{\mathscr{I}}|\boldsymbol{q}, \boldsymbol{p})$ (see Eq. (5)), where $\boldsymbol{q} = [\boldsymbol{x}_0 \hat{t} \hat{t}^{\perp}]^{\mathrm{T}}$, r is the scale variable in $\boldsymbol{p}$, and $\hat{t}_0$ is flipped if an opposite trace is desired.

2.5. Regularization of vessel axis

The vessel axis extracted with the probabilistic framework described in Section 2.1 may be prone to slight jitters due to the fact that the axis smoothness is constrained locally throughout the extraction, as shown in Fig. 3a. In order to have a globally smooth vessel axis, we minimize the magnitude of the first-order derivative along the axis with the following update equation:

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \gamma_{\text{tensile}} \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j^{(t)} - \boldsymbol{x}_i^{(t)}), \tag{9}$$

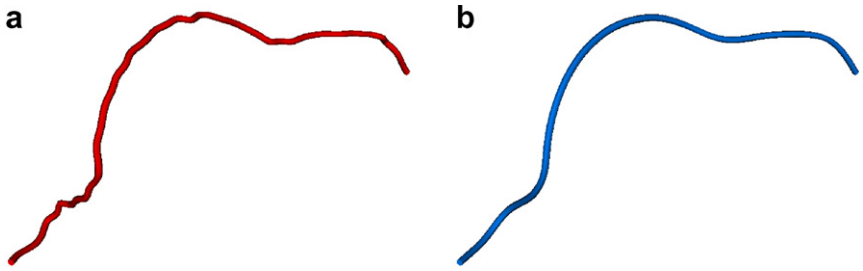

Fig. 3. (a) The vessel axis extracted with the probabilistic framework described in Section 2.1. Slight jitters are noticeable. This is due to the fact that the axis smoothness is constrained locally throughout the extraction. (b) The regularized vessel axis. Regularization is performed by minimizing the magnitude of the first-order derivative along the axis.

where $\boldsymbol{x}_i^{(t)}$ is the 3-D position of the i-th axial point at time t and $\mathcal{N}_i$ denotes the set of first-order neighbors of the i-th axial point along the trace. γ_{tensile} controls the rate of the minimization. This makes the vessel axis act like a membrane such that slight jitters can be eliminated, as discussed in (Kass et al., 1988). We set $\gamma_{\text{tensile}} = 0.1$ and run the update equation for 10 times in all the experiments conducted. The reference frame of the regularized axial point $\boldsymbol{x}_i$ is then updated with the "double-cross" successive method (Bloomenthal et al., 1996) by taking the tangent of a cardinal spline (Kochanek and Bartels, 1984) (whose spline knots are the axial points of the smoothed vessel axis) as the z-axis. Fig. 3b illustrates a regularized axis.

3. Segmentation of vessel with stream surfaces and minimum cost paths

3.1. Stream surfaces as the cross-sections

Once a vessel axis is extracted in a 3-D space and regularized, we define a steady 3-D vector field on a regular grid that is identical (in terms of sample spacing and dimensions) to the original 3-D angiogram. The vectors are the gradient of the Euclidean distance transform (EDT) of the vessel axis. For better illustration, we show the EDT image together with the axis of a straight vessel segment in Fig. 4a, and several vectors from the vector field in Figs. 4b and c. With this dedicated steady vector field, we employ a commonly used technique in flow visualization, called stream surfaces (Schroeder et al., 2003), to construct the cross-sections.

A stream surface is defined as the locus of an infinite set of streamlines that are curves always tangential to a steady vector field and originated from a rake, a continuous line segment around the axial point, as illustrated in Fig. 5a. According to this definition, our cross-sections have the property that their normals are everywhere tangential to the vectors in the field. On one hand if the vessel axis is straight, those surfaces form the cross-sectional planes, as shown in Fig. 5b, on the other hand if the vessel axis forms an arc, the surfaces are squeezed on the inner side of the curved axis (highlighted by the arrow in Fig. 5c) and they are no longer kept on flat planes.

This property guarantees no intersection amongst the cross-sections. Intersecting cross-sections are problematic if one wants to define vascular surface model from the contours extracted on the cross-sections that delineate the vessel boundary (Yim et al., 2001; Felkel et al., 2004). Fig. 6 illustrates the problem with a 2-D example, in which a part of the boundary of a curved vessel is shown together with its axis. Several cross-sectional planes (gray straight lines in the figure) are defined along the axis. It is observed that they are intersected on the RHS of the axis. The boundary on the cross-sectional planes are detected and shown as hollow dots in the figure. One can simply construct a non-self-intersecting surface to model the boundary by connecting the vertices of adjacent contours (the hollow dots), see the boundary on the LHS of the axis in Fig. 6a. The numbers enumerate the sequence of connection. As long as the *order of connection* is identical to that of the points on the axis (the solid dots), there is no self-intersection on the surface model. Nonetheless, this is not the case for the boundary modeling on the RHS of the axis (see Fig. 6b for a close-up of that region). A self-intersecting surface is, therefore, inevitable. By ensuring the cross-sections are free from intersections, the problem of surface self-intersecting can be avoided.

We adopt the stream surface construction algorithm proposed by Hultquist et al. (1992) to generate the cross-sections given the extracted axial points and their associated reference frame (defined by $\hat{t}_i$, axial tangent, and $\hat{t}_i^{\perp}$, its binormal vector, where i is the index of the axial point, see Fig. 1 for details).

The rake of a cross-section is constructed from K evenly distributed points sampled on the normal plane (defined by the reference frame) from the axial point at d_r mm (we set $K = 8$ and $d_r = \delta V$, where δV is the voxel size of the angiogram). A fourth-order Runge–Kutta ordinary differential equation (ODE) solver is employed to generate segments of streamlines (Press et al., 1992). Greedy minimal-width tiling strategy is used to tile the propagating *ribbon*, the surface between two adjacent streamlines, with triangles. This strategy selects the shortest one of the two diagonals formed by the four points of interest on the two neighboring streamlines to complete the tessellation. Fig. 7a illustrates the idea, the segment $\overline{L_iR_i}$ ($i \in \mathbb{Z}^*$) defines a portion of the propagation *front*.

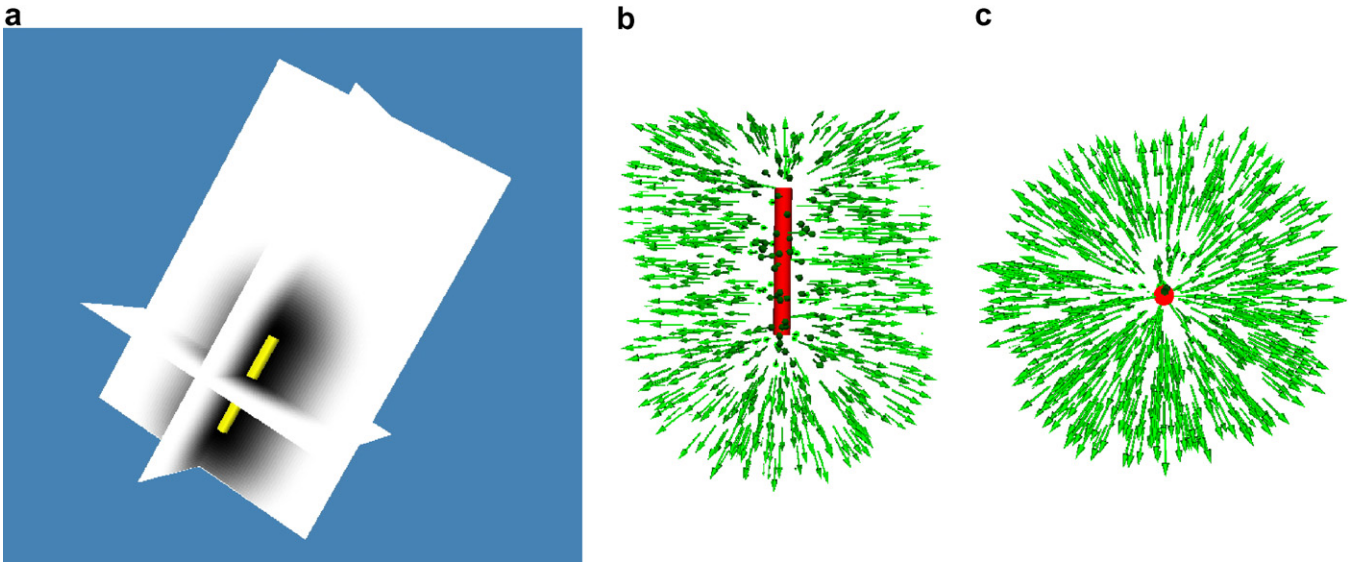

Fig. 4. (a) The Euclidean distance transform (EDT) image of the vessel axis. Three orthogonal planes that slice the EDT are shown. Bright (dark) pixels on the planes denote high (low) distance values. The vessel axis is illustrated as a tube at the crossing of two orthogonal planes. (b) A side view and (c) a top view of several vectors from the vector field which contains the gradient of the EDT. The vessel axis is shown as a tube in the middle of the field.

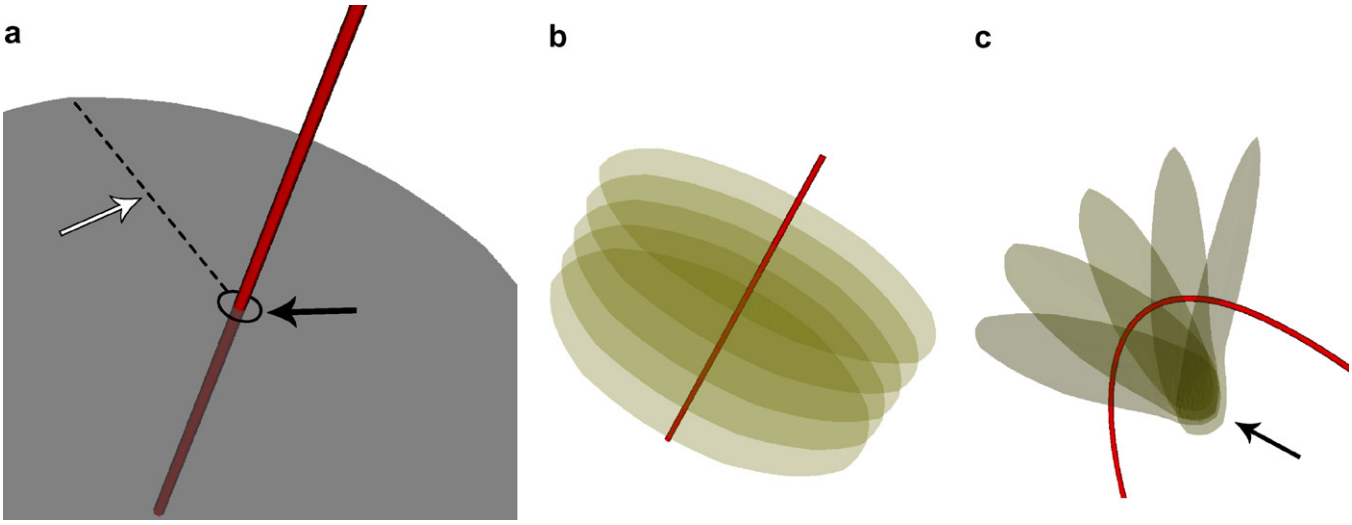

Fig. 5. (a) A stream surface. One of the streamlines is shown as a dotted line on the surface highlighted by the hollow arrow. The rake (solid line), from which the streamline is originated, around an axial point is highlighted by the solid arrow. (b) Stream surfaces along a straight axis. They form the cross-sectional planes. (c) Stream surfaces along a curved axis. They are squeezed on the inner side of the curve (highlighted by the arrow).

