

Vessel Segmentation in Medical Imaging Using a Tight-Frame Based Algorithm

Xiaohao Cai, Raymond Chan, Serena Morigi, Fiorella Sgallari

Abstract

Tight-frame, a generalization of orthogonal wavelets, has been used successfully in various problems in image processing, including inpainting, impulse noise removal, super-resolution image restoration, etc. Segmentation is the process of identifying object outlines within images. There are quite a few efficient algorithms for segmentation that depend on the variational approach and the partial differential equation (PDE) modeling. In this paper, we propose to apply the tight-frame approach to automatically identify tube-like structures such as blood vessels in Magnetic Resonance Angiography (MRA) images. Our method iteratively refines a region that encloses the possible boundary or surface of the vessels. In each iteration, we apply the tight-frame algorithm to denoise and smooth the possible boundary and sharpen the region. We prove the convergence of our algorithm. Numerical experiments on real 2D/3D MRA images demonstrate that our method is very efficient with convergence usually within a few iterations, and it outperforms existing PDE and variational methods as it can extract more tubular objects and fine details in the images.

Index Terms

Tight-frame, thresholding, image segmentation, wavelet transform.

I. INTRODUCTION

Segmentation problem of branching tubular objects in 2D and 3D images arises in many applications, for examples, extracting roads in aerial photography, and anatomical surfaces of blood vessels in medical images. In this paper, we are concerned with identifying tube-like structures in Magnetic Resonance Angiography (MRA) images. Because of the necessity to obtain as much fine details as possible in real time, automatic, robust and efficient methods are needed. However, due to the low contrast, intensity inhomogeneity, partial occlusions and intersections, and the presence of noise and possible blur in the images, this segmentation problem is very challenging.

There are many different approaches for vessel segmentation, see for example [16], [20], [23], [26], [31], [34], [42] and the extended reviews [3], [30]. In the following, we concentrate on two approaches that are related to our approach: the active contour approach and the partial differential equation (PDE) approach. For the vessel segmentation algorithms that are based on deformable models [35], because the explicit deformable model representation is usually impractical, level set techniques to

Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China (xhcai@math.cuhk.edu.hk and rchan@math.cuhk.edu.hk).
Department of Mathematics-CIRAM, University of Bologna, Bologna, Italy (morigi@dm.unibo.it and sgallari@dm.unibo.it).

evolve a deformable model have been introduced, and they provide implicit representation of the deformable model. However, the level set segmentation approach is computationally more expensive as it needs to cover the entire domain of interest, which is generally one dimension higher than the original one. Interested readers can refer to recent literature on the level set segmentation strategy for tubular structures [26], [29], [40].

Based on the curve evolution techniques, Mumford-Shah functional and level sets, a new model for active contours to detect objects in a given image was proposed in [16]. Unlike the classical active contour models, this model does not depend on the gradient of the image. In [38], a generalization of the active contour without edges model was proposed for object detection using logic operations. However, both the active contour model [16] and the logic framework [38] are not suitable for detecting tubular structures with low contrast and intensity inhomogeneity, see numerical results in Section IV.

In [23], a geometric deformable model for the segmentation of tubular-like structures was proposed. The model is characterized mainly by two components by using a suitable diffusion tensor: the mean curvature flow and the directionality of the tubular structures. The major advantage of this technique is the ability to segment twisted, convoluted and occluded structures without user interactions; and it can follow the branching of different layers, from thinner to larger structures. The dependence on the grid resolution chosen to solve the discretized PDE model is still an open problem. The authors in [23] have also applied a variant of the proposed PDE model to the challenging problem of composed segmentation in [24].

Besides the methods above, there were some initial work by using wavelets or tight-frames to do texture classification and segmentation [1], [43]. The tight-frame approach is a versatile and effective tool for many different applications in image processing, see [4], [7], [12], [13], [39], [41], [44]. There are many kinds of tight-frame systems, such as those from framelets [36], contourlets [19] and curvelets [10], [11], etc. Recently, the authors in [20] proposed to combine the tight-frame image restoration model of [8] and the total variation based segmentation model of [2], [15], [16] to do segmentation. Their approach results in a minimization problem. In this paper, we also derive a segmentation algorithm that uses the tight-frames. However our method is not based on minimizing any variational model and hence it is different from the method in [20]. In fact, our algorithm iteratively updates the set of possible boundary pixels to change the given image into a binary image. Like the method in [23], our method also has the ability to segment twisted, convoluted and occluded structures. We will show that our algorithm is convergent. In fact, numerical results show that it converges within 10 iterations for 2D as well as 3D Magnetic Resonance Angiography (MRA) images. We will see that our method can extract much more details from the given image than the method in [16], [20] and [23]. We remark that a preliminary version of our segmentation algorithm has been given in our proceeding paper [9]. The main contributions in this paper are: (1) a simpler strategy to initialize and refine the regions enclosing the possible boundary of the vessels, (2) the new strategy leads to a simple proof of convergence and easier choice of parameters, (3) a different tight-frame with better directionally selective property is used to obtain more details in the image, (4) a new 3D result with higher noise and more complicated vessel is added.

