Efficient Implementation for Spherical Flux Computation and Its Application to Vascular Segmentation

Max W. K. Law and Albert C. S. Chung

***Abstract*—Spherical flux is the flux inside a spherical region, and it is very useful in the analysis of tubular structures in magnetic resonance angiography and computed tomographic angiography. The conventional approach is to estimate the spherical flux in the spatial domain. Its running time depends on the sphere radius quadratically, which leads to very slow spherical flux computation when the sphere size is large. This paper proposes a more efficient implementation for spherical flux computation in the Fourier domain. Our implementation is based on the reformulation of the spherical flux calculation using the divergence theorem, spherical step function, and the convolution operation. With this reformulation, most of the calculations are performed in the Fourier domain. We show how to select the frequency subband so that the computation accuracy can be maintained. It is experimentally demonstrated that, using the synthetic and clinical phase contrast magnetic resonance angiographic volumes, our implementation is more computationally efficient than the conventional spatial implementation. The accuracies of our implementation and that of the conventional spatial implementation are comparable. Finally, the proposed implementation can definitely benefit the computation of the multiscale spherical flux with a set of radii because, unlike the conventional spatial implementation, the time complexity of the proposed implementation does not depend on the sphere radius.**

***Index Terms*—Efficient implementation, flux, vascular segmentation.**

I. Introduction

In vector calculus, flux of a vector field over the closed boundary of a region is the quantification of the amount of the vectors, which flows into or out of that region along the boundary surface normal direction. Mathematically, in the Euclidean space, the flux of a vector field $\vec{v}$ over the closed boundary of a region C is defined as [24]

$$\text{flux}_C = \int_{\partial C} < \vec{v}, \hat{n} > d\rho$$

where ∂C is the closed boundary of C, $\hat{n}$ is the outward normal of ∂C. By specifying the vector field $\vec{v}$ as the gradient vector field of a grayscale image I we have, by the divergence theorem

$$\text{flux}_C = \int_{\partial C} < \vec{v}, \hat{n} > d\rho \equiv \int_C \Delta I d\nu.$$

The computation of flux associated with arbitrary regions is widely used for digital image analysis [27], [3], [17], [25], [21], [1], [7], [8], [16], [18], [29], [14], [15], [23], [12], [2].

In particular, for some applications such as analysis of tubular structures, it is useful to compute flux in either a 2-D circular region or a 3-D spherical region. For instance, Siddiqi *et al.* have proposed to compute flux in discs or spheres with different radii to aid in obtaining the skeletons of target objects [25]. Pizer *et al.* have presented the use of flux for recognizing medial loci [21]. In [3], the disc-based or sphere-based calculations of flux have been utilized for centerline extraction of blood vessels in magnetic resonance angiography and colons in computed tomography. For segmentation of tubular structures, e.g., blood vessels in brain, Vasilevskiy and Siddiqi have proposed flux maximizing geometric flows [27] by approximating the divergence of image gradient as the flux estimated over the closed boundaries of discs or spheres with different radii. This segmentation algorithm is useful in extracting small and dim blood vessels because of its sensitivity to low contrast and narrow tubular structures. Its performance has been proved and validated as it has been employed for generating the ground truth for the BrainWeb project [5], which is widely used in medical image segmentation validation. In addition, Audette and Chinzei have extended the flux maximizing geometric flows for tissue identification [1].

Among the aforementioned techniques, it is essential to compute flux in discs or spheres. In this paper, flux computed in a spherical region is called *spherical flux*. The continuous form of the spherical flux f_s is given as

$$f_s(x,y,z) = \int_{\partial R_s} \vec{v}(x+t, y+p, z+q) \cdot \hat{n}_{(t,p,q)} dA \quad (1)$$

where R_s represents a spherical region with radius s; dA is the infinitesimal area on boundary ∂R_s; $\hat{n}_{(t,p,q)}$ is the boundary surface normal vector at position (t,p,q); and $\vec{v}$ is the gradient of an image I. In practice, $\vec{v}$ is obtained from a Gaussian smoothed image, which ensures the differentiability of the discrete image signal I, i.e., $\vec{v} = \nabla(g * I)$, where g is a Gaussian function. In a discrete form, (1) may be computed numerically as

$$f_s[x,y,z] = \sum_{i=1}^{K} \vec{v}[x+st_i, y+sp_i, z+sq_i] \cdot (t_i, p_i, q_i)^T \quad (2)$$

Manuscript received June 07, 2007; revised September 29, 2008. Current version published February 11, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter C. Doerschuk.

The authors are with the 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 (e-mail: maxlawwk@cse.ust.hk; achung@cse.ust.hk).

Digital Object Identifier 10.1109/TIP.2008.2010073

1057-7149/$25.00 © 2009 IEEE

where $(t_i, p_i, q_i)^T$, $\forall i \in [1, 2, \ldots, K]$, represents the ith outward normal of ∂R_s, $\sqrt{t_i^2 + p_i^2 + q_i^2} = 1$, and K is the total number of orientation samples on ∂R_s.

It is straightforward to estimate the values of spherical flux for an image using (2) in the spatial domain. It is called the *conventional spatial implementation* hereafter. Although it is easy to implement the conventional spatial formulation, the computation time taken by the conventional spatial implementation is unsatisfactorily long when the radius s is large. The reason is that a large value of s boosts the boundary surface area of the spherical region. Thus, in (2), a considerable number of orientation samples is needed to precisely approximate the surface integral, as stated in (1).

To reduce the time required to compute spherical flux, we propose a more efficient implementation, which is formulated in the Fourier domain. It avoids the orientation sampling for the discretization of the surface integral, which is given in (1) and, thus, dramatically reduces the running time complexity of the spherical flux computation. The computation time required for the proposed implementation has been compared with that of the conventional spatial implementation (2) using both synthetic and clinical image volumes with various sizes. The computation accuracies of both implementations have also been validated. It is experimentally shown that the proposed efficient implementation is accurate and capable of remarkably reducing the running time for the spherical flux computation. The proposed implementation can definitely benefit flux-based applications such as tubular structure analysis. It is because techniques for such applications grounded on spherical flux, e.g., flux maximizing geometric flows [27], can have significant computation time reduction by employing the proposed implementation.

This paper is organized as follows. Section II gives the orientation sampling strategy and time complexity for the conventional spatial implementation of the spherical flux computation. In Section III, we elaborate the formulation of the proposed implementation. The accuracy and the running time comparisons based on both synthetic and clinical image volumes are presented in Section IV. Section V concludes this paper. The detailed mathematical formulation is provided in the Appendix.

II. Conventional Spatial Implementation: Orientation Sampling Strategy and Time Complexity

This section gives the orientation sampling strategy and the time complexity of the conventional spatial implementation for estimating the spherical flux f_s using the (2). On the boundary surface of a spherical region ∂R_s, with K orientation samples, the sum of dot products of the image gradient $\vec{v}$ and outward normals is computed for approximating the surface integral given in (1). Bilinear interpolation is employed to retrieve the vector $\vec{v}$ in (2) if its position is not on the exact grid positions. Intuitively, the accuracy of the conventional spatial implementation is closely related to the number of orientation samples K taken in (2).

The value of K is a trade-off between the computation time and the accuracy. It is natural to have one orientation sample for each unit area in the unit of voxel-length on the surface of the spherical region ∂R_s for a good balance between the computation time and the accuracy. Therefore, the number of orientation samples taken by the conventional spatial implementation K is specified to ensure that there is at least one orientation sample for each unit area in the unit of voxel-length on the sphere surface. The orientation samples are organized in a grid fashion along the longitudinal and the latitudinal directions. For a sphere with radius s, there are $\lceil \pi s \rceil$ angularly equally spaced latitudinal levels. These latitudinal levels are specified by elevation angles within a range $(-\pi/2, \pi/2]$. For the lth latitudinal level with an elevation angle θ_l, there are $\lceil 2\pi s \cos\theta_l \rceil$ orientation samples taken on the circumference of the circle associated with the lth latitudinal level.

Using the above orientation sampling strategy, the number of orientation samples, K is directly proportional to the surface area of the sphere with radius s, i.e., $K \propto s^2$. These orientation samples are roughly equally spaced for the spatial computation of spherical flux. Therefore, based on (2), the time complexity for computing the spherical flux using the conventional spatial implementation in the entire image domain with N voxels is

$$O(KN) = O(s^2 N). \tag{3}$$

As the computation time increases quadratically with the value of s, it hinders the implementation of multiscale spherical flux with different radii, especially when the radius s is large. This will be demonstrated in Section IV.

III. Methodology

In this section, we present our proposed implementation of the spherical flux in the Fourier domain (Section III-A). The fast Fourier transform algorithm, which is used in our implementation, is also described in Section III-B. Finally, for the detection of blood vessels with different widths, multiscale spherical flux is utilized and a method for reducing computation time is presented in Section III-C.

A. Our Efficient Implementation in the Frequency Domain

Although the estimation of spherical flux based on (2) is straightforward in the spatial domain, as described in Sections I and II, the computation speed is unsatisfactorily slow when the size of the sphere s is large. To shorten the time for computing the spherical flux, we propose to reformulate the calculation of the spherical flux using the divergence theorem and spherical step function d_s. As such, the spherical flux for the entire image is computed based on the convolution between the input image and a function h_s. The function h_s embeds the second derivative operator, the Gaussian smoothing function and the spherical step function d_s. Then, using the convolution theorem, the spherical flux can be estimated by multiplying the Fourier expressions of the image and the function h_s, which are calculated using the fast Fourier transform algorithm. As is shown in the experiments (see Section IV), our implementation can lead to a significant computation gain, in particular when the sphere radius s is large.

1) Reformulating the Spherical Flux Computation Using Convolution Operations: We rewrite (1) into a convolution operation between the input image I and a function h_s. By employing the divergence theorem and using spherical step function

$$\begin{aligned} f_s(x, y, z) &= \int_{R_s} \mathrm{div}(\vec{v}(x+t, y+p, z+q))\, dt dp dq, \\ &= \int_{\Omega} d_s(t, p, q)[\mathrm{div}(\vec{v}(x+t, y+p, z+q))]\, dt dp dq \end{aligned} \tag{4}$$

where Ω is the entire image domain, $d_s(t,p,q)$ is a spherical step function, which is given as

$$d_s(x,y,z) = \begin{cases} 1, & \sqrt{x^2+y^2+z^2} \le s, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The main purpose of introducing the spherical step function in the above equation is to allow us to reformulate the calculation of spherical flux using the convolution operation. From (4), we have

$$\begin{aligned} f_s(x,y,z) &= \int_\Omega d_s(t,p,q) \\ &\quad \times [\operatorname{div}(\vec{v}(x+t,y+p,z+q))]dtdpdq \\ &= \int_\Omega d_s(t,p,q) \\ &\quad \times (\Delta(g*I(x+t,y+p,z+q)))dtdpdq \\ &= \int_\Omega d_s(-t,-p,-q) \\ &\quad \times (\Delta(g*I(x+t,y+p,z+q)))dtdpdq \\ &= d_s*\Delta g*I(x,y,z) \\ &= I*h_s(x,y,z) \end{aligned} \tag{6}$$

where $*$ represents the convolution operator, Δ represents the Laplacian operator, $\vec{v}$ is replaced by the Gaussian smoothed image gradient $\nabla(g*I(x,y,z))$ and $h_s(x,y,z) = \Delta g * d_s(x,y,z)$.

2) Representing the Spherical Flux in the Fourier Domain: The above derivation leads to a convolution between the image I and the function h_s. The convolution is implemented as a multiplication in the Fourier domain. Such multiplication merely involves the Fourier transformed image and the Fourier expression of the function of $h_s(x,y,z)$, denoted as $H_s(u,v,w)$, where u, v and w are frequencies (in cycle per image) in the Fourier domain along the x, y and z directions, respectively.

It is noted that the Fourier expression $H_s(u,v,w)$ embodies three components, including the three Fourier expressions of the second derivative operator, the Gaussian smoothing function and the spherical step function d_s. Let the Fourier expression of the spherical step function d_s be $D_s(u,v,w)$. Therefore, $H_s(u,v,w)$ is given as

$$\begin{aligned} H_s&(u,v,w) \\ &= \left[(j2\pi)^2\left(\left(\frac{u}{N_x}\right)^2 \right.\right. \\ &\qquad \left.\left. + \left(\frac{v}{N_y}\right)^2 + \left(\frac{w}{N_z}\right)^2\right)\right] \\ &\quad \times \left[\exp\left(-\left(\left(\frac{u}{N_x}\right)^2 \right.\right.\right. \\ &\qquad \left.\left.\left. + \left(\frac{v}{N_y}\right)^2 + \left(\frac{w}{N_z}\right)^2\right)2(\pi\sigma)^2\right)\right] \\ &\quad \times D_s(u,v,w) \end{aligned} \tag{7}$$

where the dimensions of the image are N_x, N_y and N_z along the x, y and z directions, respectively. σ is the scale parameter of the Gaussian function being applied on the image $I(x,y,z)$.