Furthermore, as the front (originated from the rake) advances, we add a new streamline when the width of the ribbon is greater than the step length of the streamline by a user-defined factor (we set the step length and the factor to δV and 1.5, respectively, in this paper), in order to increase the sampling density at the distant portion of the surface from its center. This is to alleviate the sparse sampling at the distant region, due to the fact that a fixed number of streamlines are radiated out from the axial point. We label each streamline with a level index l. Each time when a streamline is added, the level index of the new streamline is set to max $(l_L, l_R) + 1$, where l_L and l_R are the level indices of the two adjacent streamlines. The level index of the streamlines that originate from the rake is set to 0. Figs. 8a and b illustrate a stream surface with and without adding of new streamlines, respectively. The origins of the streamlines at different levels are highlighted in Fig. 8c which shows a close-up of the square region in Fig. 8a.

In the meantime, we also keep track of the distance d_s to the axial point of each streamline sample. It is to help define the smoothness constraint in finding the round contour on the cross-sections that delineates the vessel boundary (cf. Eq. (10) on Section 3.2). The value of d_s^k is the cumulative length of the streamline segment from the origin of the streamline to the sample k. If a new streamline is introduced with origin M, as illustrated in Fig. 7b, we

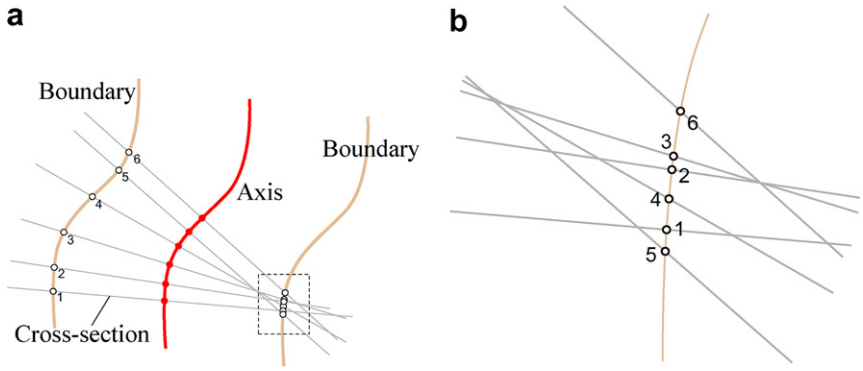

Fig. 6. One cannot simply construct a non-self-intersecting surface to model the boundary on the RHS of the axis by connecting the vertices of adjacent contours (the hollow dots) due to the intersecting cross-sections (gray straight lines). Note the order of connection (numbered starting from the cross-section at the bottom of the figure), (a) since it is identical to that of the axial points for the boundary on the LHS of the axis, there is no such problem; however it is not the case on the RHS, as shown in (b). Self-intersecting surface is therefore inevitable if those vertices are connected accordingly to the ordering.

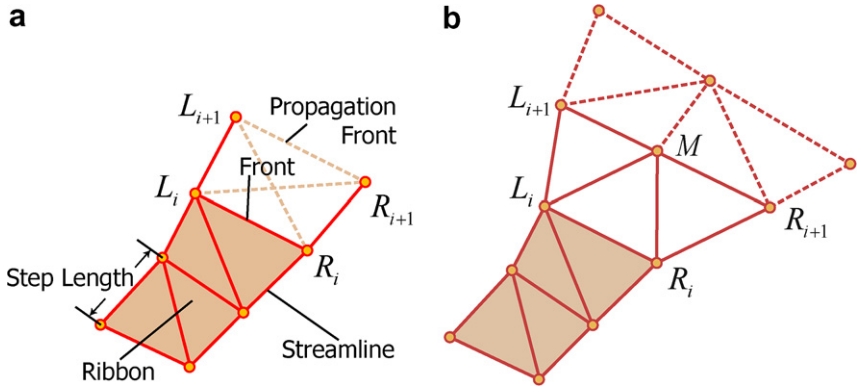

Fig. 7. Front propagation. (a) The four points of interest on two neighboring streamlines. (b) A new streamline is introduced with origin M.

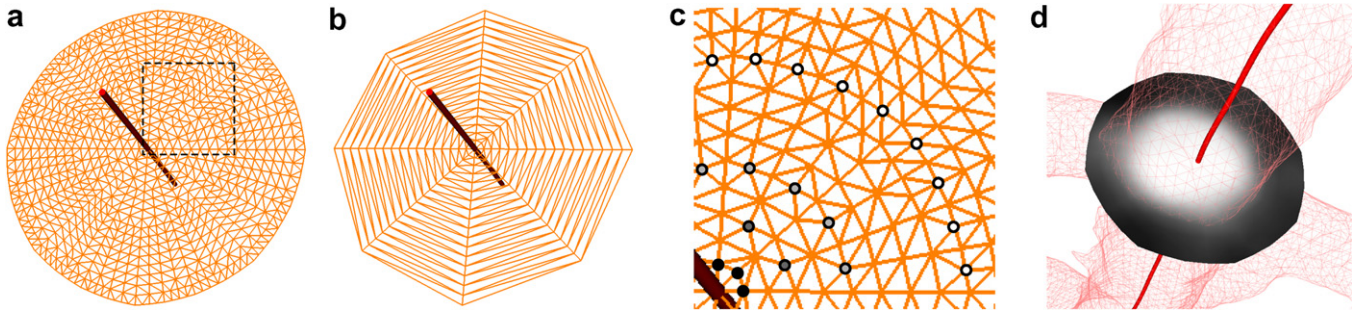

Fig. 8. Stream surface (a) with and (b) without adding of new streamlines. (c) A close-up of the highlighted region in (a). The origin of the streamlines of different levels are highlighted by different dots: black, level index $l = 0$; dark gray, $l = 1$; light gray, $l = 2$; white, $l = 3$. (d) A stream surface color-coded with linearly interpolated gray-scale values from the original angiogram at each vertices together with an iso-surface model of the vasculature and the corresponding vessel axis.

set d_s^M to $(d_s^{L_{i+1}} + d_s^{R_{i+1}})/2$, where $d_s^{L_{i+1}} = d_s^{L_i} + \delta V$ and $d_s^{R_{i+1}} = d_s^{R_i} + \delta V$.

The front stops advancing until it reaches a pre-defined maximum number of steps. To determine how far the front of the stream surface should propagate such that it can cover the vessel boundary, we employ the scale variable r_i in the solution vector $\boldsymbol{p}_i$. As described in Section 2.1, the scale r_i is proportional to the vessel width at axial point $\boldsymbol{x}_i$. Since we use the BSP image generated at the axis of a solid straight tube as the appearance model $\hat{\mathcal{I}}$ (cf. Section 2.2, Figs. 2a and e) which is constructed with the length of the projection bound equal to 1.5× the tube radius, the scale r_i in the vector $\boldsymbol{p}_i$ is therefore a good estimation of the local vessel width. In this work, we set the maximum

number of steps for the stream surface front propagation to $\lceil r_i/\delta V \rceil$ and thus the size of the stream surface constructed is adaptive to the width of the vessel.

3.2. Segmentation with minimum cost paths on stream surfaces

After the cross-sections are generated, the next step is to delineate the vessel boundary on those surfaces. Fig. 8d shows a cross-section color-coded with linearly interpolated gray-scale values from the original angiogram at each vertices together with an iso-surface model of the vasculature and the corresponding vessel axis for better illustration. In general, the vessel boundary to be outlined is round on the cross-section. Therefore a *circular* (closed and round) contour drawn on the cross-section around the axial point is sufficient for the boundary delineation.

We follow a research line (Sun and Pallottino, 2003) in which a shortest path algorithm is employed to extract circular object boundaries from 2-D images (regular grids). In that work, Sun and Pallottino applied the Bellman–Ford algorithm (Cormen et al., 1990) to find the minimum cost path (MCP) in a directed acyclic graph (DAG) defined on the elliptical polar-transformed image. The MCP corresponds to the closed and round boundary of the object to be segmented. (Readers are advised to read (Sun and Pallottino, 2003) for details.) In our application, however, the medium from which a round boundary to be extracted is not a regular grid. In fact, our cross-section (stream surface) is an irregular one. As such, we have to find an alternative way to define a DAG and its edge weights.

Fig. 9a illustrates the new way that we define the DAG amongst two adjacent streamlines and an added streamline in between. In the figure, only a small portion of the cross-section is shown. Curves S_α and S_β denote the streamlines of interest with level index equal to i. Curve S_γ is the streamline added in between the curves S_α and S_β. Its level index is therefore equal to $i+1$. The samples B_α and B_β define a segment of the front at which S_γ is inserted. We group locally the vertices into five sets, $V_1, \ldots, V_5$, as shown. It is noted that vertices B_α and B_β are shared by two pairs of sets $\{V_1, V_2\}$ and $\{V_4, V_5\}$, respectively. DAG edges are then defined as the edges of the complete bipartite directed (in anti-clockwise direction) graphs composed of three pairs of sets $\{V_1, V_4\}$, $\{V_2, V_3\}$ and $\{V_3, V_5\}$, except the edge from B_α to B_β. In doing so, we define a DAG over all the streamlines of the cross-section in anti-clockwise direction, in which a reference streamline is duplicated deliberately, as given in Fig. 9b. This allows us to find closed and round contours that delineate the vessel boundary on the stream surfaces from the MCPs on the DAG. Our DAG definition fulfills the requirements of the graph suggested in (Sun and Pallottino, 2003): (a) *stable* and *layered*, edges connect different vertex sets (layers) but not the same; and (b) *sequential*, edges connect two adjacent vertex sets (layers).