The rest of the paper is organized as follows. In Section II, we recall some basic facts about tight-frame and tight-frame algorithms. Our segmentation algorithm is given in Section III. In Section IV we test our algorithm on various real 2D and 3D MRA images and compare it with those representative algorithms from different approaches: PDE-based [23], tight-frame [20] and active contour models [16]. Conclusions are given in Section V.

II. TIGHT-FRAME ALGORITHM

In this section, we briefly introduce the tight-frame algorithm which are based on tight-frame transforms. All tight-frame transforms \mathcal{A} have a very important property, the “*perfect reconstruction property*”: $\mathcal{A}^T \mathcal{A} = \mathcal{I}$, the identity transform, see [36]. Unlike the wavelets, in general, $\mathcal{A}\mathcal{A}^T \neq \mathcal{I}$. For theories of framelets and tight-frame transforms, we refer the readers to [17] for more details. In order to apply the tight-frame algorithm, one only needs to know the filters corresponding to the framelets in the tight-frame. In the followings, we give two examples of tight-frames: the piecewise linear B-spline tight-frame [18] and the dual-tree complex wavelet tight-frame [37].

The filters in the piecewise linear B-spline tight-frame are:

$$h_0 = \frac{1}{4}[1, 2, 1], \quad h_1 = \frac{\sqrt{2}}{4}[1, 0, -1], \quad h_2 = \frac{1}{4}[-1, 2, -1], \quad (1)$$

see [36]. The tight-frame coefficients of any given vector \mathbf{v} corresponding to filter h_i can be obtained by convolving h_i with \mathbf{v} . In matrix terms, we can construct, for each filter, its corresponding filter matrix which is just the Toeplitz matrix with diagonals given by the filter coefficients, e.g. $H_0 = \frac{1}{4}\text{tridiag}[1, 2, 1]$. Then the 1D tight-frame forward transform is given by

$$\mathcal{A} = \begin{bmatrix} H_0 \\ H_1 \\ H_2 \end{bmatrix}. \quad (2)$$

To apply the tight-frame transform onto \mathbf{v} is equivalent to computing $\mathcal{A}\mathbf{v}$, and $H_i\mathbf{v}$ gives the tight-frame coefficients corresponding to the filter h_i , $i = 1, 2, 3$.

The d -dimensional piecewise linear B-spline tight-frame is constructed by tensor products from the 1D tight-frame above, see [21]. For example, in 2D, there are nine filters given by $h_{ij} \equiv h_i^T \otimes h_j$ for $i, j = 1, 2, 3$, where h_i is given in (1). For any 2D image f , the tight-frame coefficients with respect to h_{ij} are obtained by convolving h_{ij} with f . The corresponding forward transform \mathcal{A} will be a stack of nine block-Toeplitz-Toeplitz-block matrices (cf. (2)). The tight-frame coefficients are given by the matrix-vector product $\mathcal{A}\mathbf{f}$, where $\mathbf{f} = \text{vec}(f)$ denotes the vector obtained by concatenating the columns of f .

Dual-tree complex wavelet transform (DCWT) was firstly introduced by Kingsbury [32], [33]. Apart from having the usual perfect reconstruction property, shift-invariance property and linear complexity, it also has the nice directionally selective property at $\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$. The idea is to use two different sets of filters: one gives the real part of the transform and the other gives the imaginary part. Let $\{g_0, g_1\}$ and $\{h_0, h_1\}$ denote the two different sets of orthonormal filters, where g_0, h_0 are the low pass filters and g_1, h_1 are the high pass filters. Let the square matrix $A_{g_i h_j}$ denote the 2D separable wavelet transform implemented using g_i along the rows and h_j along the columns, and define $A_{h_i h_j}, A_{g_i g_j}, A_{h_i g_j}$ similarly, where $i, j = 1, 2$. Then the forward transform for the 2-dimensional DCWT is represented by

$$\mathcal{A}\mathbf{f} := \frac{1}{\sqrt{8}} \begin{bmatrix} I & -I & 0 & 0 \\ I & I & 0 & 0 \\ 0 & 0 & I & I \\ 0 & 0 & I & -I \end{bmatrix} \begin{bmatrix} A_{g_i g_j} \\ A_{h_i h_j} \\ A_{h_i g_j} \\ A_{g_i h_j} \end{bmatrix} \mathbf{f}, \quad i, j = 1, 2.$$

We see that the 2-dimensional DCWT requires four different wavelet transforms in parallel. We refer the readers to [14], [17], [28], [37] and the references therein for more details and for the implementation of 3-dimensional DCWT. In practice, the DCWT are implemented by using different sets of filters for the first level and the remaining levels, see [37]. The filters and the Matlab code for DCWT can be obtained from [14].