To simplify the formulation, we define $k(u,v,w) = 2\pi\sqrt{((u)/(N_x))^2+((v)/(N_y))^2+((w)/(N_z))^2}$, which is the radial component of the frequencies. (7) is then rewritten as

$$\begin{aligned} H_s&(u,v,w) \\ &= (jk(u,v,w))^2 e^{-k^2(u,v,w)\sigma^2/2} D_s(u,v,w) \\ &= -k^2(u,v,w) e^{-k^2(u,v,w)\sigma^2/2} D_s(u,v,w). \end{aligned} \tag{8}$$

In the above equation, the Fourier expression of the spherical step function $D_s(u,v,w)$ is calculated by applying the Fourier transform on the spherical step function and using 3-D Hankel transforms [4], details are provided in Appendix A

$$\begin{aligned} H_s&(u,v,w) \\ &= -k^2(u,v,w) e^{-k^2(u,v,w)\sigma^2/2} D_s(u,v,w) \\ &= -4\pi s e^{-k^2(u,v,w)\sigma^2/2} \\ &\quad \times [\operatorname{sinc}(sk(u,v,w)) - \cos(sk(u,v,w))]. \end{aligned} \tag{9}$$

Equation (9) is the Fourier expression of the term $\Delta g * d_s(x,y,z)$. It is a spherically symmetric and real function, and its magnitude decays along the radial direction (see an example of $H_s(u,v,w)$ in Fig. 1).

3) Estimating Spherical Flux in the Fourier Domain: By (6), the spherical flux is calculated as

$$f_s(x,y,z) = \mathcal{FFT}^{-1}\{J(u,v,w)H_s(u,v,w)\} \tag{10}$$

where $\mathcal{FFT}^{-1}$ is the inverse fast Fourier transform (IFFT) operator, $J(u,v,w)$ is the Fourier transformed image, i.e., $J(u,v,w) = \mathcal{FFT}(I(x,y,z))$, and $H_s(u,v,w)$ is defined in (9). The time complexity is $O(N\log N)$ for both FFT and IFFT, and $O(N)$ for the multiplication. Therefore, the complexity of the proposed implementation is

$$O(N\log N). \tag{11}$$

The running time of the proposed implementation is independent of the value of s, i.e., the size of the sphere. Theoretically, our implementation has speed advantage over the conventional spatial implementation when s is large and N is small. As we will show in Section IV, the proposed implementation is much faster than the conventional spatial implementation, even when s is small ($s = 1$) and N is sufficiently large (details can be found in Section IV) in the vascular image analysis applications, assuming that the orientation sampling strategy as described in Section II is used.

4) Discretizing the Function h_s: Theoretically, the formulation of $h_s(x,y,z)$ is continuous in the spatial domain. Practically, image I is a discrete signal. Therefore, discretization of $h_s(x,y,z)$ is needed and (6) becomes

$$f_s^d[x,y,z] = I[x,y,z] * h_s^d[x,y,z] \tag{12}$$

where the superscript d denotes that the corresponding function is discretized according to a predefined frequency bandwidth. Similarly, (10) is rewritten as

$$f_s^d[x,y,z] = \mathcal{DFFT}^{-1}\{J[u,v,w]H_s^{d,\infty}[u,v,w]\} \tag{13}$$

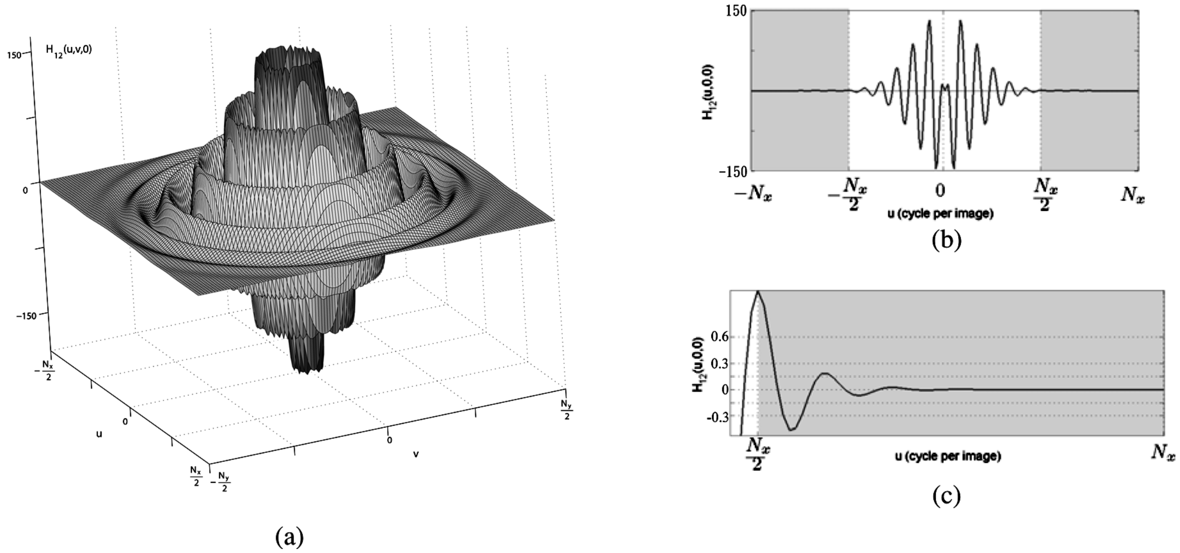


Fig. 1. Example of the function $H_s(u,v,w)$ when $s = 12$. (a) The plot of the coefficients for the plane $w = 0$ in the Fourier domain. (b) The plot of the coefficients for the line $v = 0, w = 0$ in the Fourier domain. The shaded regions are not covered by the $\mathrm{Subband}_1$. (c) The zooming of (b).

where u, v and w are integers and

$$
\begin{aligned}
-\left\lceil \frac{N_x}{2} \right\rceil + 1 \le u \le \left\lfloor \frac{N_x}{2} \right\rfloor \\
-\left\lceil \frac{N_y}{2} \right\rceil + 1 \le v \le \left\lfloor \frac{N_y}{2} \right\rfloor \\
-\left\lceil \frac{N_z}{2} \right\rceil + 1 \le w \le \left\lfloor \frac{N_z}{2} \right\rfloor .
\end{aligned} \tag{14}
$$

$\mathcal{DFFT}^{-1}$ is the inverse discrete fast Fourier transform operator, $H_s^{d,\infty}[u,v,w]$ is the Fourier expression of $h_s^d[x,y,z]$ and the symbol "∞" represents that $H_s^{d,\infty}[u,v,w]$ covers infinity bandwidth in the Fourier domain, i.e.,

$$
H_s^{d,\infty}[u,v,w] = \sum_{A,B,C=-\infty}^{\infty} H_s [u + AN_x, v + BN_y, w + CN_z] \tag{15}
$$

where $H_s^{d,\infty}[u,v,w]$ in the above equation is equivalent to performing discrete sampling on $h_s(x,y,z)$ [20].

It is noted that $H_s^{d,\infty}[u,v,w]$ is a sum-to-infinity series, which is not efficient for implementation. Indeed, the magnitude of $H_s(u,v,w)$ (9) is decaying according to the term $e^{-k^2(u,v,w)\sigma^2/2}$. Therefore, it is not necessary to evaluate $H_s(u,v,w)$ in the entire Fourier domain with infinity bandwidth. Ignoring the high frequency regions where the magnitudes of $H_s(u,v,w)$ are negligibly small does not harm the calculation of spherical flux. Noted that the decaying term $e^{-k^2(u,v,w)\sigma^2/2}$ of $H_s(u,v,w)$ is a Gaussian function having a scale factor σ^{-1}. We assume that the value of the Gaussian function, and, hence, the values of $H_s(u,v,w)$ are sufficiently small and can be treated as zero when the value of k is larger than four times of the scale factor σ^{-1}, i.e.,

$$
H_s(u,v,w) \approx 0 \text{ for } k(u,v,w) \ge \frac{4}{\sigma} \tag{16}
$$

where $e^{-k^2(u,v,w)\sigma^2/2} \le e^{-8} < 10^{-3}$. Based on this criterion, we ignore the computations involved in the regions, in which $k(u,v,w) \ge \frac{4}{\sigma}$, and, thus, the values of $H_s(u,v,w)$ are negligible. It leads to a simplification of (13) based on the scale factor of the Gaussian function, which is being applied on $I(x,y,z)$ for the calculation of the spherical flux. The region for computing $H_s(u,v,w)$ is trimmed in each dimension separately. It is essential to correctly calculate the subbands of images having different lengths along various dimensions.

We now consider the simplest case that $H_s^{d,\mathrm{Subband}_1}[u,v,w]$ directly substitutes $H_s^{d,\infty}[u,v,w]$ in (13)

$$
H_s^{d,\infty}[u,v,w] \approx H_s^{d,\mathrm{Subband}_1}[u,v,w]. \tag{17}
$$

Here, $H_s^{d,\mathrm{Subband}_1}[u,v,w]$ is obtained by evaluating $H_s[u,v,w]$ merely in the frequency $\mathrm{Subband}_1$, which is defined as

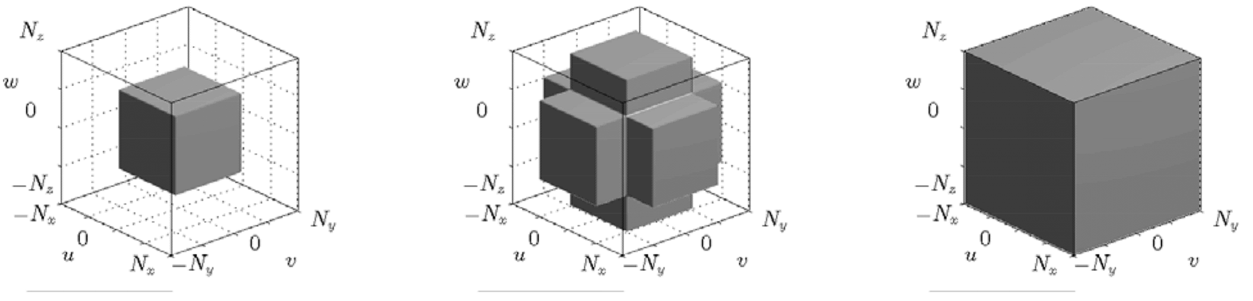
$$
\begin{aligned}
\mathrm{Subband}_1 \\
= \Bigg\{ \left\{ -\left\lceil \frac{N_x}{2} \right\rceil + 1 \le u \le \left\lfloor \frac{N_x}{2} \right\rfloor \right\} \\
\cap \left\{ -\left\lceil \frac{N_y}{2} \right\rceil + 1 \le v \le \left\lfloor \frac{N_y}{2} \right\rfloor \right\} \\
\cap \left\{ -\left\lceil \frac{N_z}{2} \right\rceil + 1 \le w \le \left\lfloor \frac{N_z}{2} \right\rfloor \right\} \Bigg\}.
\end{aligned} \tag{18}
$$

$\mathrm{Subband}_1$ exactly covers the bandwidth of the discrete image signal $J[u,v,w]$ [see Fig. 2(a)]; (13) becomes

$$
\begin{aligned}
& f_s^d[x,y,z] \\
& \approx \mathcal{DFFT}^{-1}\left\{ J[u,v,w] H_s^{d,\mathrm{Subband}_1}[u,v,w] \right\} \\
& = \mathcal{DFFT}^{-1}\{ J[u,v,w] H_s[u,v,w] \}.
\end{aligned} \tag{19}
$$

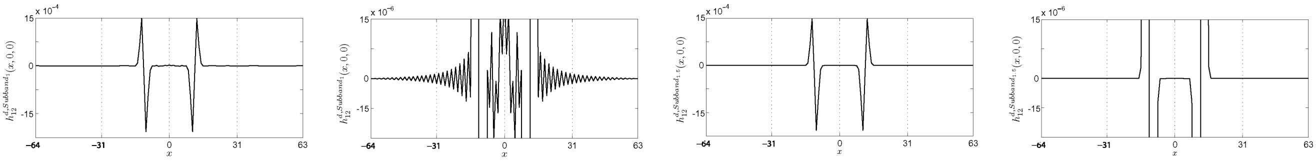
For the validity of the (17) and (19), according to (16), it is imposed that

$$
\sigma \ge \frac{4}{\min_{\{u,v,w\} \in \mathbb{Z}^3 \setminus \mathrm{Subband}_1} (k(u,v,w))}
$$



(a) (b) (c)

Fig. 2. Gray regions represent the coverages of different frequency subbands in the Fourier domain. (a) $\mathrm{Subband}_1$; (b) $\mathrm{Subband}_{1.5}$; (c) $\mathrm{Subband}_2$.



(a) (b) (c) (d)

Fig. 3. Example, which shows that $h_s(x,y,z)$ is discretized with different frequency subbands and with $s = 12, \sigma = 1, y = 0, z = 0$. (a), (b)$h_{12}^{d,\mathrm{Subband}_1}[x,y,z]$, (b) is a zooming of (a); (c), (d) $h_{12}^{d,\mathrm{Subband}_{1.5}}[x,y,z]$, (d) is a zooming of (c).

and $\min_{\{u,v,w\}\in\mathbb{Z}^3\setminus\mathrm{Subband}_1}(k(u,v,w)) = \pi$.[1] Therefore

$$\sigma \geq \frac{4}{\pi} = 1.27. \tag{20}$$

For the calculation of spherical flux, σ is the scale parameter for the Gaussian smoothing of the image $I(x,y,z)$. However, the suggested value of σ in (20) may be large ($\sigma > 1$) for some applications such as detection and segmentation of small blood vessels (1 voxel width vessels). Applying $\mathrm{Subband}_1$ with smaller value of σ can crop the regions that contain large-valued $H_s(u,v,w)$ and, thus, reduce the calculation accuracy of the spherical flux. Fig. 1(b) and (c) shows an example of $H_s(u,0,0)$ with $s = 12$ and $\sigma = 1$, where several ripples of $H_{12}(u,0,0)$ are cropped as they are located outside frequency $\mathrm{Subband}_1$.