The weight of the DAG edge from L to R, where L and R are two adjacent vertices of the DAG, is defined as follows:

$$w_e(L,R) = \frac{1}{2^\alpha}\left(K_1 \overbrace{(1-\overline{M}(L,R))}^{\text{image constraint}} + K_2 \overbrace{\left(1-\exp\left(-\frac{\delta d^2(L,R)}{2\sigma_d^2}\right)\right)}^{\text{regularization constraint}} \right), \tag{10}$$

where α is the largest level index l of the streamlines associated with L and R (see Fig. 8c)—for example, as given in Fig. 9a, if L and R are in sets V_2 and V_3, $\alpha = i+1$; if the two samples are in sets V_1 and V_4, $\alpha = i$—such quantity counteracts the increase in the radial resolution, such that the MCP would not bias towards boundary that is closer to the axial point. K_1 (=1) and K_2 (=1) are constants to control the influence of the image and regularization constraints, σ_d (=0.5) controls the sensitivity of the function $\delta d(L, R)$ which denotes the difference of the distance to axial point from L and R. $\overline{M}(L,R)$ returns the average

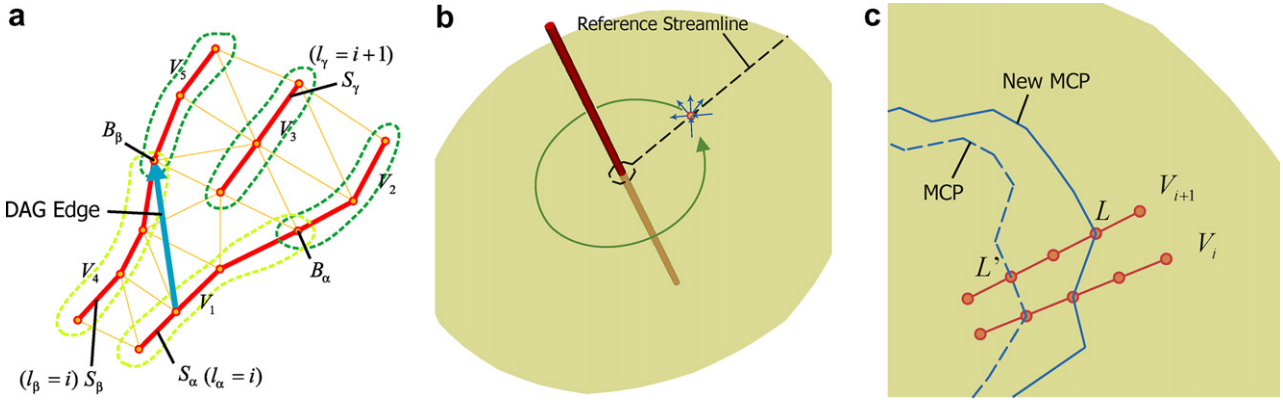

Fig. 9. (a) Definition of the directed acyclic graph (DAG) amongst two adjacent streamlines and an added streamline in between (only one DAG edge is shown for better clarity) (b) The DAG is defined over all the streamlines of the cross-section in anti-clockwise direction, in which the reference streamline is duplicated deliberately. (c) The user places anchor point at sample L and forces the minimum cost path (MCP) to pass through sample L on the cross-section, rather than other sample L' on the same streamline.

gradient magnitude at samples L and R. Those functions are expressed as follows:

$$\delta d(L,R) = |d_s^L - d_s^R| \tag{11}$$

and

$$\overline{M}(L,R) = \frac{|\dot{\nabla}\mathscr{A}(L)| + |\dot{\nabla}\mathscr{A}(R)|}{2}, \tag{12}$$

where $\mathscr{A}$ is the angiogram and $\dot{\nabla}$ denotes the normalized gradient operator. It is noted that the multiplicands of the constants K_1 and K_2 are $\in[0, 1]$. This makes the magnitudes of the two constraints comparable.

Under some circumstances, the user may want to guide the boundary extraction by given anchor point(s). Our formulation can easily satisfy such requirement by offsetting some of the edge weights. Suppose the user wants the boundary to pass through sample L on the cross-section, rather than other sample L' on the same streamline, as illustrated in Fig. 9c. In order to guarantee that the MCP passes through L via a DAG edge from vertex set (layer) V_i to V_{i+1}, we can simply subtract the in-edges' weights of vertex L by the maximum value of all edge weights.

This slight modification can also cope with the scenario that multiple samples on the same streamline are selected as anchor points. Because of the stable and layered properties of the DAG, the MCP obtained should only pass through one of those anchor points on the same streamline. Therefore, such fuzzy guidance is not a problem in our algorithm.

In this work, we employ an algorithm dedicated to solving the single-source shortest-paths problem on a weighted DAG. This algorithm exploits the property of DAG, i.e., no negative-weight cycles can exist even if there are negative edges, to compute the shortest paths from a single source in $\Theta(|\mathscr{V}| + |\mathscr{E}|)$ with a topological sort, where $\mathscr{V}$ and $\mathscr{E}$ are the set of vertices and edges of the DAG (Cormen et al., 1990). Its running time complexity is smaller than the Dijkstra algorithm, $\Theta(|\mathscr{V}| \lg |\mathscr{V}| + |\mathscr{E}|)$, and the Bellman–Ford algorithm, $\Theta(|\mathscr{V}| \times |\mathscr{E}|)$.

Algorithm 1. Closed and Round Boundary Extraction

1. $c^* \leftarrow \infty$
2. $p^* \leftarrow$ NULL
3. **for all** samples v_i in the reference streamline **do**
4. $P \leftarrow$ DAG-SHORTEST-PATHS (G, W, v_i)
5. $(p, c) \leftarrow$ BACK-TRACK (P, v_i^*)
6. **if** $c^* > c$ **then**
7. $c^* \leftarrow c$
8. $p^* \leftarrow p$
9. **end if**
10. **end for**
11. Return p^*

To summarize, the closed and round boundary extraction algorithm is outlined in Algorithm 1. The vertex v_i denotes a sample on the reference streamline. Given a DAG G, a set of edge weights W and a source v_i, DAG-SHORTEST-PATHS(·) returns a predecessor map P of the MCP. BACK-TRACK(·) returns the path p, which ended at v_i^* given the predecessor map P, and its cost c, where v_i^* is the duplicate of v_i (the corresponding vertex on the duplicated reference streamline). p^* stores the MCP found so far regardless the source. c^* keeps the corresponding cost.

Fig. 10 shows the vessel boundaries extracted from several typical cross-sections: (a) without any anchor point; (b) with anchor points, one of the vessel boundaries in a kissing vessel region; and (c) with multiple anchor points on a streamline, the boundary of a vessel branch at a bifurcation.

3.3. Formation of 3-D vessel boundary surface model

The outputs of Algorithm 1 are contours (on non-intersecting cross-sections) in a 3-D space which delineate the boundary of a vessel of interest along its axis. A 3-D vessel boundary surface model can then be constructed from the extracted contours by joining the vertices of adjacent ones (cf. Section 3.1 and Fig. 6 for the explanation). However, due to the difference in the number of vertices on each extracted contour, there is no obvious way to join the vertices. The algorithm proposed by Keppel (1975) offers a remedy for this problem.

As long as a reference vertex on each of the adjacent contours is given, the Keppel algorithm can find the optimal triangulation to approximate the surface of a volume defined by the adjacent contours. The optimality, in this work, is defined as the total length of the edges inserted in between those two contours. In this work, we use the contour vertex that lays on the reference streamline of each cross-section as the reference vertex in all the experiments conducted.

3.4. Constrained deformation on cross-sections

The 3-D vessel boundary surface model constructed so far can only serve as a rough delineation of the vessel of interest—mesh vertices are discrete samples obtained from each cross-section, which in fact is posing discontinuity along the longitudinal direction on the surface model. Owing to this, fine position adjustment on the mesh vertices is necessary. We employ a 3-D version of the active contour models proposed by Kass et al. (1988) to perform such model deformation. The only energy functionals to be minimized in this work are the internal energies and the edge functional. The internal energies (tensile and flexural) provide surface regularization, making the surface model act like a membrane and a thin-plate to resist the development of discontinuity and corner (singularity), respectively. The edge functional, $E_{\text{edge}} = -|\nabla\mathscr{A}|^2$, on the other hand, attracts the surface moves towards the object edges (the vessel boundary). The weights to control the influence of tensile, flexural, edge functionals and the time step are set to 0.5, 0.5, 2, and 1/50, respectively. The deformation halts until 100 iterations are executed or the net displacement of

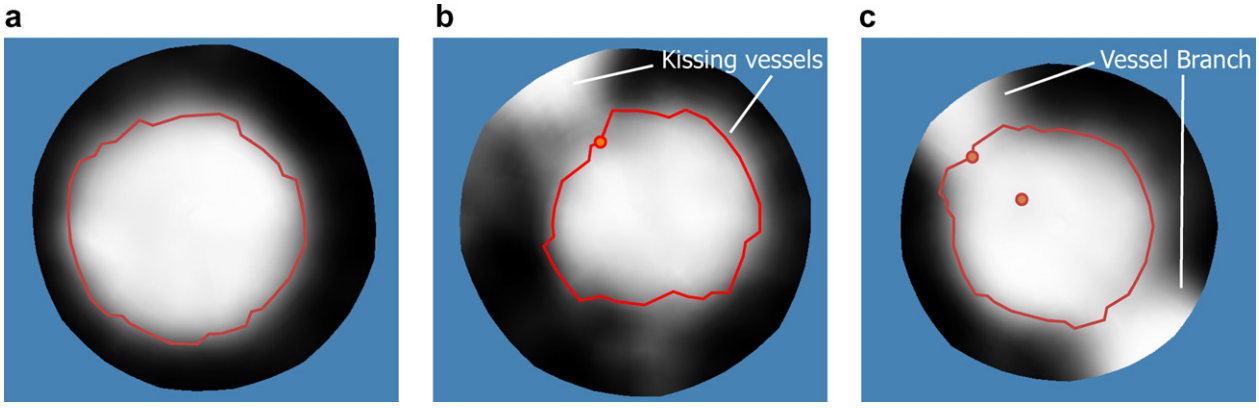

Fig. 10. Vessel boundaries (lines on the surface) extracted from several typical cross-sections. (a) Without any anchor point. (b) With an anchor point (dot on the surface), one of the vessel boundaries in a kissing vessel region. (c) With multiple anchor points on a streamline, the boundary of a vessel branch at a bifurcation.

each vertex is <1/10 of the voxel size. All those values of the parameters are found empirically and are fixed in all the experiments conducted.

Nonetheless, if the deformation is performed in a free-form fashion, two undesirable effects are resultant, as pointed out by Lobregt and Viergever (1995): (1) the tensile and flexural functionals make the deformable model shrink and (2) the edge functional drives the mesh vertices to bunch together at regions that with high intensity gradient magnitude (e.g. corners). As shown in Fig. 11a, the model obtained from free-form deformation is shrunk and contracted severely at both ends of the model, which leads to an incomplete modeling of the vessel of interest.