The tight-frame algorithms, as given in [4], [5], [6], [7], [13], are of the following generic form:

$$\mathbf{f}^{(i+\frac{1}{2})} = \mathcal{U}(\mathbf{f}^{(i)}), \quad (3)$$

$$\mathbf{f}^{(i+1)} = \mathcal{A}^T \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}^{(i+\frac{1}{2})}), \quad i = 1, 2, \dots \quad (4)$$

Here $\mathbf{f}^{(i)}$ is an approximate solution at the i -th iteration, \mathcal{U} is a problem-dependent operator, and $\mathcal{T}_\lambda(\cdot)$ is the soft-thresholding operator defined as follows. Given vectors $\mathbf{v} = [v_1, \dots, v_n]^T$ and $\lambda = [\lambda_1, \dots, \lambda_n]^T$, $\mathcal{T}_\lambda(\mathbf{v}) \equiv [t_{\lambda_1}(v_1), \dots, t_{\lambda_n}(v_n)]^T$, where

$$t_{\lambda_k}(v_k) \equiv \begin{cases} \text{sgn}(v_k)(|v_k| - \lambda_k), & \text{if } |v_k| > \lambda_k, \\ 0, & \text{if } |v_k| \leq \lambda_k. \end{cases} \quad (5)$$

For how to choose λ_k , see [22].

We remark that (4) performs a tight-frame denoising and smoothing on the image while (3) performs a data-fitting according to the specific problem at hand. For high-resolution image reconstruction problem [13], astronomical infra-red imaging [5], impulse noise removal [6], and inpainting problem [4], the tight-frame algorithm (3)–(4) has been shown to be convergent to a functional with the regularization term being the 1-norm of the tight-frame coefficients, see [7].

III. TIGHT-FRAME BASED ALGORITHM FOR SEGMENTATION

The technology of Magnetic Resonance Angiography (MRA) imaging is based on detection of signals from flowing blood and suppression of signals from other static tissues, so that the blood vessels appear as high intensity regions in the image, see Fig. 1(a). The structures to be segmented are vessels of variable diameters which are close to each other. In general in medical images, speckle noise and weak edges make it difficult to identify the structures in the image. Fortunately, the MRA images contain some properties that can be exploited to derive a good segmentation algorithm. From Fig. 1(a), we see that the pixels near the boundary of the vessels are not exactly of one value, but they are in some range, whereas the values of the pixels in other parts are far from this range. Thus the main idea of our algorithm is to approximate this range accurately. We will obtain the range iteratively by a tight-frame algorithm. The main steps are as follows. Suppose in the beginning of the i th iteration, we are given an approximate image $f^{(i)}$ and a set $\Lambda^{(i)}$ that contains all possible boundary pixels. Then we (i) use $\Lambda^{(i)}$ to estimate an appropriate range $[\alpha_i, \beta_i]$ that contains the pixel values of possible boundary pixels; (ii) use the range to separate the image into three parts—those below the range (background pixels), inside the range (possible boundary pixels), and above the range (pixels in the vessels); (iii) denoise and smooth the inside part by the tight-frame algorithm to get a new image $f^{(i+1)}$. We stop when the image becomes binary and this happens within 10 iterations for the real MRA images we tested, see Table I in Section IV. In the followings, we elaborate each of the steps. Without loss of generality, we assume that the given image f has dynamic range in $[0, 1]$.

Initialization. To start the algorithm at $i = 0$, we need to define $f^{(0)}$, the initial guess, and $\Lambda^{(0)}$, the initial set of possible boundary pixels. Naturally, we set $f^{(0)} = f$, the given image. For $\Lambda^{(0)}$, since we do not have any knowledge of where the

boundary pixels are at the beginning, we identify them by using the gradient of f , i.e. we locate them as pixels where the gradient is bigger than a threshold ϵ . More precisely, let Ω be the index set of all the pixels in the image, then we define

$$\Lambda^{(0)} \equiv \{j \in \Omega \mid \|[\nabla f]_j\|_1 \geq \epsilon\}. \quad (6)$$

Here $[\nabla f]_j$ is the discrete gradient of f at the j th pixel. Once $f^{(0)}$ and $\Lambda^{(0)}$ are defined, we can start the iteration.

Step (i): computing the range $[\alpha_i, \beta_i]$. Given $\Lambda^{(i)}$, we first compute the mean pixel value on $\Lambda^{(i)}$:

$$\mu^{(i)} = \frac{1}{|\Lambda^{(i)}|} \sum_{j \in \Lambda^{(i)}} f_j^{(i)}, \quad (7)$$

where $|\cdot|$ denotes the cardinality of the set and $f_j^{(i)}$ is the pixel value of pixel j in image $f^{(i)}$. Then we compute the mean pixel values of the two sets separated by $\mu^{(i)}$:

$$\mu_-^{(i)} = \frac{1}{|\{j \in \Lambda^{(i)} : f_j^{(i)} \leq \mu^{(i)}\}|} \sum_{\{j \in \Lambda^{(i)} : f_j^{(i)} \leq \mu^{(i)}\}} f_j^{(i)}, \quad (8)$$

and

$$\mu_+^{(i)} = \frac{1}{|\{j \in \Lambda^{(i)} : f_j^{(i)} \geq \mu^{(i)}\}|} \sum_{\{j \in \Lambda^{(i)} : f_j^{(i)} \geq \mu^{(i)}\}} f_j^{(i)}. \quad (9)$$