To let σ have smaller value (e.g., $\sigma = 1$) without discarding large-valued $H_s(u,v,w)$, we propose to have larger frequency subband to approximate (15). This subband is defined as

$$\begin{aligned}
&\mathrm{Subband}_{1.5}\\
&= \left\{ \left\{ (-N_x + 1 \leq u \leq N_x) \right.\right.\\
&\quad \cap \left(-\left\lceil \frac{N_y}{2} \right\rceil + 1 \leq v \leq \left\lfloor \frac{N_y}{2} \right\rfloor \right)\\
&\quad \left.\cap \left(-\left\lceil \frac{N_z}{2} \right\rceil + 1 \leq w \leq \left\lfloor \frac{N_z}{2} \right\rfloor \right) \right\}\\
&\quad \cup \left\{ \left(-\left\lceil \frac{N_x}{2} \right\rceil + 1 \leq u \leq \left\lfloor \frac{N_x}{2} \right\rfloor \right) \right.\\
&\quad \cap (-N_y + 1 \leq v \leq N_y)\\
&\quad \left.\cap \left(-\left\lceil \frac{N_z}{2} \right\rceil + 1 \leq w \leq \left\lfloor \frac{N_z}{2} \right\rfloor \right) \right\}\\
&\quad \cup \left\{ \left(-\left\lceil \frac{N_x}{2} \right\rceil + 1 \leq u \leq \left\lfloor \frac{N_x}{2} \right\rfloor \right) \right.\\
&\quad \cap \left(-\left\lceil \frac{N_y}{2} \right\rceil + 1 \leq v \leq \left\lfloor \frac{N_y}{2} \right\rfloor \right)\\
&\quad \left.\left.\cap (-N_z + 1 \leq w \leq N_z) \right\} \right\}.
\end{aligned} \tag{21}$$

Fig. 2(b) illustrates the coverage of $\mathrm{Subband}_{1.5}$ in the Fourier domain. Using $\mathrm{Subband}_{1.5}$, $H_s^{d,\infty}$ becomes

$$H_s^{d,\infty} \approx H_s^{d,\mathrm{Subband}_{1.5}}. \tag{22}$$

For $\mathrm{Subband}_{1.5}$, based on the criterion stated in (16), the range of acceptable values of σ is

$$\sigma \geq \frac{4}{\min_{\{u,v,w\}\in\mathbb{Z}^3\setminus\mathrm{Subband}_{1.5}}(k(u,v,w))} \tag{23}$$

where $\min_{\{u,v,w\}\in\mathbb{Z}^3\setminus\mathrm{Subband}_{1.5}}(k(u,v,w)) = \sqrt{2}\pi$[2]

$$\sigma \geq \frac{4}{2\pi\left(\frac{1}{\sqrt{2}}\right)} = \frac{\sqrt{8}}{\pi} = 0.90. \tag{24}$$

Hence, the coverage of $\mathrm{Subband}_{1.5}$ is large enough for the calculation of spherical flux when $\sigma = 1$ and, thus, $H_s^{d,\mathrm{Subband}_{1.5}}[u,v,w]$ is employed in the rest of this paper. In Fig. 3, when approximating $H_s^d[u,v,w]$, the differences between $\mathrm{Subband}_1$ and $\mathrm{Subband}_{1.5}$ are shown. Fig. 3(b) shows that observable artifacts are generated in the function $h_s^{d,\mathrm{Subband}_1}[x,y,z]$ because the coverage of $\mathrm{Subband}_1$ is not large enough to approximate $h_s^{d,\infty}[x,y,z]$. In contrast, there is no similar artifact for $h_s^{d,\mathrm{Subband}_{1.5}}[x,y,z]$ [see Fig. 3(c) and (d)].

For some applications that possibly require smaller value of σ, we suggest devising a valid frequency subband based on the

[1] $\min_{\{u,v,w\}\in\mathbb{Z}^3\setminus\mathrm{Subband}_1}(k(u,v,w)) = \pi$ at the coordinates $(N_x/2,0,0)$, $(0,N_y/2,0)$, $(0,0,N_z/2)$, $(-N_x/2,0,0)$, $(0,-N_y/2,0)$, and $(0,0,-N_z/2)$.

[2] $\min_{\{u,v,w\}\in\mathbb{Z}^3\setminus\mathrm{Subband}_1.5}(k(u,v,w)) = 2\pi/\sqrt{2}$ at the coordinates $(N_x/2,N_y/2,0)$, $(0,N_y/2,N_z/2)$, $(0,N_y/2,N_z/2)$, $(-N_x/2,N_y/2,0)$, $(0,-N_y/2,N_z/2)$, $(0,-N_y/2,N_z/2)$, $(N_x/2,-N_y/2),0)$, $(0,N_y/2,-N_z/2)$, $(0,N_y/2,-N_z/2)$, $(-N_x/2,-N_y/2,0)$, $(0,-N_y/2,-N_z/2)$, and $(0,-N_y/2,-N_z/2)$.

criterion stated in (16). As an example, $\mathrm{Subband}_2$, which supports $\sigma \geq 2\pi^{-1} = 0.64$ is shown in Fig. 2(c). The subscript "2" of $\mathrm{Subband}_2$ represents that this subband encloses doubled bandwidth for all dimensions. Comparing different subbands in Fig. 2(a)–(c), it is observed that the coverage of $\mathrm{Subband}_{1.5}$ is between $\mathrm{Subband}_1$ and $\mathrm{Subband}_2$, and a larger subband coverage supports smaller value of σ.

Finally, using $H_s^{d,\mathrm{Subband}_{1.5}}$ with $\sigma \geq 0.90$, $f_s^d[u,v,w]$ is computed as

$$\begin{aligned} & f_s^d[x,y,z] \\ & \quad \approx \mathcal{DFFT}^{-1}\left\{J[u,v,w]H_s^{d,\mathrm{Subband}_{1.5}}[u,v,w]\right\} \\ & \quad = \mathcal{DFFT}^{-1}\{J[u,v,w](H_s[u,v,w] \\ & \qquad + H_s[u-\mathrm{sign}[u]N_x,v,w] \\ & \qquad + H_s[u,v-\mathrm{sign}[v]N_y,w] \\ & \qquad + H_s[u,v,w-\mathrm{sign}[w]N_z])\} \end{aligned} \tag{25}$$

where

$$\mathrm{sign}[x] = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{otherwise.} \end{cases}$$

5) Reducing the Computation Time by Exploiting Fourier Coefficient Redundancies: Based on the characteristics of $I[x,y,z]$ and $h_s^d[x,y,z]$, we describe two simplifications to further reduce the computation load of the spherical flux calculation in the Fourier domain.

First, since $I[x,y,z]$ and $h_s^d[u,v,w]$ are real signals, almost half of the coefficients of $J[u,v,w]$ and $H_s^d[u,v,w]$ are redundant as $J[u,v,w] \equiv J^*[-u,-v,-w]$ and $H_s^d[u,v,w] \equiv H_s^{d*}[-u,-v,-w]$, where the superscript $*$ represents the conjugate of a complex number. Computation time is shortened by ignoring the redundant coefficients. For the simplicity of the implementation, we treat the coefficients with negative values of w to be redundant. Therefore, (25) is evaluated only when w is non-negative. This halves the number of multiplication operations for evaluating (25).

Second, the function $H_s^d[u,v,w]$ is a spherically symmetric function. The spherical symmetry implies that

$$H_s^d[u,v,w] \equiv H_s^d[|u|,|v|,|w|]. \tag{26}$$

Therefore, evaluating $H_s^d[u,v,w]$ when u, v and w are non-negative is adequate for acquiring $H_s[u,v,w]$ with all values of u, v and w. It facilitates the computation of (25), which is further simplified as[3]

$$\begin{aligned} & f_s^d[x,y,z] \\ & \quad \approx \mathcal{DFFT}^{-1}\left\{J[u,v,w]\left(H_s^d[|u|,|v|,|w|]\right.\right. \\ & \qquad + H_s^d[|u|-N_x,|v|,|w|] + H_s^d[|u|,|v|-N_y,|w|] \\ & \qquad \left.\left. + H_s^d[|u|,|v|,|w|-N_z]\right)\right\}. \end{aligned} \tag{27}$$

With the aid of these two simplifications, the number of times to evaluate $H_s^d[u,v,w]$ is reduced to one-eighth of those without the simplifications and number of multiplication operations taken is halved.

Finally, although the concepts behind the above speed enhancement techniques are based on $\mathrm{Subband}_{1.5}$, they can be applied for various subbands as long as $I[x,y,z]$ and $h_s^d[x,y,z]$ are real signals and $H_s^d[u,v,w]$ is spherically symmetric. As such, they can be tailored for other subbands when different values of σ are needed for the calculation of the spherical flux.

B. Fast Fourier Transform Algorithm in Our Implementation

As previously elaborated, the proposed implementation makes use of the characteristics in the Fourier domain to speed up the calculation of the spherical flux. The fast Fourier transforms (FFT) is a necessity to efficiently compute $J[u,v,w]$ for (15)–(27) and retrieve the calculation result of the spherical flux from the coefficients $F_s^d[u,v,w]$, which is the Fourier representation of $f_s^d[x,y,z]$. FFT algorithms implicitly assume that images are discrete and periodic. The image periodicity causes artifacts appearing along the image boundaries when the spherical flux is computed. However, the artifacts can be greatly reduced by padding additional voxels outside the image to mirror-reflect the image content along the image boundaries prior to the spherical flux computation. The number of padding voxels depends on two values: the radius s and the Gaussian smoothing scale factor σ in $f_s^d[x,y,z]$. In practice, $s + \sigma$ is a good choice for setting the number of padding voxels.

It is well-known that the FFT algorithms are based on the divide-and-conquer strategy. In the standard FFT approach, an input signal is recursively partitioned into two equal halves. This divide-and-conquer strategy imposes a restriction on the dimensions of input images to be power-of-2. Such restriction is overcome by the recent development of the FFT algorithms, for instance, the Cooley-Tukey algorithm [6], the codelet method [11], the prime factor algorithm [19] or the Rader's method [22]. These algorithms permit FFT to be applied on an image with any size and the time complexity is still $O(N \log N)$.

In this work, the FFT routine in our implementation utilizes a publicly available and cross platform library *FFTW* [10], [11], which implements the aforementioned FFT algorithms and supports 3-D input data without the limitation of data size.

C. Multiscale Spherical Flux for Vessel Segmentation

1) Multiscale Spherical Flux: One of the applications of tubular structure analysis is vascular segmentation in angiography. Due to the variation of vessel width, computing the values of the spherical flux in spheres having different scales (or different radii) is needed to precisely detect vessels with different widths as in [27] and [1]. In the conventional spatial implementation, the multiscale spherical flux involves repetitive calculation of spherical flux in different scales, i.e., evaluating (1) using different values of s.

For instance, as described in [27] and [1], a set of scales (radii) is specified by users according to the sizes of target structures. Different values of spherical flux are computed in the set of spheres with the user-defined scales. For each scale, the value of spherical flux is normalized by the surface area of the sphere with the corresponding radius. All values of the normalized spherical flux in different scales are compared to retrieve

[3] In the proposed implementation, each evaluated value of $H_s^d[u,v,w]$ is utilized up to four times, which multiplies with the terms $J[u,v,w]$, $J[u,-v,w]J[-u,v,w]$ and $J[-u,-v,w]$ in (25), where $0 \leq u \leq \lfloor 0.5N_x \rfloor, 0 \leq v \leq \lfloor 0.5N_y \rfloor$ and $0 \leq w \leq \lfloor 0.5N_z \rfloor$.

the maximally magnitude-valued spherical flux, which is called *multiscale spherical flux*, and is given as

$$\frac{f_{m(x,y,z)}(x,y,z)}{4\pi m^2(x,y,z)} \tag{28}$$

where m is the scale among the set of scales (radii of spheres) M that returns the maximally magnitude-valued spherical flux, i.e.,

$$m(x,y,z) = \arg\max_{s\in M} \left|\frac{f_s(x,y,z)}{4\pi s^2}\right|. \tag{29}$$

2) Speed Enhancement in the Computation of Multiscale Spherical Flux: As previously stated, the spherical flux is computed within a set of scales M. It requires $H_s^d[u,v,w]$ to be repetitively evaluated for different values of s. To minimize the computation load for the evaluation of $H_s^d[u,v,w]$, the components of $H_s^d[u,v,w]$, which are independent of scales s, are precalculated and stored in the preprocessing step. These components include the Gaussian term $-4\pi e^{-k^2[u,v,w]\sigma^2/2}$ and the radial component $k[u,v,w]$. They are firstly evaluated and buffered. Their values are retrieved directly upon the evaluation of (27).

IV. Experiments

The proposed implementation was studied using two synthetic and numerical volumes with the size of $180 \times 180 \times 180$ voxels and five phase contrast magnetic resonance angiographic (PC-MRA) image volumes with various sizes. Details of the image volumes are given in the Sections IV-C and IV-D. In this section, the spherical flux computation times and accuracies for both the proposed implementation and the conventional spatial implementation are compared. In all the experiments, the value of σ being used in (9) for the proposed implementation and the Gaussian smoothing process for the conventional spatial implementation was 1 voxel, the set of scales M (radii of spheres) in (29) for both the proposed implementation and the conventional spatial implementation was $1, 2, \ldots, 10$ voxels.