A practical solution to solve such problem, as discussed in (Lobregt and Viergever, 1995; Yim et al., 2001), is to insist radial vertex displacement throughout the deformation. The streamlines generated in constructing the cross-sections provide excellent loci for the mesh vertices to following. Fig. 12 shows three illustrations to demonstrate step-by-step on how the constrained displacement is accomplished. $\boldsymbol{v}_i$, $\boldsymbol{v}_{i+1}$ and $\boldsymbol{v}_{i+2}$ are three consecutive samples on a streamline. $\boldsymbol{u}$ is the mesh vertex of interest, $\boldsymbol{f}$ is the deformation driving force acting on it. T_0 and T_1 are the cursors pointing to the start and end points of the line segment that $\boldsymbol{u}$ lays. $\hat{a}$ is the unit vector of $\overrightarrow{T_0T_1}$ and d equals $|\overrightarrow{T_0T_1}|$. $\boldsymbol{u}'$ and $\boldsymbol{f}'$ denote the new quantity of $\boldsymbol{u}$ and $\boldsymbol{f}$, respectively. The steps are as follows,

Step 1: Project $\boldsymbol{f}$ along the line $\boldsymbol{v}_i + n\hat{a}$ ($n \in \mathbb{R}$, and $\boldsymbol{v}_i$ is the point at cursor T_0), as given in Fig. 12a.

Step 2: Advance $\boldsymbol{u}$ according to the projection before it reaches the other end point at cursor T_1 (i.e., $\boldsymbol{v}_{i+1}$). Then we obtain $\boldsymbol{u}'$ as given in Fig. 12b.

Step 3: Update $\boldsymbol{f}$ with the residual driving force and advance the cursors to the adjacent segment, as given in Fig. 12c.

Step 4: Continue the above process until the displacement, $|\boldsymbol{u}' - \boldsymbol{u}|$, is very small.

This illustration describes the scenario that a mesh vertex moves down the streamline (away from the streamline origin). The opposite scenario—vertex moves towards the

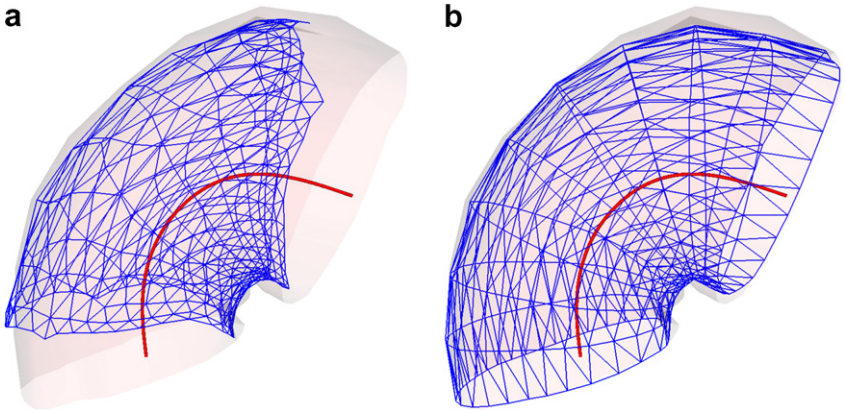

Fig. 11. Deformation of the vessel boundary surface model. Model before the deformation is shown in semi-transparent surface. Deformed models are illustrated in wire-frame. Vessel axis is shown for reference. (a) Free-form deformation, the model is shrunk and contracted severely at both ends, which leads to an incomplete modeling of the vessel of interest. (b) Constrained deformation along the streamlines on the cross-sections.

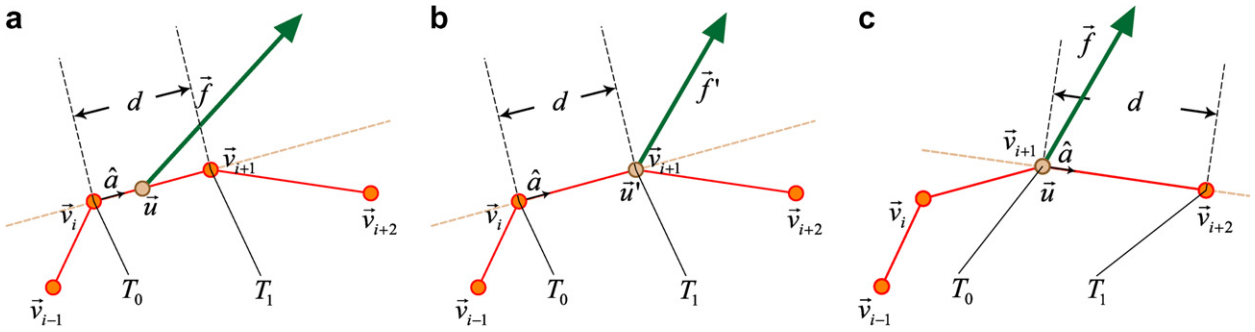

Fig. 12. Three illustrations to demonstrate step-by-step on how the constrained displacement is accomplished: (a) Step 1; (b) Step 2; and (c) Step 3.

streamline origin—is handled in a similar fashion. However, we need to handle a special case that when the vertex is going to move off the streamline at the origin. If the streamline's level index is equal to 0 (i.e., its origin is on the rake), we prohibit any further displacement of the vertex. (This is identical to the case that when the vertex reaches the edge of the cross-section.) Otherwise, we have to remove the vertex from the mesh. Fig. 13 shows how the surface gap is tiled due to the vertex deletion. The closest immediate contour neighbor $\boldsymbol{u}_c$ of the vertex-to-be-deleted $\boldsymbol{u}$ is selected. Then, new edges originated from $\boldsymbol{u}_c$ are inserted to connect to the neighbors of $\boldsymbol{u}$, except edge $(\boldsymbol{u}, \boldsymbol{u}_c)$ and the duplicated edges. Such deletion of the off-streamline vertex has the advantage to controlling the mesh transverse resolution. A vessel boundary surface model deformed along the streamlines on the cross-sections is shown in Fig. 11b.

Algorithmically, the constrained deformation is given in Algorithm 2, where α is the interpolation coefficient of the line segment $\overline{T_0T_1}$ and α' is the new quantity of it. m is the vertex displacement along the streamline driven by $\boldsymbol{f}$ measuring in the unit of the interpolation coefficient. ϵ is a very small constant. Flags B_{moved} and $B_{\text{prev\_moved}}$ are marked if the vertex is moved in the current and the previous iteration, respectively, on the line segment $\overline{T_0T_1}$. Flag $B_{\text{terminate}}$ marks if it is going to terminate the repeat-until loop. Function Pos(·) returns the 3-D position of the given point, At-Streamline-Origin(·) and At-Stream-Surface-Edge(·) give a true value if the given point is at the streamline origin and stream surface edge, respectively. Prev-Segment(·) and Next-Segment(·) return the previous and next streamline segments of the given line segment.

Algorithm 2. Constrained Deformation along Streamline

1. $\epsilon \leftarrow 0.001$, $m \leftarrow \infty$
2. $B_{\text{moved}} \leftarrow$ False, $B_{\text{prev\_moved}} \leftarrow$ True
3. $B_{\text{terminate}} \leftarrow$ False
4. **repeat**
5. $\quad \alpha \leftarrow ((\boldsymbol{u} - \text{Pos}(T_0)) \cdot \hat{a})/d$
6. $\quad m \leftarrow (\hat{a} \cdot \boldsymbol{f})/d$
7. $\quad$ **if** $(\alpha = 0) \wedge (m \leqslant 0)$ **then**
8. $\quad\quad$ **if** At-Streamline-Origin (T_0) **then**
9. $\quad\quad\quad m \leftarrow 0$, $B_{\text{moved}} \leftarrow$ True
10. $\quad\quad$ **else**
11. $\quad\quad\quad (T_0, T_1) \leftarrow$ Prev-Segment$(\overline{T_0T_1})$
12. $\quad\quad\quad m \leftarrow 0$, $B_{\text{moved}} \leftarrow$ False‘
13. $\quad\quad$ **end if**
14. $\quad$ **else if** $(\alpha = 1) \wedge (m \geqslant 0)$ **then**
15. $\quad\quad$ **if** At-Stream-Surface-Edge (T_1) **then**
16. $\quad\quad\quad m \leftarrow 0$, $B_{\text{moved}} \leftarrow$ True
17. $\quad\quad$ **else**
18. $\quad\quad\quad (T_0, T_1) \leftarrow$ Next-Segment$(\overline{T_0T_1})$
19. $\quad\quad\quad m \leftarrow 0$, $B_{\text{moved}} \leftarrow$ False
20. $\quad\quad$ **end if**
21. $\quad$ **else**
22. $\quad\quad \alpha' \leftarrow \max(0, \min(1, \alpha + m))$
23. $\quad\quad m \leftarrow \alpha' - \alpha$
24. $\quad\quad \boldsymbol{u}' \leftarrow (1 - \alpha')\text{Pos}(T_0) + \alpha'\text{Pos}(T_1)$
25. $\quad\quad \boldsymbol{f}' \leftarrow (\boldsymbol{f} - (\boldsymbol{u}' - \boldsymbol{u}))$
26. $\quad\quad \boldsymbol{u} \leftarrow \boldsymbol{u}'$, $\boldsymbol{f} \leftarrow \boldsymbol{f}'$, $B_{\text{moved}} \leftarrow$ True
27. $\quad$ **end if**
28. $\quad B_{\text{terminate}} \leftarrow \neg B_{\text{moved}} \wedge \neg B_{\text{prev\_moved}}$
29. $\quad B_{\text{prev\_moved}} \leftarrow B_{\text{moved}}$
30. **until** $B_{\text{terminate}} \vee (m < \epsilon)$

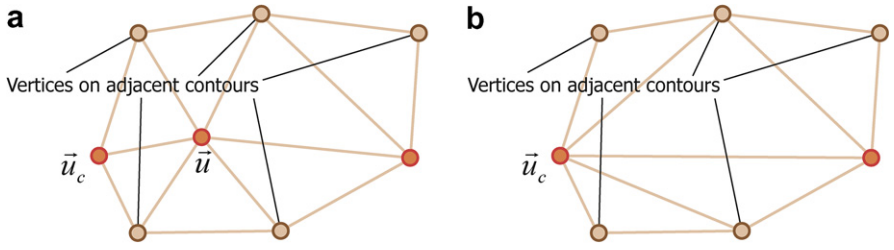

Fig. 13. Vessel boundary surface model re-tessellation due to vertex deletion: (a) before deletion and (b) after deletion.

In the algorithm outlined above, there are three scenarios embedded in the first level if-block. The first two scenarios state that when the mesh vertex $\boldsymbol{u}$ is going to move off the line segment $\overline{T_0T_1}$ at point T_0 or T_1, then the streamline segment of interest is changed to the adjacent one accordingly. The third scenario states that the new quantity of the driving force is obtained by subtracting the vertex displacement vector from the old driving force, $\boldsymbol{f}$, when the vertex moves within the line segment.

3.5. Voxelization of deformable model

As a final step, the deformable model is voxelized into a binary volume with voxel size and dimensions identical to the original angiogram. It has an advantage to aggregating different segmentation results with a simple union operation, for example, merging the segmentations of two vessel branches of a bifurcation to compose the segmentation of the vascular junction, and producing the segmentation of an aneurysmal lumen by combining the binary volumes of the aneurysmal sac and its parent vessels.