While $\mu^{(i)}$ reflects the mean energy of the set of possible boundary pixels, $\mu_-^{(i)}$ and $\mu_+^{(i)}$ reflect the mean energies of the pixels on the boundary closer to the background and closer to the vessels respectively. We define

$$\alpha_i \equiv \max \left\{ \frac{\mu^{(i)} + \mu_-^{(i)}}{2}, 0 \right\}, \quad \beta_i \equiv \min \left\{ \frac{\mu^{(i)} + \mu_+^{(i)}}{2}, 1 \right\}. \quad (10)$$

Step (ii): thresholding the image into three parts. Using the range $[\alpha_i, \beta_i] \subseteq [0, 1]$, we can separate the image $f^{(i)}$ into three parts—those below, inside, and above the range, see Fig. 1(b) for $i = 0$. Since our aim is to create a binary image, we threshold those pixel values that are smaller than α_i to 0, those larger than β_i to 1, and those in between, we stretch them between 0 and 1 using a simple linear contrast stretch, see [27]. If there are no pixels in between α_i and β_i , then the threshold image is binary and the algorithm stops. More precisely, let

$$M_i = \max\{f_j^{(i)} \mid \alpha_i \leq f_j^{(i)} \leq \beta_i, j \in \Lambda^{(i)}\}, \quad m_i = \min\{f_j^{(i)} \mid \alpha_i \leq f_j^{(i)} \leq \beta_i, j \in \Lambda^{(i)}\}, \quad (11)$$

then we define

$$f_j^{(i+\frac{1}{2})} = \begin{cases} 0, & \text{if } f_j^{(i)} \leq \alpha_i, \\ \frac{f_j^{(i)} - m_i}{M_i - m_i}, & \alpha_i \leq f_j^{(i)} \leq \beta_i, \\ 1, & \text{if } \beta_i \leq f_j^{(i)}, \end{cases} \quad \text{for all } j \in \Omega. \quad (12)$$

Fig. 1(c) shows the threshold and stretched image from Fig. 1(b), where the yellow pixels are pixels we have classified not on the boundary, i.e. $f_j^{(i+\frac{1}{2})} = 0$ (signifying pixel j is in the background) or $f_j^{(i+\frac{1}{2})} = 1$ (signifying pixel j is inside the vessel). The remaining pixels are remained to be classified and we denote the set by:

$$\Lambda^{(i+1)} = \{j \mid 0 < f_j^{(i+\frac{1}{2})} < 1, j \in \Omega\}. \quad (13)$$

Note that those pixels with values m_i and M_i are mapped to 0 and 1 respectively and hence will not be in $\Lambda^{(i+1)}$. Next we denoise and smooth $f^{(i+\frac{1}{2})}$ on $\Lambda^{(i+1)}$.

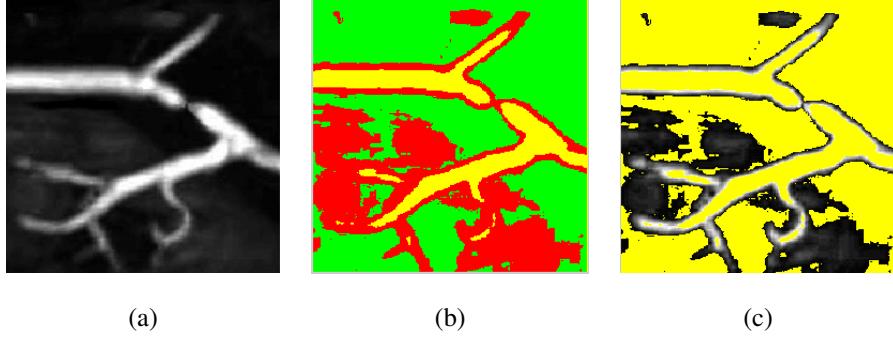


Fig. 1. (a) Given MRA image. (b) Three parts of the given image (green–below, red–in between, and yellow–above). (c) Threshold and stretched image by (12) (yellow pixels are with value 0 or 1).