A. Flux Maximizing Geometric Flows

The proposed method is implemented along with a multiscale spherical flux based vessel extraction algorithm, Flux maximizing geometric flows [27] (**FMGF**). The segmentation performances of **FMGF** based on both the proposed implementation and the conventional spatial implementation were examined.

In [27], the multiscale spherical flux (28) is utilized as an evolution speed term to drive an active contour $\mathcal{C}$ for extracting the vasculatures

$$\mathcal{C}_t(x,y,z) = \left(\frac{f_{m(x,y,z)}(x,y,z)}{4\pi m^2(x,y,z)} + \kappa\nabla \cdot \left(\frac{\nabla C(x,y,z)}{|\nabla C(x,y,z)|}\right)\right)\mathcal{N}(x,y,z) \tag{30}$$

where $\mathcal{N}(x,y,z)$ is the outward normal of the contour $\mathcal{C}$ and κ is a weight attached to the mean curvature term. The value of κ was set to 0.03 for all the segmentation experiments. As regions inside vessels have higher intensity than the background regions, the values of $(f_{m(x,y,z)}(x,y,z))/(4\pi m^2(x,y,z))$ are negative. Negative values can produce an expansion effect to the evolving contours inside those vascular regions. In contrast, the evolving contours are shrunk because of the positive values of $(f_{m(x,y,z)}(x,y,z))/(4\pi m^2(x,y,z))$ at the positions near and outside those vessels. As a result, the evolving contours are eventually halted over the vessel boundaries.

Based on [27], a summary for implementing multiscale spherical flux, which is subsequently utilized by **FMGF** (30), is shown in Fig. 4(a). Another summary for the proposed implementation based on (27) is provided in Fig. 4(b). The differences between the proposed implementation and the conventional spatial implementation are highlighted using the gray boxes. The bold boxes represent the crucial steps that have different time complexities, which are $O(Ns^2)$ in the conventional spatial implementation, and $O(N \log N)$ in the proposed implementation.

B. Experiment Setup

In this paper, both the proposed implementation and the conventional spatial implementation were written in C++ and compiled using Microsoft Visual C++. Net 2003 in the Windows XP 32-bit environment. The representation of contours $\mathcal{C}$ of (30) was based on the sparse field levelset framework [28], which was implemented in the Insight-Tool Kit library [13]. The evolution of the levelset function was halted when the per-voxel change of the levelset function accumulated over ten iterations was less than 10^{-5}. The experiments were conducted on a PC with a Pentium IV 3.2-GHz CPU and 1-GB RAM.

C. Synthetic and Numerical Image Volumes

Two synthetic and numerical image volumes were prepared for the validation of the proposed implementation. These image volumes, as shown in Fig. 5, contain tubes and tori having various radii and intensity values. The intensity ranges between 0.6 and 1 for these structures, and is 0 for the background regions. These volumes were generated to mimic the appearance of vasculatures in the PC-MRA image volumes. To study the effect induced by noise, which commonly exists in the clinical images, the synthetic volumes were corrupted with an additive zero mean Gaussian noise having a set of standard deviation values 0.01, 0.02, 0.03, 0.05, and 0.1 (see Fig. 6).

The performance of the proposed implementation was studied and compared with the conventional spatial implementation in three aspects. First, the accuracy of the segmentation and the similarity of the segmentation results obtained by using **FMGF** based on the both implementations. Second, the mean absolute difference between the values of the multiscale spherical flux computed by the both implementations. Third, the times for computing the multiscale spherical flux using the both implementations. To eliminate the boundary effect when evaluating the segmentation accuracy, all accuracies were measured in the center $150 \times 150 \times 150$ voxel regions of the image volumes.

In Table I, the segmentation accuracies are presented in terms of "True positive," "True negative," "False positive," "False negative," "Positive predictive value," and "Negative predictive value." These values were obtained by comparing the ground

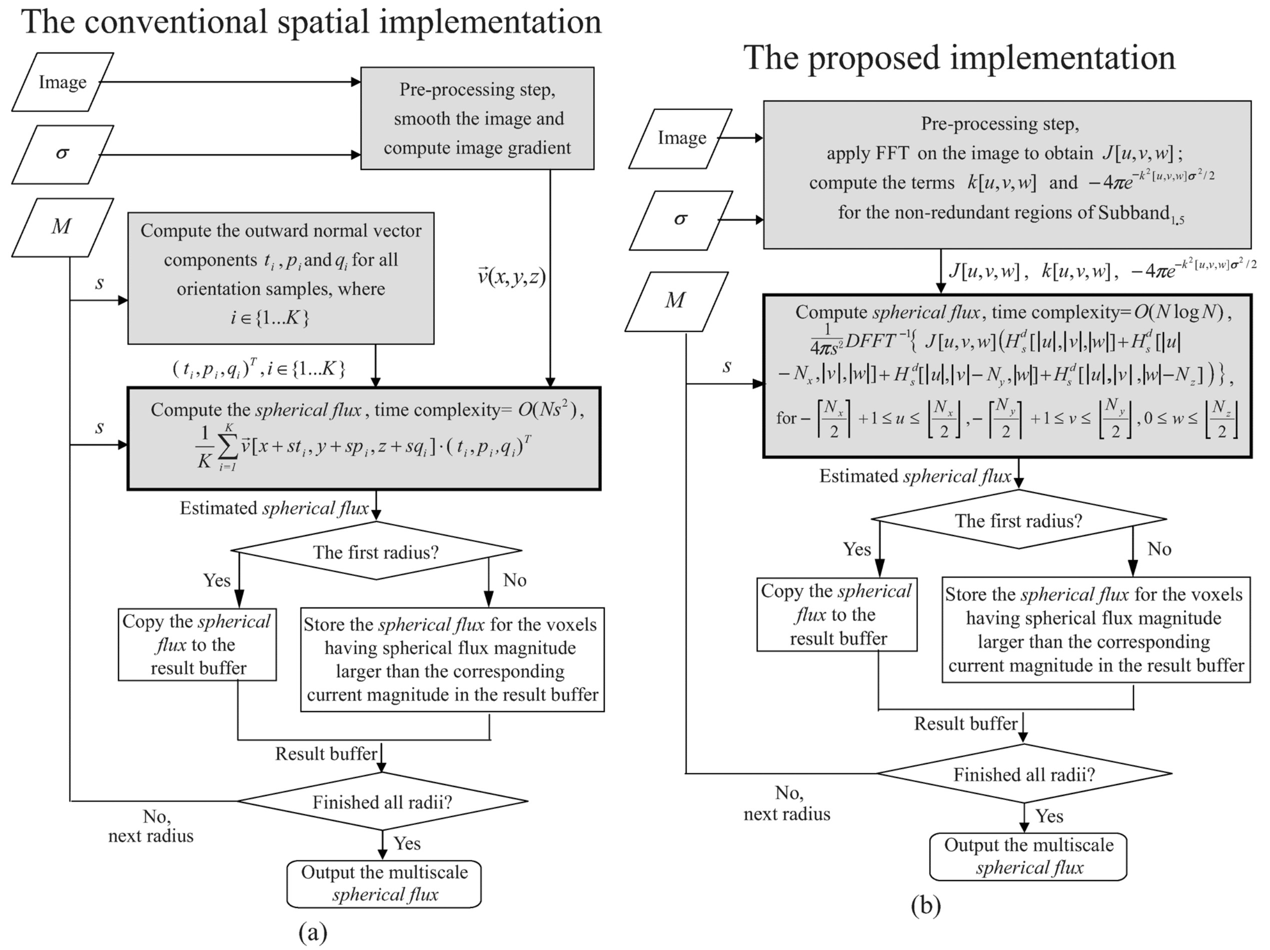

The conventional spatial implementation
Image
σ
Pre-processing step, smooth the image and compute image gradient
M
s
Compute the outward normal vector components t_i, p_i and q_i for all orientation samples, where $i \in \{1...K\}$
$\vec{v}(x,y,z)$
$(t_i,p_i,q_i)^T, i \in \{1...K\}$
s
Compute the *spherical flux*, time complexity= $O(Ns^2)$, $\frac{1}{K}\sum_{i=1}^{K}\vec{v}[x+st_i, y+sp_i, z+sq_i]\cdot(t_i,p_i,q_i)^T$
Estimated *spherical flux*
The first radius?
Yes
No
Copy the *spherical flux* to the result buffer
Store the *spherical flux* for the voxels having spherical flux magnitude larger than the corresponding current magnitude in the result buffer
Result buffer
Finished all radii?
No, next radius
Yes
Output the multiscale *spherical flux*
(a)
The proposed implementation
Image
σ
Pre-processing step, apply FFT on the image to obtain $J[u,v,w]$; compute the terms $k[u,v,w]$ and $-4\pi e^{-k^2[u,v,w]\sigma^2/2}$ for the non-redundant regions of Subband$_{1,5}$
$J[u,v,w]$, $k[u,v,w]$, $-4\pi e^{-k^2[u,v,w]\sigma^2/2}$
M
s
Compute *spherical flux*, time complexity=$O(N\log N)$, $\frac{1}{4\pi s^2}DFFT^{-1}\{J[u,v,w](H_s^d[|u|,|v|,|w|]+H_s^d[|u|-N_x,|v|,|w|]+H_s^d[|u|,|v|-N_y,|w|]+H_s^d[|u|,|v|,|w|-N_z])\}$, for $-\left\lceil\frac{N_x}{2}\right\rceil+1\le u\le\left\lfloor\frac{N_x}{2}\right\rfloor, -\left\lceil\frac{N_y}{2}\right\rceil+1\le v\le\left\lfloor\frac{N_y}{2}\right\rfloor, 0\le w\le\left\lfloor\frac{N_z}{2}\right\rfloor$
Estimated *spherical flux*
The first radius?
Yes
No
Copy the *spherical flux* to the result buffer
Store the *spherical flux* for the voxels having spherical flux magnitude larger than the corresponding current magnitude in the result buffer
Result buffer
Finished all radii?
No, next radius
Yes
Output the multiscale *spherical flux*
(b)

Fig. 4. Flowcharts of the computation of the multiscale spherical flux. (a) The conventional spatial implementation. (b) The proposed implementation. M represents a set of scales. σ represents the scale parameter of the Gaussian function being applied on the image $I(x, y, z)$.

truth with the segmentation results obtained by using **FMGF** based on the proposed implementation and the conventional spatial implementation. Noted that the ground truth was known in the experiments because the image volumes were synthetic. The proposed implementation was also validated by calculating the Dice similarity coefficients (DSC) [30] for the segmentation results obtained by using **FMGF** based on the proposed and conventional spatial implementations. The DSC measure quantifies the similarity between two sets, S_A and S_B as

$$\mathrm{DSC}(S_\mathrm{A}, S_\mathrm{B}) = \frac{2|S_\mathrm{A} \bigcap S_\mathrm{B}|}{|S_\mathrm{A}| + |S_\mathrm{B}|}. \tag{31}$$

The DSC value ranges from 0 to 1, where the DSC value of 1 indicates a perfect agreement between the sets S_A and S_B. As stated in [30], the DSC measure is capable of precisely reporting the similarities between the sets S_A and S_B if $|\bar{S}_\mathrm{A}| \gg |S_\mathrm{A}|$ and $|\bar{S}_\mathrm{B}| \gg |S_\mathrm{B}|$. In vascular segmentation, the nonvessel regions occupy a very large proportion in the entire vascular image volume. Thus, it is illustrative to study the DSC values computed from the segmentation results obtained by using the proposed implementation of **FMGF** and the conventional spatial implementation of **FMGF**. These DSC values are listed in Table I in order to show the similarities between the segmentation results based on the two implementations.

It is observed that, across different noise levels, 0.01, 0.02, 0.03, 0.05, and 0.1, the positive predictive value and the negative predictive value (Table I) are around 88% and 99.9% respectively for tubes, and around 79% and 99.9% respectively for tori (see the segmentation examples in Fig. 7). For both tubes and tori, the differences between the positive predictive values obtained from the proposed implementation and the conventional spatial implementation are very small. Similar observations for the negative predictive values are obtained in Table I. These observations reveal that, using **FMGF**, both the proposed implementation and conventional spatial implementation give very similar segmentation accuracy. In addition, the DSC values listed in Table I for different cases are all above the value of 0.97. Since a DSC value larger than 0.7 generally represents a good agreement [30], the DSC values lisetd in Table I indicate that the segmentation results based on the two implementations for each case are very similar to each other.