4. Results

4.1. Vessel axis

We apply a modified version of (1 + 1)-Evolution Strategy (ES) for the optimization (modification is referred to the optimization with simple bounds on the parameters). (1 + 1)-ES is a special type of the Evolutionary Algorithms with both the population size and the number of children generated equal to one, which has an automatic step size and provides search direction adaptation (Styner et al., 2000). Thus, it has the ability to step out of non-optimal minima and can provide the axis tracing problem with a long capture range. We have tested the proposed method on three synthetic data sets, a clinical data set—3-D rotational angiography (RA), available at http://www.gris.uni-tuebingen.de/areas/scivis/volren/datasets/datasets.html, courtesy of Philips Research, Hamburg, Germany. The size of the image volume is $256 \times 256 \times 256$ vox. with isotropic voxel in 0.33 mm—and three sub-volumes of 3-D RA acquired at the Department of Diagnostic Radiology and Organ Imaging, Prince of Wales Hospital (PWH), Hong Kong (see Section 4.3 for the description of the data sets). The design of the synthetic data took several challenging vascular structures for the trace-based axis extraction methods into consideration: kissing vessels, a bifurcation, and a typical saccular aneurysm in the cerebral circulation.

Figs. 14a–c and 15 show the experimental results obtained from our vessel axis tracer on the synthetic and clinical data sets, respectively. It is worth mentioning that the regularization procedure outlined in Section 2.5 is not performed to obtain the experimental results presented in this sub-section. This is to make fair comparisons with other algorithms since the regularization procedure is not included in the original algorithms. The dark axes in those figures are traced without any user guidance. It is evident that the vessel axes extracted from the kissing vessels (see Figs. 14a and 15a), at the furcations (see Figs. 14b and 15b), and from the thin vessel (<2 voxels in diameter, see Fig. 15c) are satisfactory. Due to the existence of the synthetic saccular aneurysm (see Fig. 14c), the axis extracted are steered, as a result it is deviated from the desired vessel axis. User guidance is therefore needed to drive the trace towards the target axis in this case, as well as to the side branches of the bifurcations shown in Figs. 14b and c. The necessity of such guidance is determined visually by the user. In the current implementation, iso-surface of the vasculature is shown together with the traced axis (MIP image can also serve the purpose). No guidance is required until the user found that the extracted axis is deviating from the one expected. The user-drawn 2-D curves (green in color with circles, those circles are the 2-D locations picked by the user on the screen to compose the curves) are depicted in Figs. 14d, h and l.[1] The curve shown in Fig. 14d helps guide the trace (lighter in color) to go to the right side branch of the bifurcation given in Fig. 14b. The other two curves given in Figs. 14h and l altogether steer the trace (lighter in color) to the left side branch of the saccular aneurysm shown in Fig. 14c. Two extra 2-D curves are needed to produce the trace to the right side branch, however, they are not shown here.

As a comparison, we have also tested the methods presented in (Aylward and Bullitt, 2002; Shim et al., 2005) on the data sets. Figs. 14e–g and i–k show the results obtained from Aylward and Bullitt's (2002) and Shim et al.'s (2005) methods on the synthetic data sets, respectively. The shorter dark axes shown in Figs. 14e–g are the vessel axes extracted with the large spatial discontinuity detection (LSDD). Such detection mechanism preempts the tracing if the next axial points is >1 voxel apart from the current one. It is observed that, with the detection, Aylward and Bullitt's method stops the tracing just before entering the kissing, bifurcation and diseased regions. The method, however, produces axes (in lighter color) with a large axial point displacement in the aforementioned regions if there is no LSDD. Shim et al. method gives similar results except for the kissing vessels. In addition, several U-turns can be observed at the bifurcation and in the aneurysmal sac. Comparisons on the clinical data set (from Philips Research) are presented in Fig. 15, in which the green and white axes are the vessel axes obtained from Aylward and Bullitt's, and Shim et al. methods, respectively. In the regions that contain kissing vessels, a furcation and a thin vessel, the axes extracted from the two methods are either jittered in between the two adjacent vessels, dragged towards the center of the furcation or off the target axis when the vessel is too thin to be traced.

[1] For interpretation of the references to color in this figure, the reader is referred to the web version of this article.

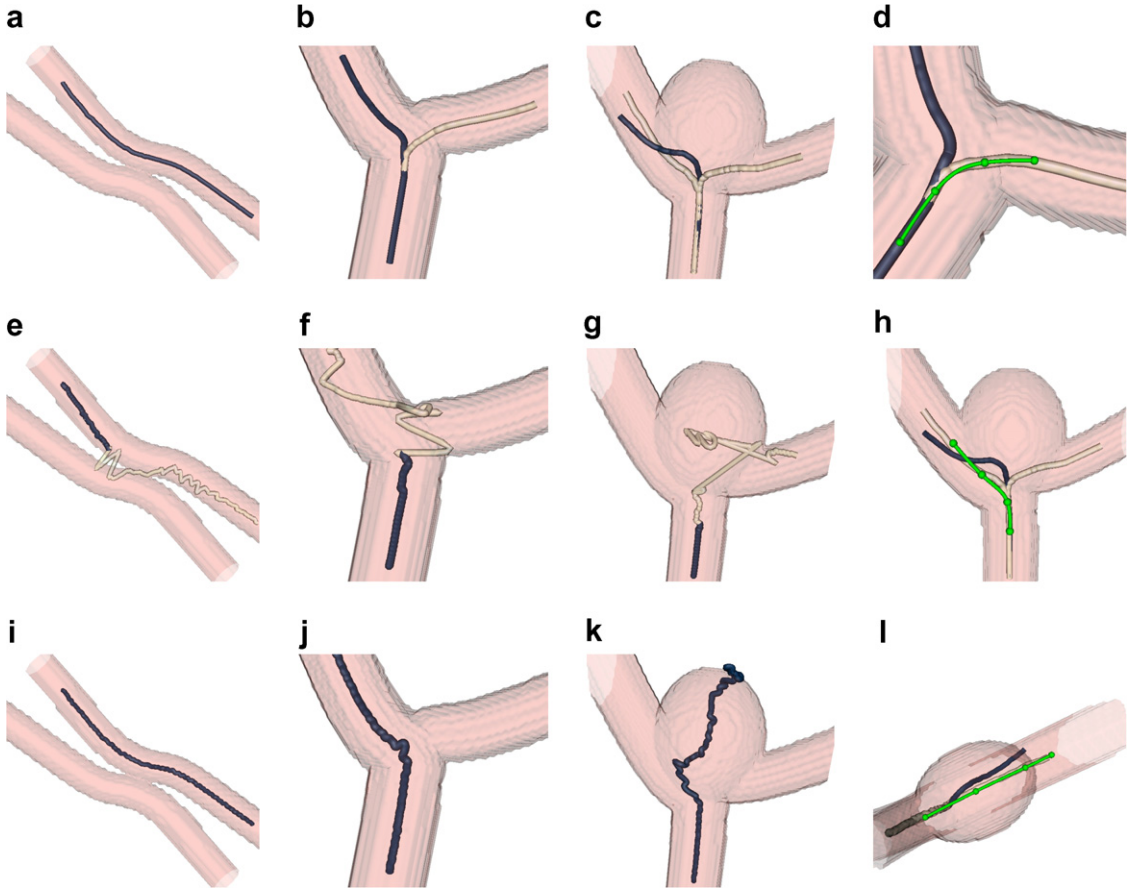

Fig. 14. Results obtained from (a–c) our novel vessel axis tracer, the methods proposed by (e–g) Aylward and Bullitt (2002) and (i–k) Shim et al. (2005) on the synthetic data sets which contain (a, e and i) kissing vessels, (b, f and j) a bifurcation and (c, g and k) a saccular aneurysm. The 2-D curves drawn on the screen for the user guidance on the tracing are given in (d), (h) and (l). The semi-transparent surfaces are iso-surfaces and are displayed for better visualization only.

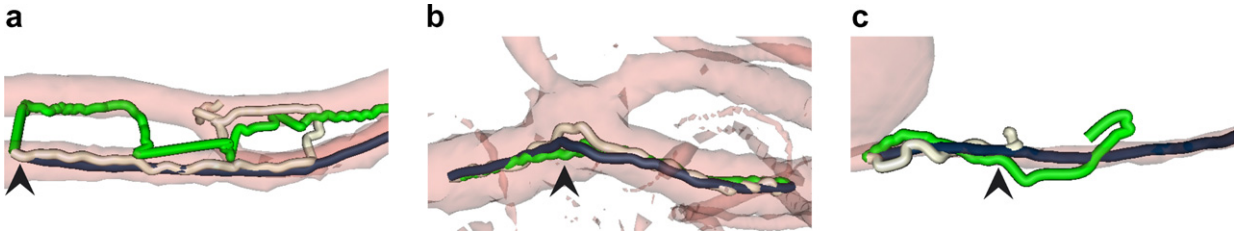

Fig. 15. Results on the clinical data set from Philips Research. The blue, green and white axes are the vessel axes obtained from ours, Aylward and Bullitt's (2002) and Shim et al.'s (2005) methods, respectively, on different vascular regions: (a) the two anterior cerebral arteries (ACA), (b) the first furcation of middle cerebral artery (MCA) and (c) opthalmic artery (OA). The semi-transparent surfaces are iso-surfaces and are displayed for better visualization only.

Generally speaking, their methods produce vessel axis with more jitters than ours.

Furthermore, we have extracted the axes of nine major vessels, including the internal carotid artery (ICA), the two anterior cerebral arteries (ACA), a few branches of the middle cerebral artery (MCA) and the posterior communicating artery (PCoA), in the Philips Research's clinical data set with our method. They are depicted in Fig. 16a. An MIP image is shown in Fig. 16b for visual comparison. User guidance is given at the furcations to steer the traces towards the side branches and at the base of the aneurysm to bypass the diseased portion (highlighted by arrows). It is demonstrated that our method can trace vessel segments, including the side branches and that in the diseased region. Since the other algorithms do not support user guidance, they may have problems in producing continuous traces through bifurcations and diseased lumens (as illustrated in Figs. 14 and 15). Hence they are not used to extract the axes of the major vessels, except the ICA (results are tabulated in Table 1) which is the first

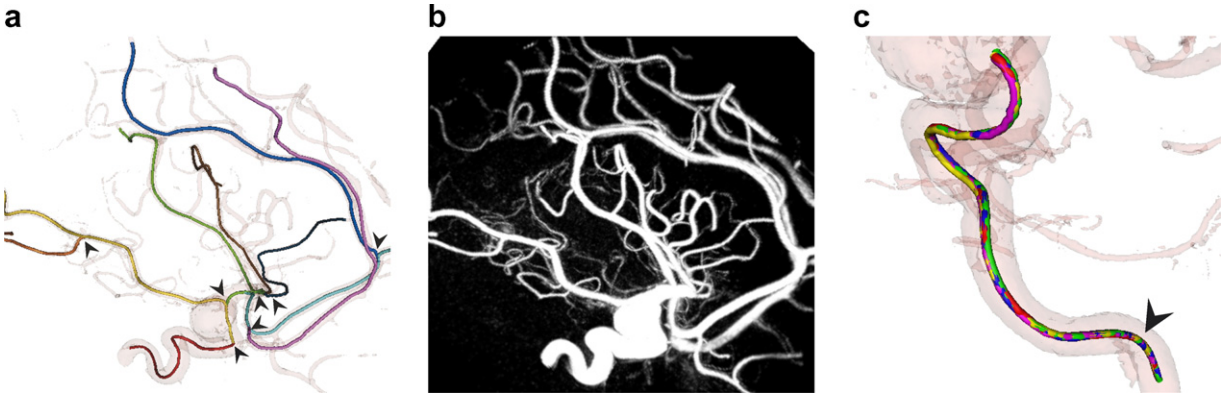

Fig. 16. (a) Extracted axes of nine major vessels in the clinical data set, including the ICA, the two ACA, a few branches of the MCA and the PCoA. (b) MIP image. (c) Axis of the ICA extracted at five difference times. The semi-transparent surfaces are iso-surfaces and are displayed for better visualization only.