Step (iii): tight-frame iteration. To denoise and smooth the image $f^{(i+\frac{1}{2})}$ on $\Lambda^{(i+1)}$, we apply the tight-frame iteration (4) on $\Lambda^{(i+1)}$. More precisely, if $j \notin \Lambda^{(i+1)}$, then we set $f_j^{(i+1)} = f_j^{(i+\frac{1}{2})}$; otherwise, we use (4) to get $f_j^{(i+1)}$. To write it out clearly, let $\mathbf{f}^{(i+\frac{1}{2})} = \text{vec}(f^{(i+\frac{1}{2})})$, and $P^{(i+1)}$ be the diagonal matrix where the diagonal entry is 1 if the corresponding index is in $\Lambda^{(i+1)}$, and 0 otherwise. Then

$$\mathbf{f}^{(i+1)} \equiv (I - P^{(i+1)})\mathbf{f}^{(i+\frac{1}{2})} + P^{(i+1)}\mathcal{A}^T\mathcal{T}_\lambda(\mathcal{A}\mathbf{f}^{(i+\frac{1}{2})}). \quad (14)$$

By reordering the entries of the vector $\mathbf{f}^{(i+1)}$ into columns, we obtain the image $f^{(i+1)}$. Note that the effect of (14) is to denoise and smooth the image on $\Lambda^{(i+1)}$, see [7]. Since the pixel values of all pixels outside $\Lambda^{(i+1)}$ are either 0 or 1, the cost of the tight-frame transform in (14), such as the computation of $\mathcal{A}\mathbf{f}^{(i+\frac{1}{2})}$, can be reduced significantly by taking advantage of the fact that all computations can be done only on pixels around $\Lambda^{(i+1)}$.

Stopping criterion. We stop the iteration when all the pixels of $f^{(i+\frac{1}{2})}$ are either of value 0 or 1, or equivalently when $\Lambda^{(i)} = \emptyset$. For the binary image $f^{(i+\frac{1}{2})}$, all the pixels with value 0 are considered as background pixels and pixels with value 1 constitute the tubular structures.

Below we give the full algorithm and show that it always converges to a binary image.

Algorithm 1: Tight-frame algorithm for segmentation

-
1. Input: given image f .
 2. Set $f^{(0)} = f$ and $\Lambda^{(0)}$ by (6)
 3. Do $i = 0, 1, \dots$, until stop
 - (a) Compute $[\alpha_i, \beta_i]$ by (10).
 - (b) Compute $f^{(i+\frac{1}{2})}$ by (12).
 - (c) Stop if $f^{(i+\frac{1}{2})}$ is a binary image.
 - (d) Compute $\Lambda^{(i+1)}$ by (13).
 - (e) Update $f^{(i+\frac{1}{2})}$ to $f^{(i+1)}$ by (14).
 4. Output: binary image $f^{(i+\frac{1}{2})}$.
-

Theorem 1. Our tight-frame algorithm will converge to a binary image.

Proof: From (13), it suffices to prove that $|\Lambda^{(i)}| = 0$ at some finite step $i > 0$. By (11), if $f^{(i+\frac{1}{2})}$ is not yet a binary image, then there will be at least one $j \in \Lambda^{(i)}$ such that $f_j^{(i)} = M_i$. By (12), $f_j^{(i+\frac{1}{2})}$ will be set to 1 and hence by (13), $j \notin \Lambda^{(i+1)}$. Hence $|\Lambda^{(i+1)}| < |\Lambda^{(i)}|$. Since $|\Lambda^{(0)}|$ is finite, there must exist some i such that $|\Lambda^{(i)}| = 0$. ■

We emphasize that the algorithm actually converges within 10 iterations for the 2D and 3D real images we have tested.

Finally, let us estimate the computation cost of our method for a given image with n pixels. Since the costs of computing $\mu^{(i)}$, $\mu_-^{(i)}$, $\mu_+^{(i)}$, and $[\alpha_i, \beta_i]$ are all of $O(n)$ operations, see (7)–(10); and the cost of a tight-frame transform is also linear with respect to n (see e.g. (2) where H_i are all tri-diagonal matrices), we see that the complexity of our algorithm is $O(n)$ per iteration. In fact, one can speed up the computation tremendously as all computations can be done only on pixels around $\Lambda^{(i)}$ and there is no need to carry out the computations in Ω —though in the numerical tests, we did not optimize the code and we just carried out the computations in Ω . According to Table I, after just 3 iterations, the set $\Lambda^{(i)}$ contains only a hundred pixels in the 2D case and just two thousand pixels in the 3D case.

IV. NUMERICAL EXAMPLES

In this section, we try our tight-frame segmentation algorithm (Algorithm 1) on 2D and 3D real MRA images tested in [23] and [24]. Our algorithm is written in Matlab. The thresholding parameters λ_k used in (5) are all chosen to be $\lambda_k \equiv 0.1$; and we choose $\epsilon = 0.003$ and 0.06 for 2D and respectively 3D images in (6). To display the boundary and the surface in the binary images visually, we use the Matlab commands “contour” and “isosurface” for 2D and 3D images respectively. The tight-frame we used is the DCWT downloaded from [14] where all parameters in the code are chosen to be the default values. In particular, the number of wavelet levels used in the code is 4. The 2D images are tested in a MacBook with 2.4 GHz processor and 4GB RAM, while the 3D images, because of their sizes, are tested on a node with 120GB RAM in a PC-cluster. We compare our method with those representative algorithms from different approaches: PDE-based [23], tight-frame [20] and active contour models [16], where the programs are provided by the authors.