To further validate the similarity between the multiscale spherical flux obtained using the proposed implementation and the conventional spatial implementation, the mean absolute difference between two implementations was calculated (the last column of Table II). As the magnitudes of the multiscale spherical flux vary from image to image, for estimating the mean absolute difference, those values were normalized into

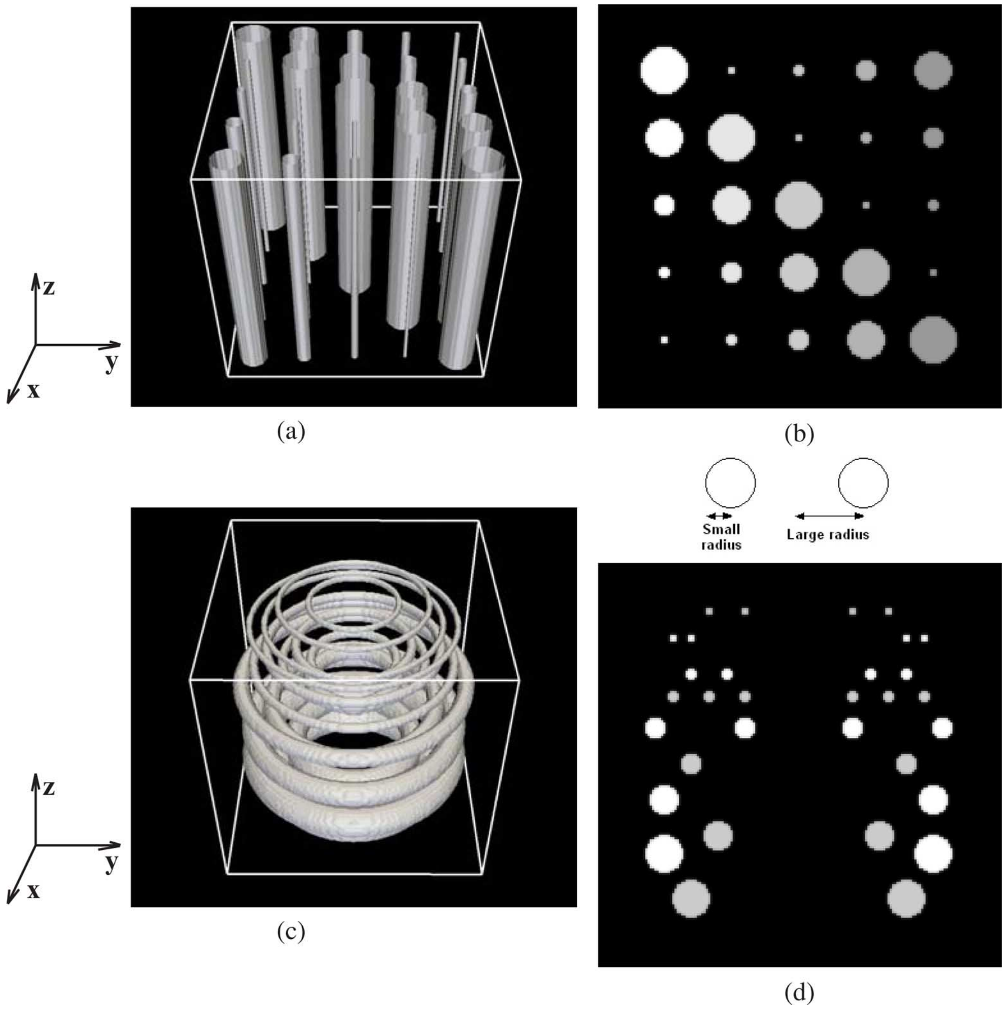


Fig. 5. Synthetic and numerical image volumes. (a) The isosurfaces of the synthetic and numerical tubes. (b) The slice of the tube image volume at $z = 20$. The tubes have radii 1, 2, 4, 6 and 8 voxels, and intensity values 0.6, 0.7, 0.8, 0.9, and 1. (c) The isosurfaces of the synthetic and numerical tori. (d) Top: A vertical cross section of a torus; bottom: the slice of the torus image volume at $y = 90$. The intensity values of the tori are 0.8 for the gray tori and 1 for the white tori. Expressing the sizes of tori in voxels using the representation (small radius, large radius), from top to bottom, the tori have sizes (1, 24), (1, 48), (1, 40), (1, 56), (2, 24), (2, 32), (2, 40), (2, 48), (2, 56), (4, 24), (4, 48), (4, 64), (6, 36), (6, 60), (8, 48), and (8, 60).

a range between -1 and 1. The mean absolute difference is calculated as

$$\mathbf{E}_i \left(\mathbf{E}_{(x,y,z)\in\Omega_i} \left(\left| \frac{S_i(x,y,z)}{\max_{(x,y,z)\in\Omega_i} |S_i(x,y,z)|} - \frac{Q_i(x,y,z)}{\max_{(x,y,z)\in\Omega_i} |Q_i(x,y,z)|} \right| \right) \right) \tag{32}$$

where Ω_i is the center $150 \times 150 \times 150$ voxel region of the ith image volume, S_i and Q_i are the multiscale spherical flux computed by the conventional spatial implementation and the proposed implementation, respectively, and $\mathbf{E}(.)$ is the expected value. As listed in the last column of Table II, the mean absolute difference is 0.0060, which is negligible. Therefore, the proposed implementation is capable of providing the same computation results as obtained in the conventional spatial implementation.

In addition to the good agreement between the computation results and segmentation results obtained by using the proposed implementation and the conventional spatial implementation, the former possesses a significant speed advantage as well. On average, as listed in Table II, the proposed implementation computes the multiscale spherical flux with ten scales for one image volume in around 11 s. The average was estimated by averaging the running times of computing multiscale spherical flux in ten different noise corrupted image volumes (5 for tubes and 5 for tori). The average computation time of the conventional spatial implementation for the same set of image volumes is 19970 s. To illustrate the characteristics of different implementations in computing multiscale spherical flux, the running times of the conventional spatial implementation for different steps (or scales) are listed in Table III and plotted in Fig. 8. The corresponding running times of the proposed implementation are also listed in Table III. Fig. 4 shows the details of the preprocessing steps in Table III for the both implementations. In Fig. 8, the running time of the conventional spatial implementation soars as the scale increases (exact times can be found in Table III). It is because a large scale requires more orientation samples on the spherical boundary surface (see Fig. 9). In contrast, as shown in Table III, the running time of the proposed implementation is

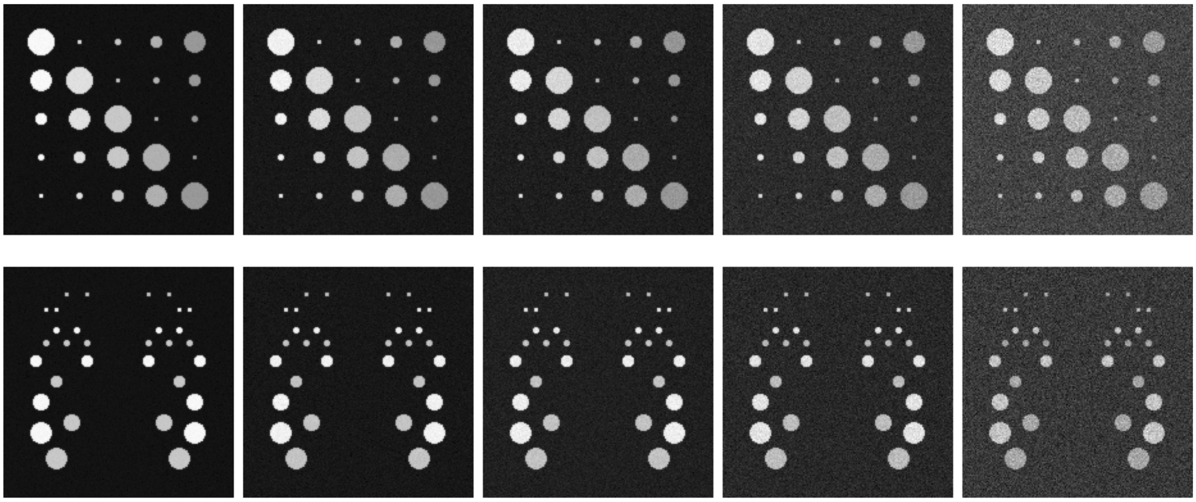


Fig. 6. Top: The image slices of the noise corrupted synthetic and numerical tubes at $z = 20$; bottom: the image slices of the noise corrupted synthetic and numerical tori at $y = 90$. From left to right, the image volumes were corrupted by an additive zero mean Gaussian noise with standard deviations 0.01, 0.02, 0.03, 0.05, and 0.1, respectively.

TABLE I

Segmentation Accuracies of the Synthetic and Numerical Image Volumes, As Shown in Fig. 5, Using the Proposed Implementation Based and the Conventional Spatial Implementation Based Flux Maximizing Geometric Flows. Values Obtained by the Conventional Spatial Implementation Are Enclosed by the Brackets. Values for "True Positive," "True Negative," "False Positive," and "False Negative" are in Voxels

	$\sigma =$	0.01	0.02	0.03	0.05	0.1
Tubes	True positive	504742 (504738)	504412 (504377)	504685 (504341)	504657 (504341)	504346 (504304)
	True negative	2800095 (2799940)	2801176 (2801206)	2800646 (2800985)	2801808 (2800111)	2804712 (2799902)
	False positive	70155 (70310)	69074 (69044)	69604 (69265)	68442 (70139)	65538 (70348)
	False negative	8 (12)	338 (373)	65 (409)	93 (409)	404 (446)
	Positive predictive values	**87.80%** **(87.77%)**	**87.96%** **(87.96%)**	**87.88%** **(87.92%)**	**88.06%** **(87.79%)**	**88.50%** **(87.76%)**
	Negative predictive values	**100.00%** **(100.00%)**	**99.99%** **(99.99%)**	**100.00%** **(99.99%)**	**100.00%** **(99.99%)**	**99.99%** **(99.98%)**
	DSC values	**0.9855**	**0.9846**	**0.9847**	**0.9831**	**0.9830**
Tori	True positive	323681 (323284)	323680 (323582)	323675 (323591)	323677 (323594)	323698 (323615)
	True negative	2963672 (2966166)	2963751 (2966134)	2963608 (2966301)	2963920 (2966424)	2964878 (2966431)
	False positive	87520 (85026)	87441 (85058)	87584 (84891)	87272 (84768)	86314 (84761)
	False negative	127 (224)	128 (226)	133 (217)	131 (214)	110 (193)
	Positive predictive values	**78.72%** **(79.19%)**	**78.72%** **(79.19%)**	**78.70%** **(79.22%)**	**78.76%** **(79.24%)**	**78.95%** **(79.24%)**
	Negative predictive values	**100.00%** **(99.99%)**	**100.00%** **(99.99%)**	**100.00%** **(99.99%)**	**99.99%** **(99.99%)**	**100.00%** **(99.99%)**
	DSC values	**0.9730**	**0.9729**	**0.9735**	**0.9724**	**0.9720**

relatively constant for all scales, except that, when $s = 1$, computation time is slightly shorter than other scales. It is because there is no comparison between the estimated spherical flux and the result buffer, as shown in Fig. 4. It is experimentally shown that the proposed implementation offers significant computation time reduction as compared to the conventional spatial implementation when the value of s grows (see the computation time percentages, which are enclosed by brackets in Table III). In general, it is observed that the proposed implementation utilizes between 0.02% and 0.79% of computation time taken by the conventional spatial implementation for evaluating the spherical flux of an image volume. Regarding the total running times for multiscale spherical flux with ten scales, the proposed implementation needs 0.05% (the forth column in Table II) of the time taken by the conventional spatial implementation.

D. Clinical Cases

There were five clinical phase contrast magnetic resonance angiographic (PC-MRA) image volumes being employed in the experiments. These image volumes were upsampled along the z direction using bilinear interpolation to acquire isotropic

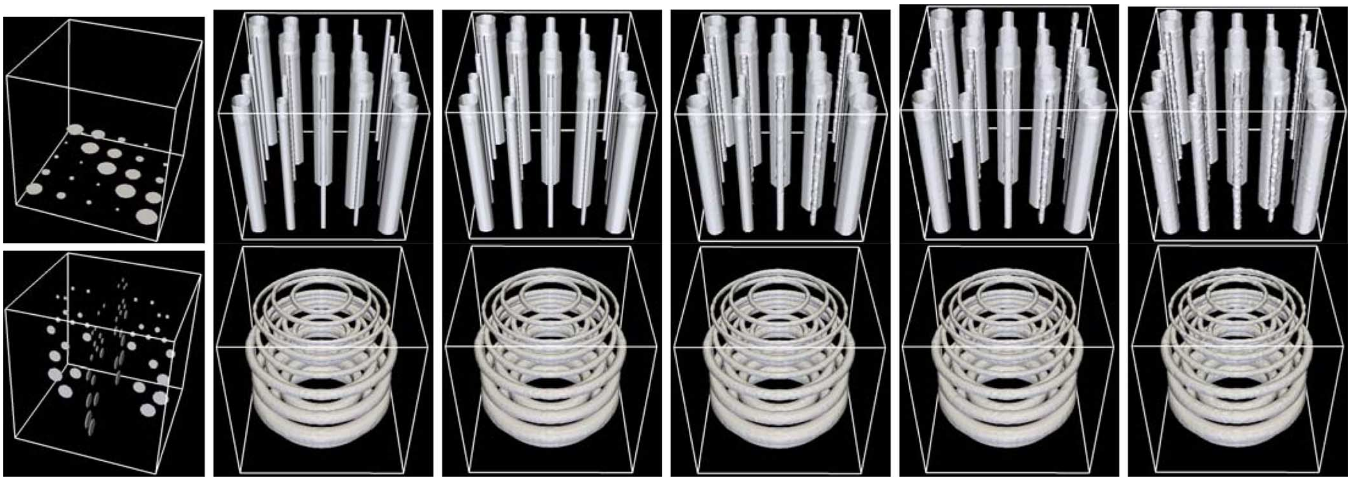


Fig. 7. Top: The isosurfaces of the initial levelset function (the first column) and the segmentation results of noise corrupted synthetic tubes (from the second column to the sixth column). Bottom: The isosurfaces of the initial levelset function (the first column) and the segmentation results of noise corrupted synthetic tori (the second column to the sixth column). From the second column to the sixth column, the segmentation results of the image volumes corrupted by an additive zero mean Gaussian noise with standard deviations 0.01, 0.02, 0.03, 0.05, and 0.1. The results are obtained by applying the flux maximizing geometric flows along with the proposed implementation.