Table 1
The Euclidean distance errors in percentage of diagonal length of the voxel between the extracted vessel axes and the manually drawn axes

Data Sets	Ours	Aylward and Bullitt (2002) with LSDD	Aylward and Bullitt (2002)	Shim et al. (2005)
Synthetic	29% (126%)	16% (140%)	464% (1274%)	304% (798%)
Clinical Sub-regions	23% (60%)	10% (53%)	96% (182%)	70% (228%)
Clinical Major vessels	29% (100%)	–	–	–
Clinical ICA	32% (77%)	3% (76%)	46% (137%)	6% (86%)

The mean errors are tabulated with the maximum errors in the parentheses. The synthetic data sets are those shown in Fig. 14. The clinical data set of sub-regions, major vessels and ICA are those presented in Figs. 15, 16a and c, respectively.

vessel segment (with no bifurcation and diseased lumen) located at the root of the vascular tree in the data set. To study the repeatability of the method, the ICA is traced for five times. The axes extracted are shown in Fig. 16c. Visually, there is no significant discrepancy amongst those axes.

We have also evaluated quantitatively the extracted vessel axes with the manually drawn axes built with reference to the reformatted cross-sectional images (defined on flat planes) along the vessels. The mean Euclidean distance (ED) errors (maximum errors are given in the parentheses) in percentage of diagonal length of the voxel (ΔV) are listed in Table 1. The error is defined as the shortest ED from an axial point of the extracted axis to the manually drawn axis. Apparently Aylward and Bullitt's method with LSDD performs better according to the errors tabulated. However, it produces preempted traces whenever large spatial discontinuity is encountered (the locations are highlighted by arrows in Figs. 15 and 16c). Compared with the other algorithms which produce axes with large errors in the synthetic and several sub-regions of the clinical data set, our method is capable of extracting the complete axes of the vessel segments consistently with mean errors $<\Delta V/3$ and maximum errors $\approx\Delta V$. Concerning the processing time, the axes of the nine major vessels (total length = 585 mm) took <15 min (computational time is ≈10 min, time for user intervention is ≈4 min) to extract with the novel method on a 2.0 GHz PC. The manual process, by contrast, took 169 min to finish. (To recapitulate, the image volume is $256 \times 256 \times 256$ voxels.) In a typical application where the region of interest is at a diseased portion or a furcation, the length of the vessel segment to be traced is ≈30 mm and our algorithm takes ≈30 s to extract (note that the current implementation of the algorithm is not optimized).

Several vessels of interest are traced in the three sub-volumes of the 3-D RA acquired at PWH. Fig. 17 shows the extracted vessel axes together with iso-surfaces of the volumes for the visualization of the vascular boundaries. It is observed that our method permits the user to trace vessel axes at furcations (two- or even three-way vascular junction) and diseased portion (stenotic lumen where vessel width changes suddenly). Later in Section 4.3, these axes help the generation of the stream surfaces which are then used to produce the vascular segmentation presented.

4.2. Cross-sections

We have compared the cross-sections generated by our method with two different approaches. The first one is the

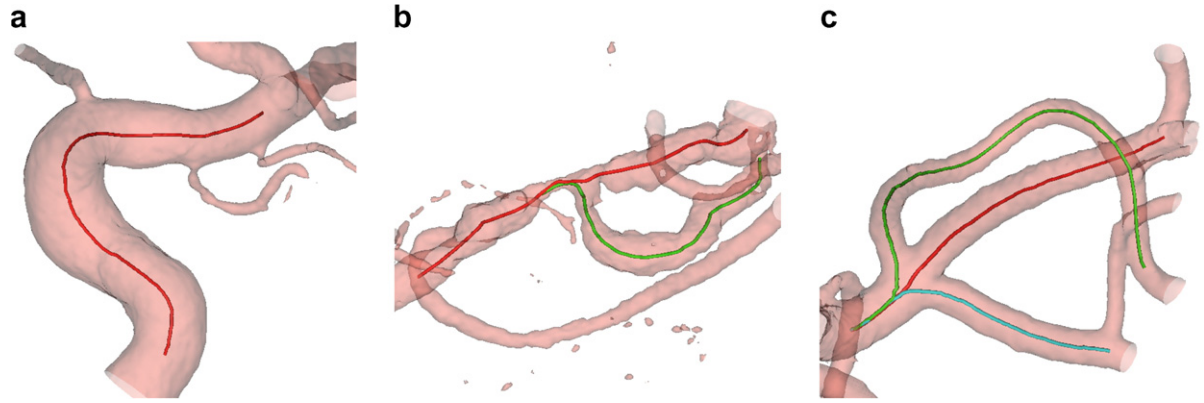

Fig. 17. The vessel axes extracted from the three sub-volumes of the 3-D rotational angiographies (RA) acquired at the Department of Diagnostic Radiology and Organ Imaging, Prince of Wales Hospital, Hong Kong. See Section 4.3 for the description of the data sets. The semi-transparent surfaces are iso-surfaces and are displayed for better visualization only: (a) Single vessel lumen; (b) stenotic lumen, bifurcation; and (c) 3-way junction.

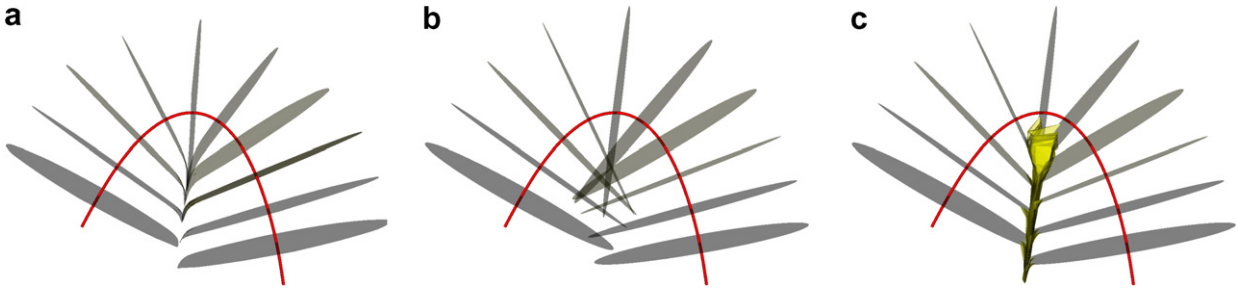

Fig. 18. The cross-sections generated with the three methods: (a) our method, (b) naïve method and (c) Yim et al. method (Yim et al., 2001).

commonly used naïve method (Hernández-Hoyos et al., 2000; Wink et al., 2000; Aylward and Bullitt, 2002) which gives flat cross-sectional planes that are normal to the tangents of the vessel axis. The second is the one proposed by Yim et al. (2001), in which a fixed number of straight radial lines are emitted from the axial points to define cross-sections and if one radial line intersects[2] the other, such straight radial line is then truncated and is merged to an un-truncated radial line (or previously truncated but then concatenated to an un-truncated radial line) whose corresponding axial point is the nearest in the axial direction.

The comparison is conducted on a synthetic data set that contains a tube in arc-shape. The cross-sections generated with the three methods (our method, naïve method and Yim et al. method) are given in Figs. 18a–c, respectively. It is noticeable that the cross-sectional planes generated by the naïve method are intersecting with each others at the arc on the inner side. By warping the straight radial lines, Yim et al. method successfully prevents intersecting cross-sections. Indeed their surfaces are overlapping on the inner side of the arc. However, due to the fact that the straight radial lines are concatenated in the warping region, the surfaces constructed from those warped lines are composed of planes with different, and yet discrete, orientations, hence they have artifacts. The other disadvantage of their approach is that the sampling is relatively too dense near the axis, while it is too sparse at distant region (similar to our stream surface without adding new streamlines as shown in Fig. 8b). On the contrary, ours results show no intersecting cross-sections, surface artifacts and severe uneven sampling on the surfaces (see Fig. 8a).

The above observations are further quantified by studying the maximum positive (convex surface) and negative (concave surface) mean curvatures of each cross-sections along the axis. Fig. 19a shows a plot (mean curvatures vs. unit length along the axis) of the study. Lines above and below zero mean curvature depict the values of the maximum positive and negative mean curvatures, respectively. It is noted that mean curvature values (both positive and negative) are high, especially at the arc region, in the results obtained from Yim et al. method (the dashed lines). This implies that there are ripples (combination of convex and concave surface patches) on their surfaces, see Fig. 19b for an example. Our method,

[2] Two radial lines are considered as intersected if one enters the territory of the other. The territory is the region that a radial line is closer to its corresponding axial point than to that of any other radial line (Yim et al., 2001).

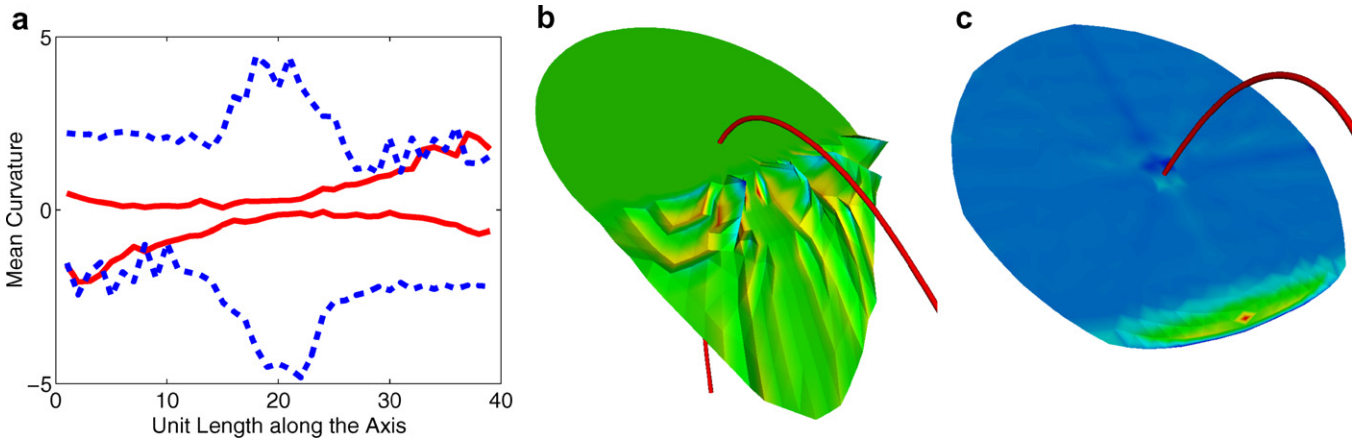

Fig. 19. (a) Plot of mean curvatures vs. unit length along the axis. Curvatures shown are the maximum positive (convex surface) and negative (concave surface) mean curvatures of each cross-section along the axis. Lines above and below zero mean curvature depict the values of the maximum positive and negative mean curvatures, respectively. Solid lines show the mean curvatures of our surfaces. Dashed lines show the mean curvatures of Yim et al. surfaces. (b) A cross-section obtained from Yim et al. method. Ripples (combination of convex and concave surface patches) are noticeable on their surface. (c) A stream surface (color-coded with mean curvatures) towards one end of the axis.

however, is capable of keeping the surfaces as flat as possible at the arc region, as reflected in the close-to-zero mean curvatures in the plot.