Example 1. The test image is a 182×182 MRA image of a carotid vascular system, see Fig. 2(a). The blood vessels contain regions with high and low intensities, including some very thin vessels in the middle with intensities as low as the intensity of the background. Intersections of partial structures even increase the difficulty of the segmentation. The segmentation results by different methods are given in Fig. 2 where the given image is overlaid so that we can compare the accuracy of the methods. Clearly, the results of Figs. 2(b) and (c) are not satisfactory since the vessels obtained in Fig. 2(b) are disconnected and some vessels in Fig. 2(c) are not detected. By comparing the parts inside the rectangles in Fig. 2(d) with those in Fig. 2(e), we see that our method can extract clearer boundary than the method in [23], especially for handling the artifacts near the boundary. We also see that the noise in the image has been removed because of the denoising property of formula (4) as explained in Section II. Our method converges in 5 iterations and requires 0.64 seconds. The first column of Table I gives $|\Lambda^{(i)}|$ at each iteration.

Example 2. The test image is a 256×256 MRA image of a kidney vascular system as shown in Fig. 3(a). This example shows the ability of our method to reconstruct structures which present small occlusions along the coherence direction. Our

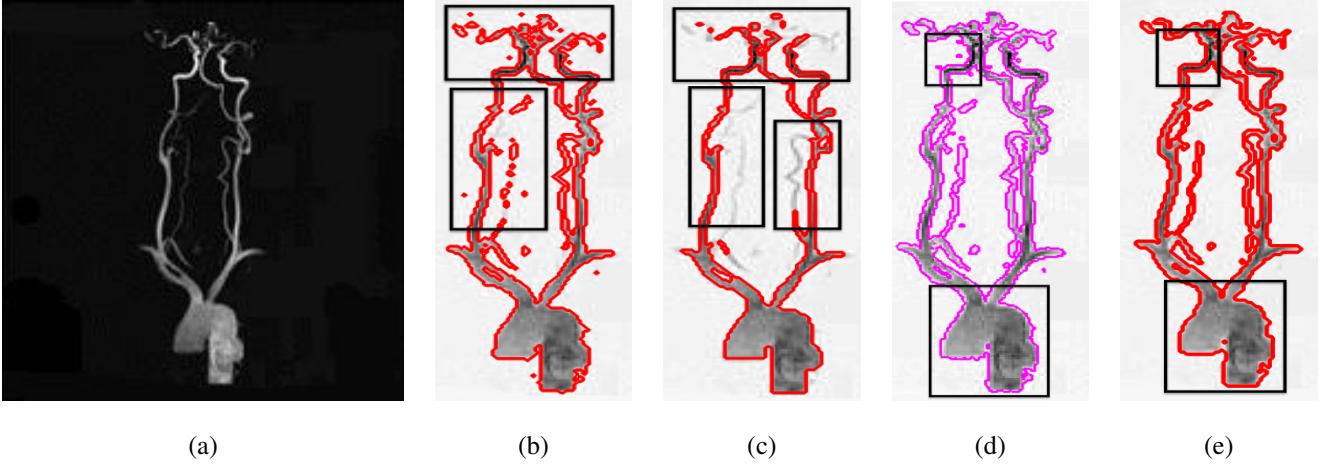


Fig. 2. Carotid vascular system segmentation. (a) Given image. (b), (c) and (d) Results by the methods in [16], [20] and [23] respectively. (e) Result of our method.

TABLE I
CARDINALITY OF $\Lambda^{(i)}$ AT EACH ITERATION OF THE FOUR EXAMPLES

$ \Lambda^{(i)} $	Example 1 $ \Omega = 11284$	Example 2 $ \Omega = 66049$	Example 3 $ \Omega = 8120601$	Example 4 $ \Omega = 6000000$
$i = 0$	1721	9444	137330	152898
$i = 1$	354	1943	32760	32064
$i = 2$	82	464	8795	8565
$i = 3$	26	133	2475	2391
$i = 4$	4	32	689	650
$i = 5$	0	8	189	187
$i = 6$	-	0	56	59
$i = 7$	-	-	15	12
$i = 8$	-	-	3	2
$i = 9$	-	-	0	0

method converges in 6 iterations with 0.78 seconds; and the second column in Table I gives $|\Lambda^{(i)}|$ at each iteration. The results by the method in [16] (Fig. 3(b)) and by the method in [20] (Fig. 3(c)) are not good since they can not recover the small occlusions along the coherence direction, while this can be done by our method and the method in [23], see Figs. 3(d) and (e). Furthermore, our method is better than the method in [23] by comparing the rectangular parts of Fig. 3(d) with those in Fig. 3(e), since our method can detect smoother edges; see Figs. 3(f)–(k) which are the results of zooming in the rectangular parts of Figs. 3(d) and (e) respectively. The results also show that our method is very effective in removing artifacts.