TABLE II

(SECOND AND THIRD COLUMNS): TOTAL RUNNING TIMES (IN SECONDS) FOR MULTISCALE SPHERICAL FLUX COMPUTATION WITH TEN SCALES BASED ON THE PROPOSED IMPLEMENTATION AND THE CONVENTIONAL SPATIAL IMPLEMENTATION. (LAST COLUMN) MEAN ABSOLUTE DIFFERENCE BETWEEN THE MULTISCALE SPHERICAL FLUX VALUES COMPUTED BY BOTH IMPLEMENTATIONS. THE TIMES LISTED ARE OBTAINED BY AVERAGING THE RUNNING TIMES OF COMPUTING MULTISCALE SPHERICAL FLUX IN TEN DIFFERENT NOISE CORRUPTED IMAGE VOLUMES (5 FOR TUBES AND 5 FOR TORI) WHICH WERE GENERATED FOR THE SEGMENTATION EXPERIMENTS ON THE SYNTHETIC AND NUMERICAL IMAGE VOLUMES, AS SHOWN IN FIG. 5. VALUES IN THE SECOND, THIRD, AND LAST COLUMNS ARE ROUNDED TO FOUR SIGNIFICANT DIGITS

	Total running time			Mean absolute
Resolution	Proposed implementation (PI)	Conventional spatial implementation (CSI)	$\frac{\text{PI}}{\text{CSI}} \times 100\%$	difference
$180 \times 180 \times 180$	10.82 seconds	19970 seconds	0.05%	0.0060

TABLE III

RUNNING TIMES (IN SECONDS) OF DIFFERENT STEPS FOR MULTISCALE SPHERICAL FLUX COMPUTATION BASED ON THE PROPOSED IMPLEMENTATION (TOP) AND THE CONVENTIONAL SPATIAL IMPLEMENTATION (BOTTOM). VALUES LISTED IN THE TABLE ARE OBTAINED BY AVERAGING THE RUNNING TIMES OF COMPUTING MULTISCALE SPHERICAL FLUX IN TEN DIFFERENT NOISE CORRUPTED IMAGE VOLUMES (5 FOR TUBES AND 5 FOR TORI) WHICH WERE GENERATED FOR THE SEGMENTATION EXPERIMENTS ON THE SYNTHETIC AND NUMERICAL IMAGE VOLUMES, AS SHOWN IN FIG. 5. VALUES IN THE BRACKETS ARE THE PERCENTAGES, WHICH ARE GIVEN AS $((\text{PI})/(\text{CSI}) \times 100\%)$. VALUES ARE ROUNDED TO FOUR SIGNIFICANT DIGITS, EXCEPT THE VALUES ENCLOSED BY BRACKETS

	Proposed implementation (PI)										
Resolution	Pre-processing	$s = 1$	2	3	4	5	6	7	8	9	10
$180 \times 180 \times 180$	0.9983 (3.05%)	0.9127 (0.79%)	1.014 (0.35%)	1.019 (0.20%)	0.9943 (0.11%)	0.9826 (0.07%)	0.9845 (0.05%)	0.9859 (0.04%)	0.9751 (0.03%)	0.9798 (0.02%)	0.9709 (0.02%)
	Conventional spatial implementation (CSI)										
$180 \times 180 \times 180$	32.77	115.7	289.0	508.4	877.6	1324	1845	2477	3339	4085	5076

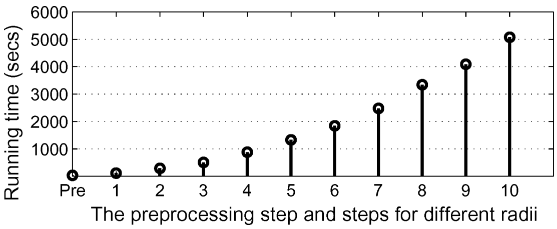


Fig. 8. Plot of the average running times for multiscale spherical flux computation using the conventional spatial implementation against different steps (or scales). The times listed are obtained by averaging the running times of computing multiscale spherical flux in ten different noise corrupted image volumes (5 for tubes and 5 for tori), which were generated for the segmentation experiments on the synthetic and numerical image volumes, as shown in Fig. 5.

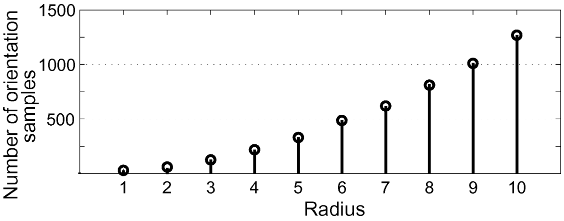
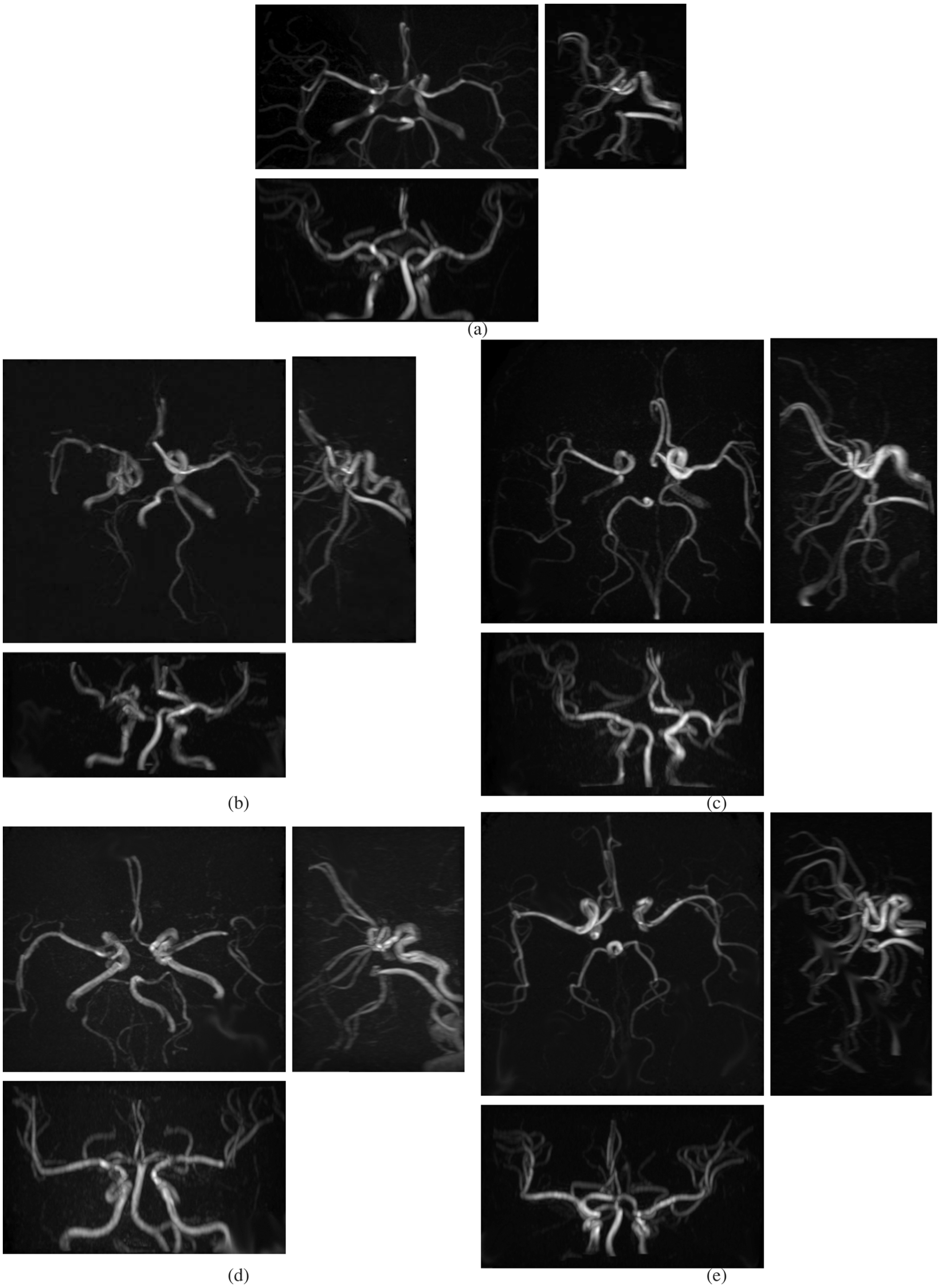


Fig. 9. Number of orientation samples taken for the conventional computation of spherical flux with various radii.

voxels prior to subsequent processing. The maximum intensity projections (MIPs) of these image volumes are shown in Fig. 10(a)–(e). Their resolutions and the voxel sizes are tabulated in Table IV. All the clinical cases are the axial brain scans and were acquired using a Philips 3T ACS Gyroscan MR scanner at the University Hospital of Zurich, Switzerland. The standard PC-MRA imaging protocols were utilized without contrast agents.



(a)

(b)

(c)

(d)

(e)

Fig. 10. Maximum intensity projections of the five clinical PC-MRA image volumes used in the experiments. For each subfigure, the top-left image is the axial view, the bottom image is the coronal view, and the top-right image is the sagittal view.

TABLE IV
RESOLUTIONS AND VOXEL SIZES OF THE IMAGE VOLUMES SHOWN IN FIG. 10

Case	Resolutions	Voxel sizes
a	174 × 296 × 150	0.39mm × 0.39mm × 0.39mm
b	296 × 296 × 130	0.39mm × 0.39mm × 0.39mm
c	276 × 276 × 162	0.37mm × 0.37mm × 0.37mm
d	234 × 270 × 166	0.39mm × 0.39mm × 0.39mm
e	280 × 280 × 162	0.37mm × 0.37mm × 0.37mm

As mentioned in the previous section regarding the synthetic image experiments, the proposed implementation was evaluated by measuring the mean absolute difference between the values of the multiscale spherical flux obtained by using the both implementations and comparing the times for computing the multiscale spherical flux with ten scales using the both implementations in the experiments on the above five clinical PC-MRA image volumes. The computation of the DSC values of the clinical segmentation results is the same as the computation described in the synthetic experiments (31). The mean absolute difference utilized in this section was measured analogous to the synthetic image experiments (32), except that the mean absolute

TABLE V

(SECOND AND THIRD COLUMNS): RUNNING TIMES (IN SECONDS) FOR MULTISCALE SPHERICAL FLUX COMPUTATION BASED ON THE PROPOSED IMPLEMENTATION AND THE CONVENTIONAL SPATIAL IMPLEMENTATION. (FIFTH COLUMN): MEAN ABSOLUTE DIFFERENCES BETWEEN THE SPHERICAL FLUX VALUES COMPUTED BY THE CONVENTIONAL SPATIAL IMPLEMENTATION AND THE PROPOSED IMPLEMENTATION. (LAST COLUMN) ESTIMATED DSC VALUES BASED ON THE SEGMENTATION RESULTS OBTAINED BY USING THE PROPOSED IMPLEMENTATION AND THE CONVENTIONAL SPATIAL IMPLEMENTATION. THE VALUES IN THE SECOND, THIRD, FIFTH, AND LAST COLUMNS ARE ROUNDED TO FOUR SIGNIFICANT DIGITS

	Total running times			Mean absolute	
Resolutions	Proposed implementation (PI)	Conventional spatial implementation (CSI)	$\frac{\text{PI}}{\text{CSI}} \times 100\%$	differences	DSC values
$174 \times 296 \times 150$	27.53	24240	0.11%	0.0042	0.9353
$296 \times 296 \times 130$	40.77	36460	0.11%	0.0020	0.9551
$276 \times 276 \times 162$	43.83	40610	0.11%	0.0033	0.9460
$234 \times 270 \times 166$	45.72	34970	0.13%	0.0038	0.9502
$280 \times 280 \times 162$	23.66	43130	0.05%	0.0049	0.9327
Average			0.10%	0.0036	0.9439

TABLE VI

RUNNING TIMES (IN SECONDS) OF DIFFERENT STEPS FOR MULTISCALE SPHERICAL FLUX COMPUTATION BASED ON THE PROPOSED IMPLEMENTATION (TOP) AND THE CONVENTIONAL SPATIAL IMPLEMENTATION (BOTTOM). VALUES IN THE BRACKETS ARE THE PERCENTAGES, WHICH ARE GIVEN AS $((\text{PI})/(\text{CSI}) \times 100\%)$. VALUES ARE ROUNDED TO FOUR SIGNIFICANT DIGITS, EXCEPT THE VALUES ENCLOSED BY BRACKETS