One may notice that the curvature values shown in the plot of the stream surfaces towards both axial ends are comparatively high. This is because the relatively sharp bending at the edge of the surfaces located in the inner region of the arc, see Fig. 19c for the surface (color-coded with mean curvatures) towards one end of the axis.

4.3. Vascular segmentation

We have tested the proposed segmentation method on a synthetic data set. The data set contains a segment of a horizontal tube with the same diameter along its axis and has two consecutive opposite turns at the middle of the segment, as given in Fig. 20a. The data sets are created to emulate the field of view of the region of interest (6–150 mm) and the image volume ($80 \times 120 \times 80$ vox.) of typical 3-D angiographies taken at intracranial circulations. The intensity value of the voxels laying inside the tube equals 255, and it is set to 0 in the background. Fig. 20d shows the 40th slice of the data set.

Vessel axes and boundary surface models are built from four noise corrupted copies (with signal-to-noise ratio equal to 2, 3, 5 and 7) of the synthetic data set. White Gaussian noise is added to produce the noisy volumes. Signal-to-noise ratio (SNR) is defined as the ratio of the peak intensity value to the sample standard deviation of the noise. For example, Gaussian noise with the sample standard deviation equals 85 is added to the data set to generate a noisy volume with SNR equal to 3. Figs. 20e and f shows the 40th slice of the noise corrupted volumes with SNR equal to 5 and 2, respectively. The

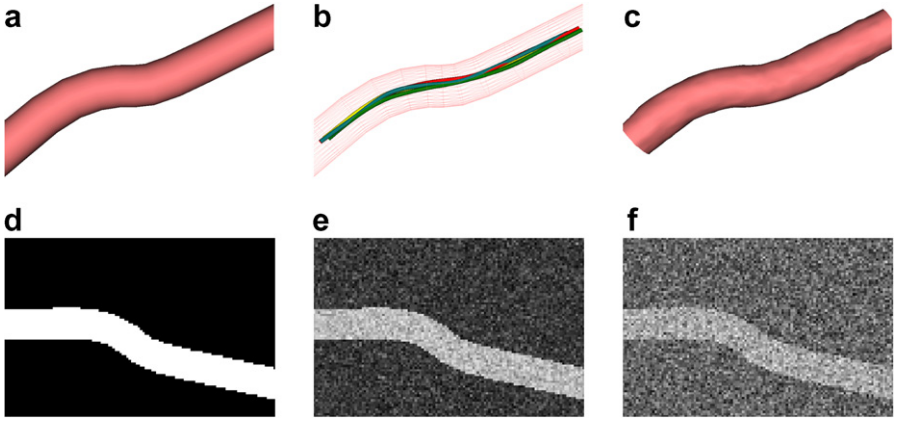

Fig. 20. (a) The synthetic model, a segment of a horizontal tube with the same diameter along its axis and has two consecutive opposite turns at the middle of the segment. (b) The axes extracted from each noisy volume with the algorithm described in Section 2. (c) The surface model built from the noisy volume with SNR equal to 2. (d) The 40th slice of the synthetic data set. The intensity value of the voxels laying inside the tube equals 255, and it is set to 0 in the background. (e) and (f) The 40th slice of the noise corrupted volumes with SNR equal to 5 and 2, respectively.

axes extracted from the noisy volumes with the algorithm described in Section 2 are shown in Fig. 20b. Although there are small discrepancies in the positions of the extracted axial points amongst those volumes, they are all within the confines of the tube boundary and we have experienced no undesired effect on the construction of the boundary surface models. The surface model built from the noisy volume with SNR equal to 2 is given in Fig. 20c for illustration.

Table 2
Quantitative analysis on the segmentations of the synthetic data sets which are corrupted by additive white Gaussian noise

Noise sample std. dev.	SNR	JSC
36.4	7	0.9646
51.0	5	0.9422
85.0	3	0.9646
127.5	2	0.9387

The sample standard deviation of the additive noise, the signal-to-noise ratio (SNR) and the Jaccard similarity coefficient (JSC) computed with the truth segmentation (the noiseless volume) are tabulated.

Quantitative analysis on the segmentations produced with the algorithm presented in Section 3 is conducted on the basis of Jaccard similarity coefficient (JSC) between the obtained segmentations and the truth (the noiseless volume). JSC is defined as the ratio of the size of the intersection volume to the size of the union volume of the two given segmentations (Leemput et al., 1999). It is used to quantify the accuracy of a segmentation. JSC gives value 1 if the segmentation equals the truth. Table 2 lists the JSC values obtained from sub-volumes (20th–99th slices along the y-axis; this is to ignore the portions of no interest) of the segmentations. It is observed that our algorithm can produce segmentations with high degree of similarity to the truth segmentation across different levels of additive white Gaussian noise. This demonstrates the robustness of the proposed algorithm towards noise, even severe noise at the level of SNR equal to 2.

Furthermore, we have applied the proposed algorithm to sub-volumes of three clinical data sets. These data are 3-D rotational angiographies (RA) acquired by the Philips Integris imager at the Department of Diagnostic Radiology and Organ Imaging, Prince of Wales Hospital, Hong

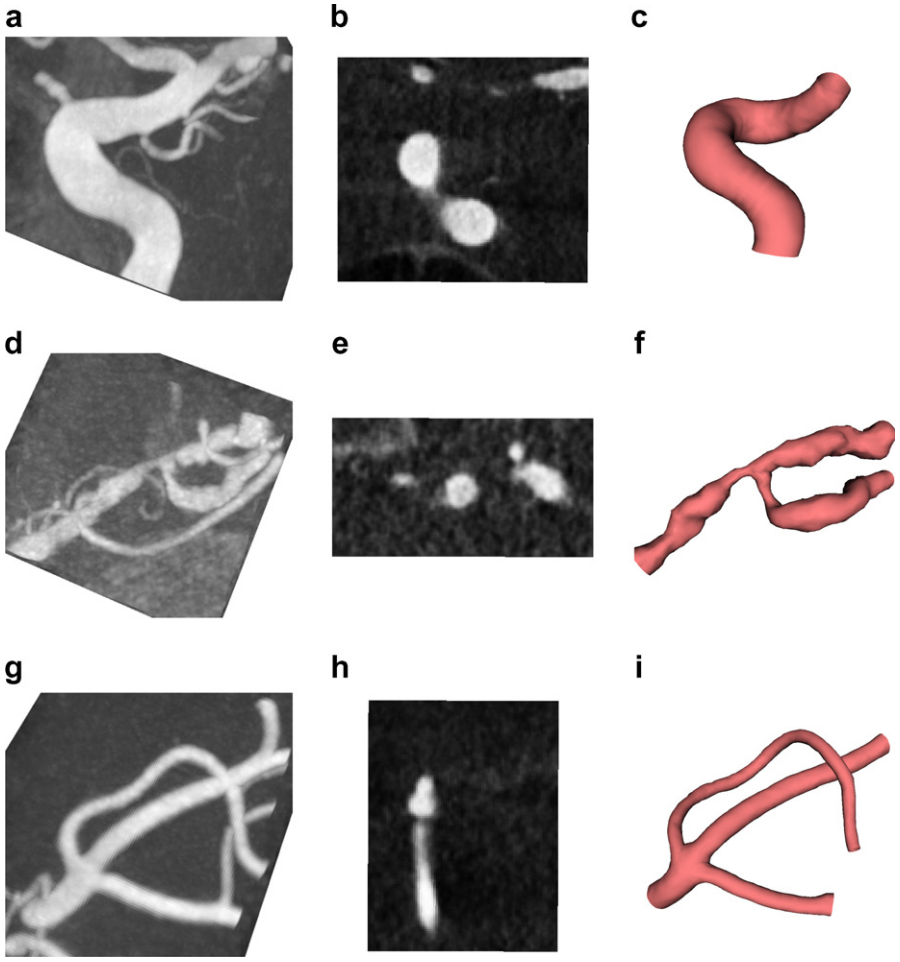

Fig. 21. Volume rendering (VR) images, sample slices and the vessel boundary surface models built of the three sub-volumes from the clinical data sets: (a–c) a portion of ICA that lays inside the cranium; (d–f) a bifurcation at the horizontal (M1) segment of MCA; and (g–i) a furcation at the Sylvian (M2) segment of MCA.

Table 3
Jaccard similarity coefficient (JSC) values calculated amongst every pair of the three segmentations—obtained from our framework and slice-by-slice segmentation approach by two human raters—on the three sub-volumes which contain a portion of ICA that lays inside the cranium, a bifurcation at the horizontal (M1) segment of MCA and a furcation at the Sylvian (M2) segment of MCA

Data Set	Rater 1 vs. Rater 2	Rater 1 vs. Ours	Rater 2 vs. Ours
ICA	0.8854	0.9094	0.9102
M1 MCA	0.7479	0.7564	0.7784
M2 MCA	0.7279	0.7219	0.7355

Kong. The size of the image volume is $256 \times 256 \times 256$ vox. with the field of view 30–50 mm. The three sub-volumes contain individually (a) a portion of ICA that lays inside the cranium, (b) a bifurcation at the horizontal (M1) segment of MCA and (c) a furcation at the Sylvian (M2) segment of MCA. The segmentation of the vascular junctions are composed by aggregating the segmentation of each vessel branch as described in Section 3.5. A volume rendering image, a sample slice and the vessel boundary surface model(s) built of each sub-volumes are given in Fig. 21 for illustrations. It is evident that the proposed algorithm is capable of producing not only a segment of a single vessel lumen (see Fig. 21c), but also a diseased lumen where vessel width changes dramatically (see Fig. 21f) and multi-way vascular junctions in which the width of the branches may not necessarily be the same (see Figs. 21f and i).