Example 3. This is a 3D example where we extracted a volumetric data set of size $201 \times 201 \times 201$ from a $436 \times 436 \times 540$ CTA (Computed Tomographic Angiography) image of the kidney vasculature system, see Fig. 4(a). Because of different curvatures, diameters, bifurcations, and weak surfaces impaired by noise, it is hard to detect the tips of the thin vessels. Figs. 4(b) and (c) give the results by using the method in [24] and our proposed method respectively. The figure shows that our method can give much more details. A visual comparison of the given image with our result shows that almost all the vessels are correctly segmented. We note that our method converges in 9 iterations, see the third column in Table I.

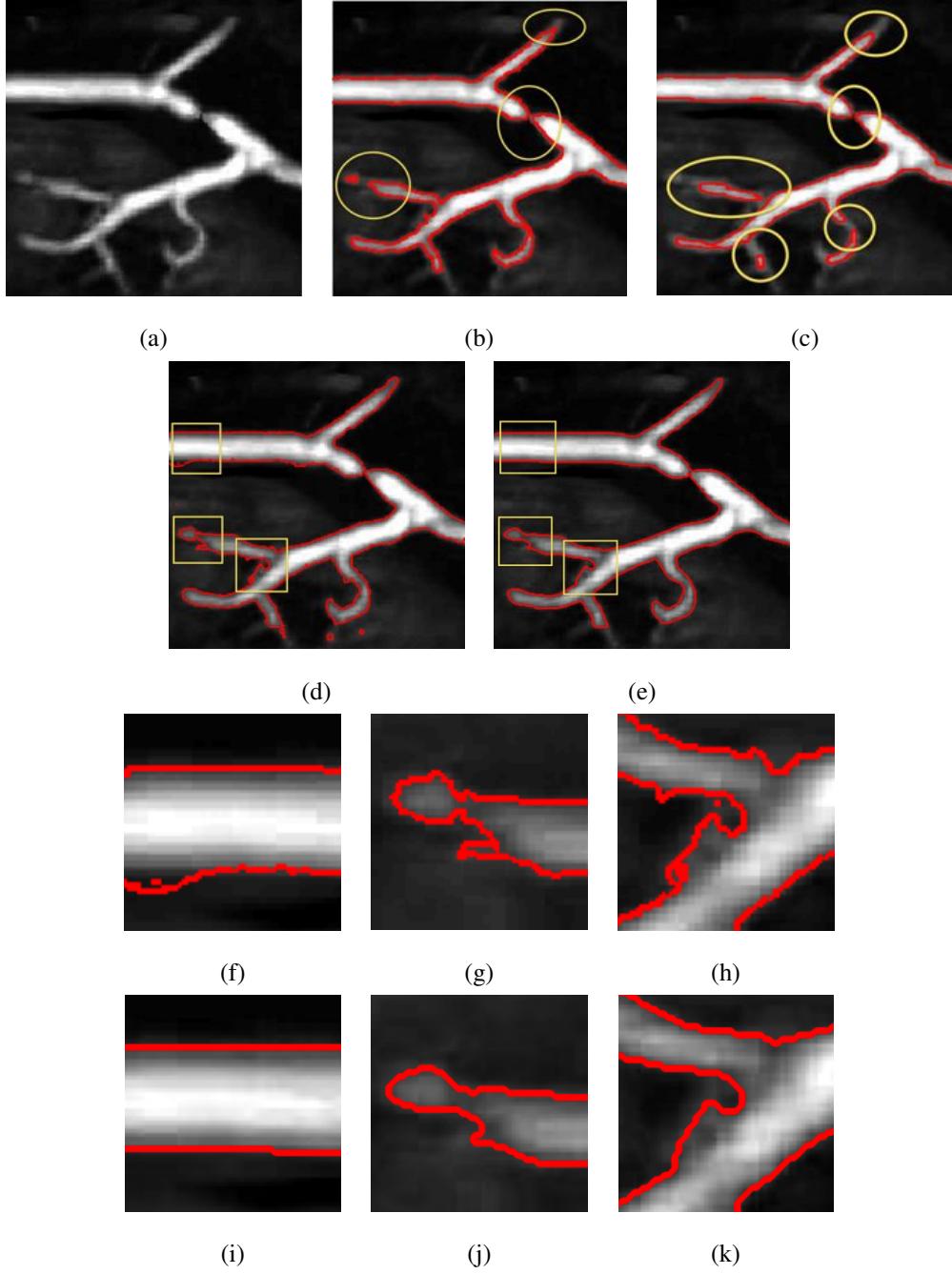


Fig. 3. Kidney vascular system segmentation. (a) Given image. (b), (c) and (d) Results by the methods in [16], [20] and [23] respectively. (e) Results by our method. (f)–(k) are the zoomed-in parts of (d) and (e).