	Proposed implementation (PI)										
Resolutions	Pre-processing	$s = 1$	2	3	4	5	6	7	8	9	10
$174 \times 296 \times 150$	2.391	2.359	2.625	2.594	2.593	2.547	2.516	2.484	2.469	2.484	2.469
	(7.08%)	(1.64%)	(0.73%)	(0.41%)	(0.24%)	(0.16%)	(0.11%)	(0.08%)	(0.06%)	(0.05%)	(0.04%)
$296 \times 296 \times 130$	3.500	3.407	3.687	3.687	3.688	3.640	3.688	3.734	3.844	3.921	3.969
	(7.31%)	(1.59%)	(0.69%)	(0.39%)	(0.23%)	(0.15%)	(0.11%)	(0.08%)	(0.06%)	(0.05%)	(0.04%)
$276 \times 276 \times 162$	3.969	3.750	4.047	4.078	4.046	4.016	4.000	4.015	3.969	3.984	3.953
	(6.50%)	(1.55%)	(0.68%)	(0.39%)	(0.22%)	(0.15%)	(0.11%)	(0.08%)	(0.06%)	(0.05%)	(0.04%)
$234 \times 270 \times 166$	4.047	3.922	4.172	4.203	4.172	4.172	4.312	4.125	4.204	4.140	4.250
	(14.06%)	(1.90%)	(0.8%)	(0.46%)	(0.27%)	(0.18%)	(0.13%)	(0.10%)	(0.07%)	(0.06%)	(0.05%)
$280 \times 280 \times 162$	2.172	1.891	2.219	2.234	2.203	2.188	2.172	2.156	2.125	2.156	2.141
	(2.83%)	(0.74%)	(0.35%)	(0.20%)	(0.12%)	(0.08%)	(0.05%)	(0.04%)	(0.03%)	(0.02%)	(0.02%)
	Conventional spatial implementation (CSI)										
$174 \times 296 \times 150$	33.79	143.9	359.1	634.8	1090	1630	2257	3048	4022	4936	6090
$296 \times 296 \times 130$	47.88	214.0	535.2	942.5	1619	2426	3349	4518	6093	7483	9236
$276 \times 276 \times 162$	61.02	241.5	598.6	1049	1804	2707	3736	5035	6781	8327	10270
$234 \times 270 \times 166$	28.78	206.2	515.9	906.7	1555	2332	3215	4331	5842	7178	8859
$280 \times 280 \times 162$	76.82	254.3	634.5	1116	1915	2872	3961	5339	7200	8852	10910

difference was estimated individually for each clinical image volume. Thus, the mean absolute difference is given as

$$\mathbf{E}_{(x,y,z)\in\Omega}\left(\left|\frac{S(x,y,z)}{\max_{(x,y,z)\in\Omega}|S(x,y,z)|} - \frac{Q(x,y,z)}{\max_{(x,y,z)\in\Omega}|Q(x,y,z)|}\right|\right)$$

where Ω is the whole image domain excluding the positions, which have distances away from the image boundary within 15 voxels to eliminate the boundary effect when evaluating the differences.

In the fifth column of Table V, the mean absolute differences between the multiscale spherical flux obtained in the both implementations for each clinical case are listed. Among all the clinical cases, the mean absolute differences are only ranged between 0.0020 and 0.0049. Similar to the previous synthetic image experiments, such differences are very small. As such, the proposed implementation can give comparable results of the multiscale spherical flux computation, as compared with the results obtained in the conventional spatial implementation.

More importantly, the computation time of multiscale spherical flux is sharply reduced by using the proposed implementation, as listed in the second and the third columns of Table V. For the proposed implementation, the computation times of multiscale spherical flux for the five clinical cases are around 28, 41, 44, 46, and 24 s. Comparing to the computation times for the same set of image volumes using the conventional spatial implementation, which are 24240, 36460, 40610, 34970, and 43130 s, respectively, the proposed implementation utilizes 0.10% of the computation time taken by the conventional spatial implementation on average (the forth column of Table V). In Table VI, the computation times required by the both implementations for different steps (or scales) are listed. The computation time of the conventional spatial implementation shoots up as s increases while the proposed implementation has fairly consistent computation times among all scales. Thus, the amount of computation time reduction gained by the proposed implementation becomes more significant when the value of s is large (see the computation time percentages, which are enclosed by brackets in Table VI).

On the other hand, the vasculatures in the clinical image volumes extracted by **FMGF** based on multiscale spherical flux computed by the proposed implementation and the conventional spatial implementation are shown in Figs. 11(a)–(e) and 12(a)–(e), respectively. The initial levelset functions for these cases were generated by a global thresholding scheme. In this global thresholding scheme, the regions having the highest 0.5% negative values of the multiscale spherical flux of each volume were selected. The selected regions were evolved under the pure curvature flows with $\kappa = 0.2$ and then utilized as the initial seed points for **FMGF**. This thresholding scheme ensures all the initial seed points were located inside the vessels. The

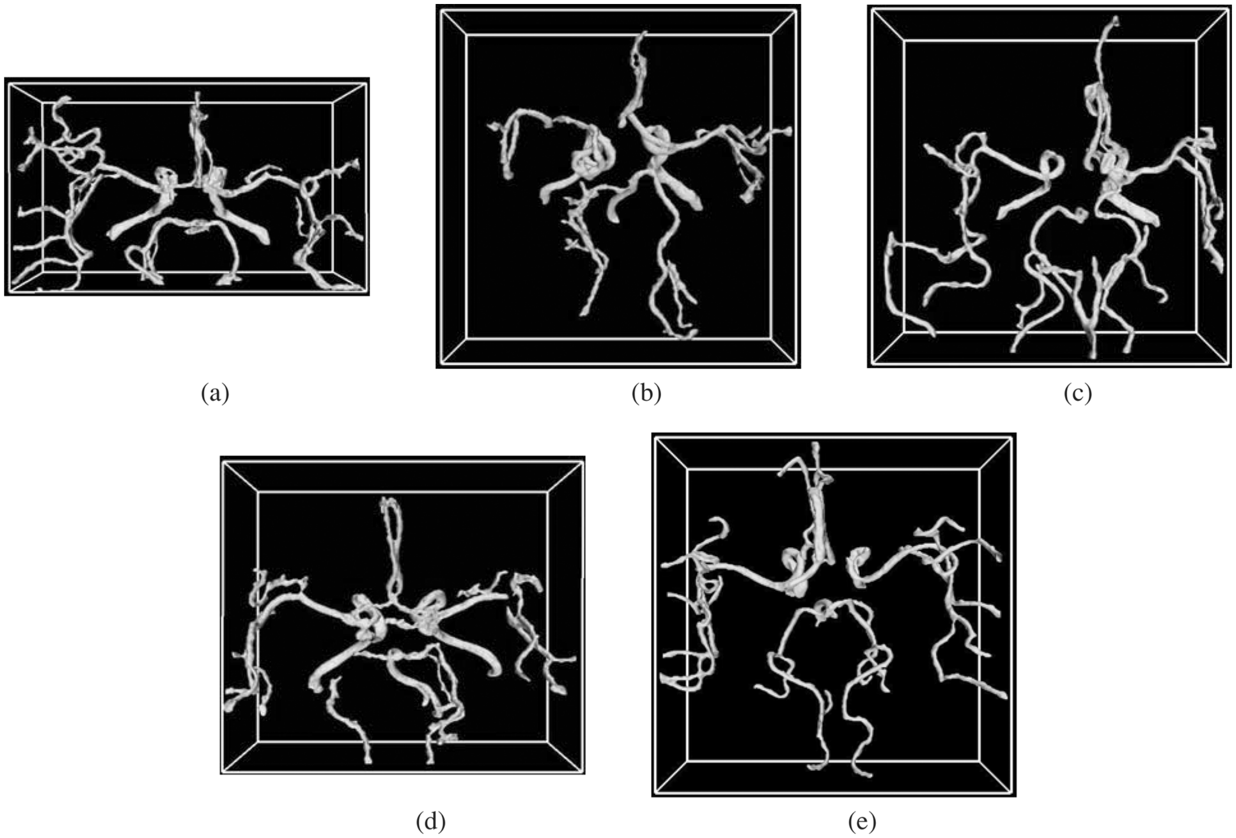


Fig. 11. Axial views of the segmentation results of the five clinical PCMRA image volumes based on **FMGF** using the proposed implementation. The corresponding maximum intensity projections are shown in Fig. 10(a)–(e).

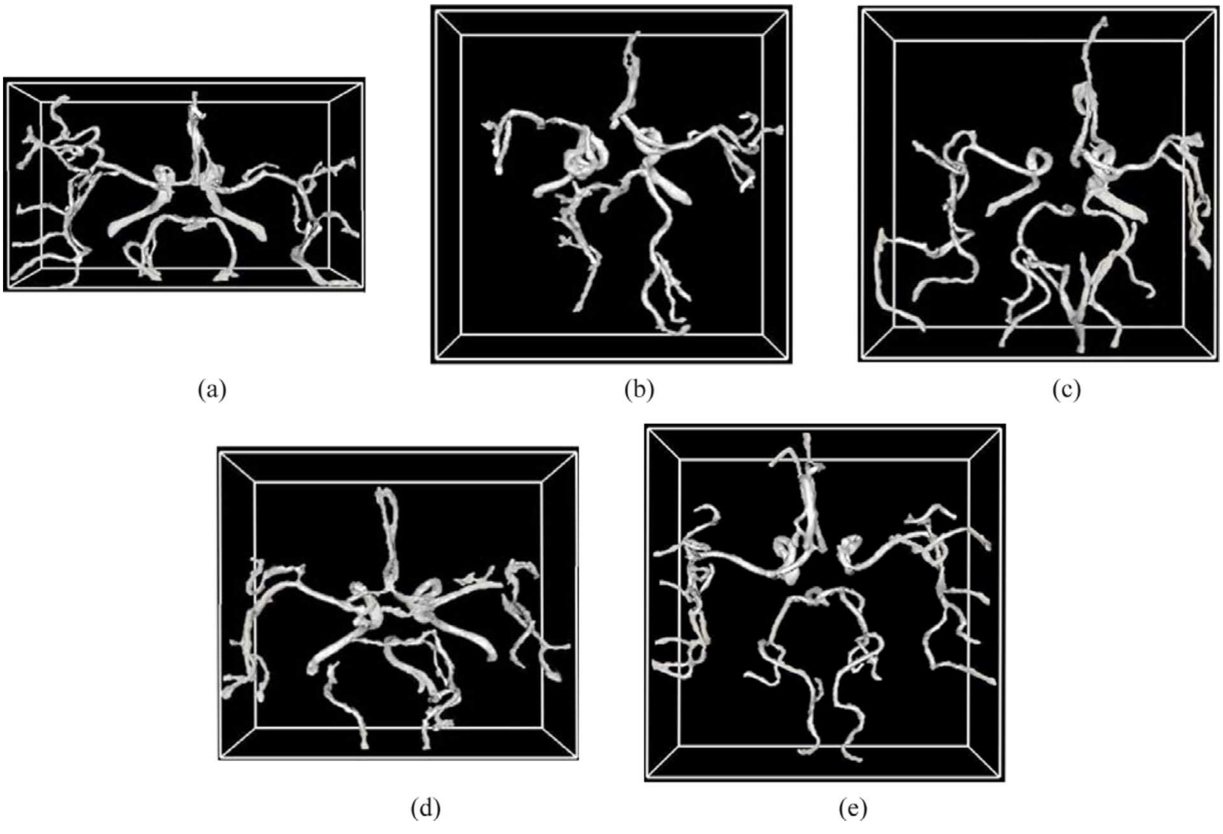


Fig. 12. Axial views of the segmentation results of the five clinical PCMRA image volumes based on **FMGF** using the conventional implementation. The corresponding maximum intensity projections are shown in Fig. 10(a)–(e).

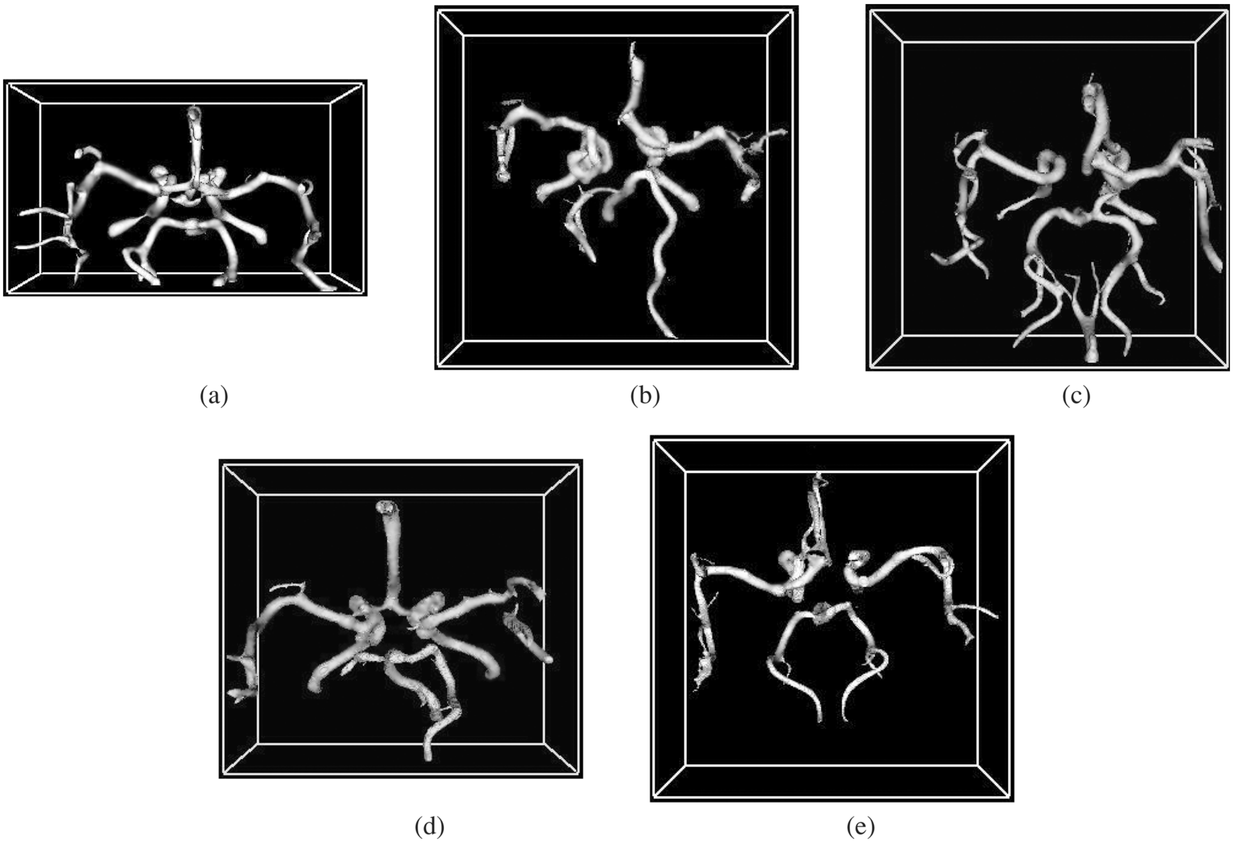


Fig. 13. Axial views of the vessel extraction results of the five clinical PCMRA image volumes based on the vesselness measure, proposed by Frangi *et al.* [9]. The corresponding maximum intensity projections are shown in Fig. 10(a)–(e).

segmentation results shown in Figs. 11(a)–(e) and 12(a)–(e) illustrate that the proposed implementation of **FMGF** is capable of delivering similar segmentation results obtained by using the conventional spatial implementation. This observation is further validated by the DSC values presented in last column of Table V. In the table, the DSC values indicating the segmented region similarities between the proposed implementation of **FMGF** and the conventional spatial implementation of **FMGF** for all five cases are at least 0.9327, which are all above the DSC value of 0.7 for good agreement between results, as suggested in [30].