Due to the fact that the truth segmentations of those clinical data sets do not exist, we have asked two human raters to compose segmentations by assigning binary labels to each voxel in the slices. The open source software SNAP (available at http://www.itksnap.org) is employed for the composition. JSC values are calculated amongst every pair of the three segmentations (obtained from our algorithm and the two human raters) per sub-volume. The results are tabulated in Table 3. It is evident that our framework is capable of producing vascular segmentations that have level of variability similar to those obtained from the human raters. This study suggests that our algorithm can be good enough to replace the slice-by-slice manual segmentation approach, which indeed is very tedious and time consuming, to delineating vasculatures.

5. Conclusions

We have proposed a novel framework to extract vessel axes and delineate vessel boundaries in a decoupled fashion. This framework avoids incomplete segmentation of the vessel of interest due to axis-trace re-initialization and disconnected vessel axis in regions that contain furcations, diseased portions and kissing vessels (aka problematic regions), as found in the other axis-based vascular segmentation algorithms. It is particularly suitable for delineating the (diseased) vasculature of a small region of interest, in which a greater degree of user control is desired. Our framework starts with the extraction of vessel axis from the original angiogram, followed is the minimum cost paths (MCP)-based vessel boundary delineation on cross-sections defined by the extracted axis. The design philosophy of this framework is to promote a more intuitive vessel boundary delineation on the vessel cross-sections, where the boundaries are mostly round, compared with those on either axial, coronal or sagittal image planes. Moreover, such design allows direct vessel-axis-tracing, which avoids the need of further rectification of the vessel axes if they are extracted from a segmentation that is subject to topological or morphological incorrectness.

Our vessel cross-sections are generated as stream surfaces. They are not necessarily on flat planes such that non-intersecting cross-sections are guaranteed. This makes the construction of a 3-D vessel boundary surface model with no self-intersection very easily—by just joining the vertices of adjacent contours that delineate the vessel boundaries on the cross-sections. Insertion of extra cross-sections between the existing ones is also feasible without any re-computation of the existent cross-sections. This permits an increase in the longitudinal resolution of the vessel boundary surface model if it is desirable. Furthermore, the surface model constructed inherently has a couple of properties that suit the finite element grid generation for computational fluid dynamics (CFD) application: (1) the vertices at high curvature regions (i.e., the saddle regions, the inner boundaries of a curved vessel) are denser which allows more accurate modeling; and (2) the triangulations of the cross-sections partition the vascular volume into non-overlapping layers—volume between two adjacent cross-section triangulations—(thanks to the non-intersecting property of the cross-sections) which can form the basis (i.e., a set of points in a 3-D space) for free meshing (Huebner et al., 2001).

Because of the probabilistic formulation of the vessel axis tracer, we find the axial points and their associated reference frame in a single step as a maximum *a posteriori* (MAP) estimation. This is in marked contrast to the other algorithms which take alternate steps to search those quantities. Our approach guarantees that the axial points are apart at a user-defined distance, which helps avoid sudden jumps of the trace and large spatial discontinuities between the axial points. In addition, it allows incorporation of user guidance (2-D curves drawn on the screen by the user) on the axis tracing via the prior pdf. The user guidance can effectively help drive the axis tracer to fly-through the problematic regions, whose necessity is identified visually by the user on a display of either the iso-surface or MIP image of the vasculature and the traced axis. The MCP approach to segmenting vessel on its cross-sections also allows the user to place anchor points to guide the vessel boundary delineation.

Experimental results on synthetic and clinical data have shown that our probabilistic axis tracer can extract less jittering (as compared with the other two trace-based algorithms) and continuous axes in the problematic regions,

and the segmentation algorithm with cross-sections has high robustness towards noise and can delineate vessel boundaries that have level of variability similar to those obtained manually. This study suggests that the proposed framework can be good enough to replace the time-consuming and tedious slice-by-slice manual segmentation approach.

There are several limitations in the proposed algorithm, for instance, user intervention is necessary if one wants to trace the side branches of a furcation; and the anatomical structures to be traced in the image volumes have to be the brightest and symmetric along their axis due to the appearance model. Those are of interest for further research and may lead to potential extensions to the method: (1) reducing the amount of intervention in tracing side branches or supporting automatic branches tracing at furcations; (2) extending the method to trace axes of vessels that are not the brightest anatomical structures in the image volumes (e.g. vessels in computed tomography angiography where bones and metallic instruments are the brightest objects), and to trace axes of anatomies other than blood vessels (especially those with asymmetric cross-sections) with different appearance models and/or ways to estimate the likelihood pdf; and (3) incorporating different image features other than the gradient magnitude into the formulation of the DAG edge weights in the MCP-based segmentation with cross-sections.

Acknowledgements

This work was supported by the Research Grants Council (RGC) of Hong Kong under Grant HKUST612305. The authors thank Dr. Simon C.H. Yu for providing the clinical data sets for validation.

References

Aylward, S.R., Bullitt, E., 2002. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE Trans. Med. Imag. 21 (2), 61–75.

Bloomenthal, J., 1996. Skeletal Design of Natural Forms. Ph.D. thesis, Department of Computer Science, University of Calgary, Canada, July 1996.

Bullitt, E., Muller, K.E., Jung, I., Lin, W., Aylward, S., 2004. Analyzing attributes of vessel populations. Med. Imag. Anal. 9, 39–49.

Cormen, T., Leiserson, C., Rivest, R., 1990. Introduction to Algorithms. McGraw-Hill.

Farag, A.A., Hassouna, M.S., Falk, R., Hushek, S., 2004. Reliable Fly-throughs of Vascular Trees. Technical Report, Department of Electrical and Computer Engineering, University of Louisville. URL <http://www.cvip.louisville.edu/wwwcvip/research/publications/TechReport/ReliableFlyThrough.pdf>.

Felkel, P., Wegenkittl, R., Buhler, K. 2004. Surface models of tube trees. In: Proceedings of the Computer Graphics International, pp. 70–77.

Flasque, N., Desvignes, M., Constans, J.-M., Revenu, M., 2001. Acquisition, segmentation and tracking of the cerebral vascular tree on 3D magnetic resonance angiography images. Med. Imag. Anal. 5, 173–183.

Hassouna, M.S., Farag, A.A., Falk, R., 2005. Differential fly-throughs (DFT): a general framework for computing flight paths. Medical Imaging Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, vol. 3749. Springer-Verlag, Berlin, pp. 654–661.

Hernández-Hoyos, M., Anwander, A., Orkisz, M, Roux, J.-P., Douek, P., Magnin, I.E., 2000. A deformable vessel model with single point initialization for segmentation, quantification and visualization of blood vessels in 3D MRA. In: Medical Imaging Computing and Computer-Assisted Interventions, Lecture Notes in Computer Science, vol. 1935, pp. 735–745. URL <http://www.springerlink.com/openurl.asp?genre=article&id=Q0F4BYKWHQYMTDNP>.

Huebner, K.H., Dewhirst, D.E., adn Smith, D.L., Byrom, T.G., 2001. The Finite Element Method for Engineers, forth ed. Wiley Interscience.

Hultquist, J.P.M., 1992. Constructing stream surfaces in steady 3D vector fields. In: Visualization, Boston, MA, pp. 171–178, (also RNR Technical Report 92-025).

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: active contour models. Int. J. Comput. Vision 1, 321–331.

Keppel, E., 1975. Approximating complex surfaces by triangulation of contour lines. IBM J. Res. Dev. 19, 2–11.

Kita, Y., Wilson, D.L., Noble, J.A., Real-time registration of 3D cerebral vessels to X-ray angiograms. In: Medical Imaging Computing and Computer-Assisted Intervention, vol. 1496, 1998, pp. 1125–1133. URL <http://www.springerlink.com/openurl.asp?genre=article&id=LYJW-CHY7YN5M2BCU>.

Kochanek, D., Bartels, R., 1984. Interpolating splines with local tension, continuity and bias control. Comput. Graph. 18 (3), 33–41.

Leemput, K.V., Maes, F., Vandermeulen, D., Colchester, A., Suetens, P., 1999. Automated segmentation of MS lesions from multi-channel MR images. Medical Imaging Computing and Computer-Assisted Interventions. In: Lecture Notes in Computer Science, vol. 1679. Springer, Verlag, Berlin, pp. 11–21.

Lobregt, S., Viergever, M.A., 1995. A discrete dynamic contour model. IEEE Trans. Med. Imag. 14 (1), 12–24.

McIntosh, C., Hamarneh, G., 2006. Vessel crawlers: 3D physically-based deformable organisms for vasculature segmentation and analysis. Comput. Vis. Pattern Recogn. 1, 1084–1091.

Ogniewicz, R.L., Kubler, O., 1995. Hierarchy Voronoi skeletons. Pattern Recogn. 28 (3), 343–359.

Palágyi, K., Sorantin, E., Balogh, E., Kuba, A., Halmai, C., Erdöhelyi, B., Hausegger, K., 2001. A sequential 3D thinning algorithm and its medical applications. Internatoinal Conference on Information Processing Medical Imaging. In: Lecture Notes in Computer Science, vol. 2082. Springer-Verlag, pp. 409–415.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in C: The Art of Scientific Computing, second ed. Cambridge University Press, Cambridge.

Rinkel, G.J., Djibuti, M., Algra, A., van Gijn, J., 1998. Prevalence and risk of rupture of intracranial aneurysms: a systematic review. Stroke 29, 251–256.

Schroeder, W.J., Martin, K.M., Lorensen, W.E., 2003. The Visualization Toolkit: An Object Oriented Approach to 3D Graphics, third ed. Kitware, Inc..

Shim, H., Yun, I.D., Lee, K.M., Lee, S.U., Partition-based extraction of cerebral arteries from CT angiography with emphasis on adaptive tracking. In: International Conference Information Processing Medical Imaging, Lecture Notes in Computer Science, vol. 3565. 2005, pp. 357–368.

Styner, M., Gerig, G., Brechbuehler, C., Szekely, G., 2000. Parametric estimate of intensity inhomogeneities applied to MRI. IEEE Trans. Med. Imag. 19 (3), 153–165.

Sun, C., Pallottino, S., 2003. Circular shortest path in images. Pattern Recogn. 36 (3), 709–719.

Suri, J.S., Liu, K., Reden, L., Laxminarayan, S., 2002. A review on MR vascular image processing algorithms: acquisition and prefiltering: part I. IEEE Trans. Inform. Technol. Biomed. 6 (4), 324–337.

van der Weide, R., Zuiderveld, K.J., Mali, W.P.T.M., Viergever, M.A., 1998. CTA-based angle selection for diagnostic and interventional

angiography of saccular intracranial aneurysms. IEEE Trans. Med. Imag. 17 (5), 831–841.

Wink, O., Niessen, W.J., Viergever, M.A., 2000. Fast delineation and visualization of vessels in 3-D angiographic images. IEEE Trans. Med. Imag. 19 (4), 337–346.

Yim, P.J., Cebral, J.J., Mullick, R., Marcos, H.B., Choyke, P.L., 2001. Vessel surface reconstruction with a tubular deformable model. IEEE Trans. Med. Imag. 20 (12), 1411–1421.

Zhou, Y., Toga, A.W., 1999. Efficient skeletonization of volumetric objects. IEEE Trans. Visual. Comput. Graphics 5 (3), 196–209.