Finally, the segmentation results of **FMGF** were compared with the results obtained by using a widely used vessel extraction method based on vesselness measures, which was proposed by Frangi *et al.* [9]. The scale set used by the vesselness measure was the same as the scale set used by the multiscale spherical flux, which was $1, 2, \ldots, 10$ voxels. The other parameters used by the vesselness measure were specified strictly according to the descriptions in [9]. In the literature, similar to other works [9], [26], the vessel extraction results were obtained by firstly thresholding the multiscale vesselness responses. In this paper, 0.005 was used as the threshold value for all five clinical cases. This threshold value was acquired manually so that no excessive widening of main vessels was observed. To eliminate noisy regions in the thresholding results, the thresholded regions which were not connected with the positions having the highest 0.5% vesselness responses were removed in the final vessel extraction results. The vesselness extraction results are shown in Fig. 13. It is observed that there are many high curvature small vessels missing in the vesselness extraction results [Figs. 13(a)–(e)]. It is because the high curvature small vessels deviate from the assumption of vessels made by the vesselness measure that vessels are mainly elongated tubular shapes with circular cross sections. In contrast, the **FMGF** does not rely on this assumption for detecting vasculatures, and, thus, it is capable of discovering the high curvature small vessels [see Figs. 11(a)–(e) and 12(a)–(e)]. Comparing the extracted vessels of **FMGF** [Figs. 11(a)–(e) and 12(a)–(e)] with the axial views of the corresponding image volumes [the upper-left images of Fig. 10(a)–(e)], **FMGF** is able to deliver promising vascular segmentation results. With the aid of the proposed implementation, which remarkably reduces the computation time of multiscale spherical flux to 0.10% of the time taken by the conventional spatial implementation, the proposed implementation is a good complement of **FMGF**.

V. Conclusion

We have presented a new and efficient implementation for computing the spherical flux in an image volume. The conventional spatial implementation has two major limitations that adversely affect its usefulness in tubular structure analysis, e.g., detection and segmentation of blood vessels in brain. First, the conventional spatial implementation is not computationally efficient in the spatial domain. Second, the computation time increases quadratically with the sphere size. Our implementation overcomes these limitations by reformulating the spherical flux computation so that most of the computations are performed in the Fourier domain. We have also presented a general scheme for selecting frequency subband while maintaining high computation quality as compared with the conventional spatial implementation.

Using two synthetic image volumes with the size of $180 \times 180 \times 180$ voxels (straight tubes and tori), we have shown experimentally that our proposed implementation can achieve significant computational gain. For computing ten scales in the multiscale spherical flux, the proposed implementation needs 0.05% of the running time taken by the conventional spatial implementation, assuming that one orientation sample is taken for each unit area in the unit of voxel-length on the spherical region boundary. Given that the multiscale spherical flux is normalized between -1 and 1, our proposed implementation can give comparable computation accuracy with mean absolute difference 0.0060 between the both implementations. Our implementation has been tested on five clinical phase contrast magnetic resonance angiographic (PC-MRA) image volumes with sizes of $174 \times 296 \times 150$, $296 \times 296 \times 130$, $276 \times 276 \times 162$, $234 \times 270 \times 166$, $280 \times 280 \times 162$ voxels. It is found that, on average, our proposed implementation needs 0.10% of the running time taken by the conventional spatial implementation, and can also give comparable accuracy with mean absolute difference 0.0036.

With the recent rapid technological advances in the minimally invasive surgery and endovascular treatments, there is a growing need to perform analysis of tubular structures in magnetic resonance angiography and computed tomographic angiography, e.g., detection and segmentation of blood vessels in brain. We believe that our implementation will benefit the tubular structure analysis and can contribute in improving the surgery and treatment efficacy.

APPENDIX

A. Fourier Expression of the Spherical Step Function

The Fourier expression of a spherical step function $d_s(x,y,z)$ with radius s is given as

$$\begin{aligned} D_s(u,v,w) &= \int_0^{N_x}\int_0^{N_y}\int_0^{N_z} d_s(x,y,z) \\ &\times \quad \exp\left(-j2\pi\left(\frac{xu}{N_x}+\frac{yv}{N_y}+\frac{zw}{N_z}\right)\right)dxdydz. \end{aligned}$$

We use spherical coordinates to simplify the integrations, $x = r\cos\alpha\cos\beta$, $y = r\sin\alpha\cos\beta$, $z = r\cos\alpha$, $2\pi u/N_x = k(u,v,w)\sin\phi(u,v,w)\cos\gamma(u,v,w)$, $2\pi v/N_y = k(u,v,w)\sin\phi(u,v,w)\cos\gamma(u,v,w)$ and $2\pi w/N_z = k(u,v,w)\cos\phi(u,v,w)$. Therefore

$$\begin{aligned} D_s(u,v,w) &= \int_0^s \left[\int_{-\pi}^0\int_0^{2\pi}\right. \\ &\times \exp\Big\{-jrk(u,v,w)(\cos\phi(u,v,w)\cos\alpha \\ &+\sin\phi(u,v,w)\sin\alpha\cos(\beta-\gamma(u,v,w)))\Big\} \\ &\times \left. r^2\sin\alpha d\alpha d\beta\right] dr. \end{aligned}$$

Since the sphere function d_s is independent of α and β, so d_s is spherically symmetric. We solve the integrations of α and β by applying 3-D Hankel transforms [4]

$$\begin{aligned} D_s(u,v,w) &= \int_0^s \left[\int_{-\pi}^0\int_0^{2\pi}\right. \\ &\times \exp\{-jrk(u,v,w)(\cos\phi(u,v,w)\cos\alpha \\ &+\sin\phi(u,v,w)\sin\alpha\cos(\beta-\gamma(u,v,w)))\} \\ &\times \left. r^2\sin\alpha d\alpha d\beta\right] dr, \\ &= \frac{2\pi}{\sqrt{k(u,v,w)}}\int_0^s B_{\frac{1}{2}}(srk(u,v,w))r^{\frac{3}{2}}dr, \\ &= \frac{4\pi\sin(sk(u,v,w)) - 4\pi sk(u,v,w)\cos(sk(u,v,w))}{k^3(u,v,w)} \end{aligned}$$

where B_v is the vth order Bessel function of the first kind.

REFERENCES

[1] M. Audette and K. Chinzei, "The application of embedded and tubular structure to tissue identification for the computation of patient-specific neurosurgical simulation models," in *Proc. Int. Symp. Medical Simulation*, 2004, pp. 203–210.

[2] M. Audette, A. Fuchs, O. Astley, Y. Koseki, and K. Chinzei, "Towards patient-specific anatomical model generation for finite element-based surgical simulation," in *Proc. Int. Symp. Surgery Simulation and Soft Tissue Modeling*, 2003, vol. 2643, pp. 320–352.

[3] S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux driven automatic centerline extraction," *Med. Image Anal.*, vol. 9, no. 3, pp. 209–221, 2005.

[4] R. N. Bracewell*, The Fourier Transform and Its Application*. New York: McGraw-Hill, 1986.

[5] B. A. Broche, A. C. Evans, and L. Collins, "A new improved version of the realistic digital brain phantom," *NeuroImage*, vol. 32, no. 1, pp. 138–145, 2006.

[6] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of the complex fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.

[7] M. Descoteaux, M. Audette, K. Chinzei, and K. Siddiqi, "Bone enhancement filtering: Application to sinus bone segmentation and simulation of pituitary surgery," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention*, 2005, pp. 9–16.

[8] M. Descoteaux, L. Collins, and K. Siddiqi, "Geometric flows for segmenting vasculature in mri: Theory and validation," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention*, 2004, pp. 9–16.

[9] A. Frangi, W. Niessen, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention*, 1998, pp. 130–137.

[10] M. Frigo and S. G. Johnson, The FFTW Library [Online]. Available: http://www.fftw.org/

[11] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," in *Proc. IEEE*, 2005, vol. 93, no. 2, pp. 216–231.

[12] M. H. Gazit, D. Goldsher, and R. Kimmel, "Hierarchical segmentation of thin structures in volumetric medical images," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention*, 2003, pp. 562–569.

[13] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, The ITK Software Toolkit.

[14] I. H. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1075–1088, Oct. 2001.

[15] R. Kimmel and A. M. Bruckstein, "Regularized laplacian zero crossings as optimal edge integrators," *Int. J. Comput. Vis.*, vol. 53, pp. 225–243, 2003.

[16] V. Kolmogorov and Y. Boykov, "What metrics can be approximated by geo-cuts, or global optimization of length/area and flux," in *Proc. IEEE Int. Conf. Computer Vision*, 2005, vol. 1, pp. 564–571.

[17] K. Krissian, "Flux-based anisotropic diffusion applied to enhancement of 3-d angiogram," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1440–1442, Nov. 2002.

[18] K. Krissian, C. F. Westin, R. Kikinis, and K. G. Vosburgh, "Oriented speckle reducing anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1412–1424, May 2007.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[20] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[21] S. M. Pizer, K. Siddiqi, G. Szekely, J. N. Damon, and S. W. Zucker, "Multiscale medial loci and their properties," *Int. J. Comput. Vis.*, vol. 55, no. 2–3, pp. 155–179, 2003.

[22] C. M. Rader, "Discrete fourier transforms when number of data samples is prime," *Proc. IEEE*, vol. 56, no. 6, pp. 1107–1108, Jun. 1968.

[23] M. Rochery, I. H. Jermyn, and J. Zerubia, "Higher order active contours," *Int. J. Comput. Vis.*, vol. 69, no. 1, pp. 27–42, 2006.

[24] H. M. Schey, *Div, Grad, Curl, and All That*, 3rd ed. New York: W. W. Norton, 1997.

[25] K. Siddiqi, S. Bouix, A. R. Tannenbaum, and S. W. Zucker, "Hamilton-jacobi skeletons," *Int. J. Comput. Vis.*, vol. 48, no. 3, pp. 215–231, 2002.

[26] M. Sofka and C. V. Stewart, "Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures," *IEEE Trans. Med. Imag.*, vol. 25, no. 12, pp. 1531–1546, Dec. 2006.

[27] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, Dec. 2002.

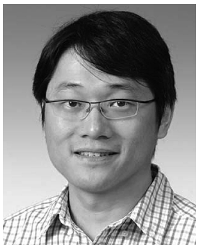
[28] R. Whitaker, "A level-set approach to 3d reconstruction from range data," *Int. J. Comput. Vis.*, vol. 29, no. 33, pp. 203–231, 1998.

[29] Y. Zhao and M. Li, "A hybrid method for vascular segmentation," *IEICE Trans. Inf. Syst.*, vol. E88D, no. 6, pp. 1304–1305, 2005.

[30] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in mr images: Method and validation," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 716–724, Apr. 1994.

Max W. K. Law received the B.Eng. degree in computer engineering and the M.Phil. degree in computer science in 2004 and 2006, respectively, from The Hong Kong University of Science and Technology. He is currently pursuing the Ph.D. degree in computer science and engineering at The Hong Kong University of Science and Technology.

His research interests include medical image processing and analysis, vascular segmentation, medical image registration, edge detection and texture recognition.



Albert C. S. Chung received the B.Eng. degree (first class Honors) in computer engineering from The University of Hong Kong in 1995 and the M.Phil. degree in computer science from The Hong Kong University of Science and Technology in 1998.

He joined the Medical Vision Laboratory, University of Oxford, Oxford, U.K., as a doctoral research student with a Croucher Foundation scholarship and graduated in 2001. He was a Visiting Scientist at the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, in 2001. He is currently an Associate Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. His research interests include medical image analysis, image processing, and computer vision.

Dr. Chung won the 2002 British Machine Vision Association Sullivan Thesis Award for the best doctoral thesis submitted to a U.K. university in the field of computer or natural vision.