Example 4. Our second 3D example is an MRA data set of a brain aneurysms (vessel wall dilatations). The $120 \times 250 \times 200$ volumetric data set has been extracted from a $120 \times 448 \times 540$ MRA image of the brain-neck vasculature system, see Fig. 5(a). As in Example 3, the different curvatures, diameters and bifurcations of the vessels make it a difficult problem. In addition, the high noise makes the thin vessels hard to see even by the naked eyes. For this topological complex tubular structures, Figs. 5(b) and (c) give the results by the method in [23] and our method respectively. Obviously, our method can segment many more thin vessels, especially those corrupted by the high noise. The results reflect that our new method is very effective in handling the noise spread on the surface of the vessels. For this complicated example, our method converges in 9 iterations only, see

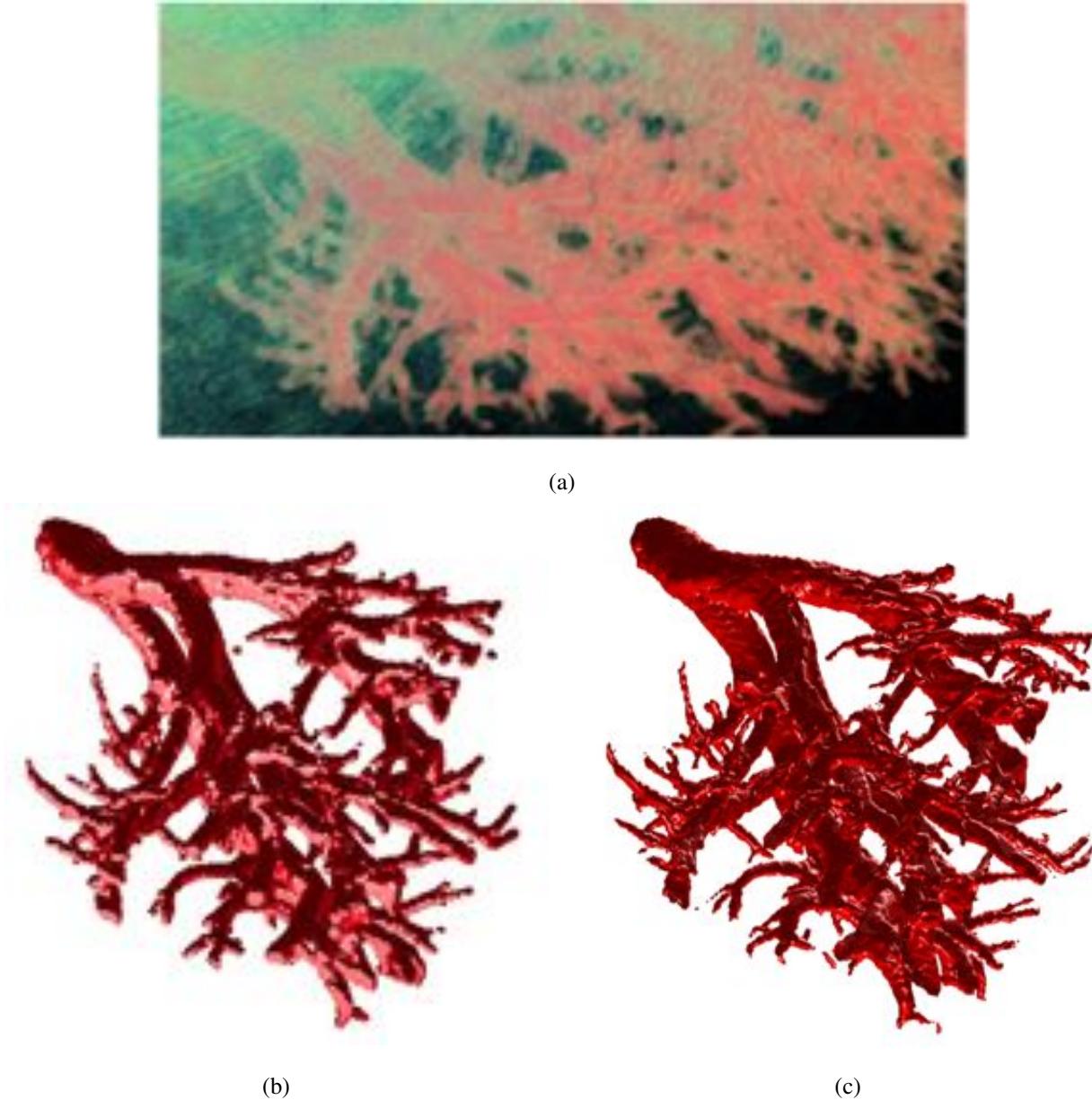


Fig. 4. Segmentation of the kidney volume data set. (a) Given CTA image. (b) Result by the method in [24]. (c) Result by our method.

the fourth column of Table I. Our segmentation technique can be used to compute vessel radii and other clinically useful measurements in case of aneurysms. In Fig. 5(c), one may argue that there are some small isolated points in the image and the surface of the vessels is not smooth. This can easily be remedied by smoothing our final binary image by the tight-frame formula (4) one time before we show the image, since (4) has the denoising property as we explained in Section II. See Fig. 5(d) for the denoised-and-smoothed image after one iteration of (4).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a new and efficient segmentation method based on the tight-frame approach. The numerical results demonstrate the ability of our method in segmenting tubular structures. The method can be implemented fast and give very accurate, smooth boundaries or surfaces. In addition, since the pixel values of more and more pixels will be set to either

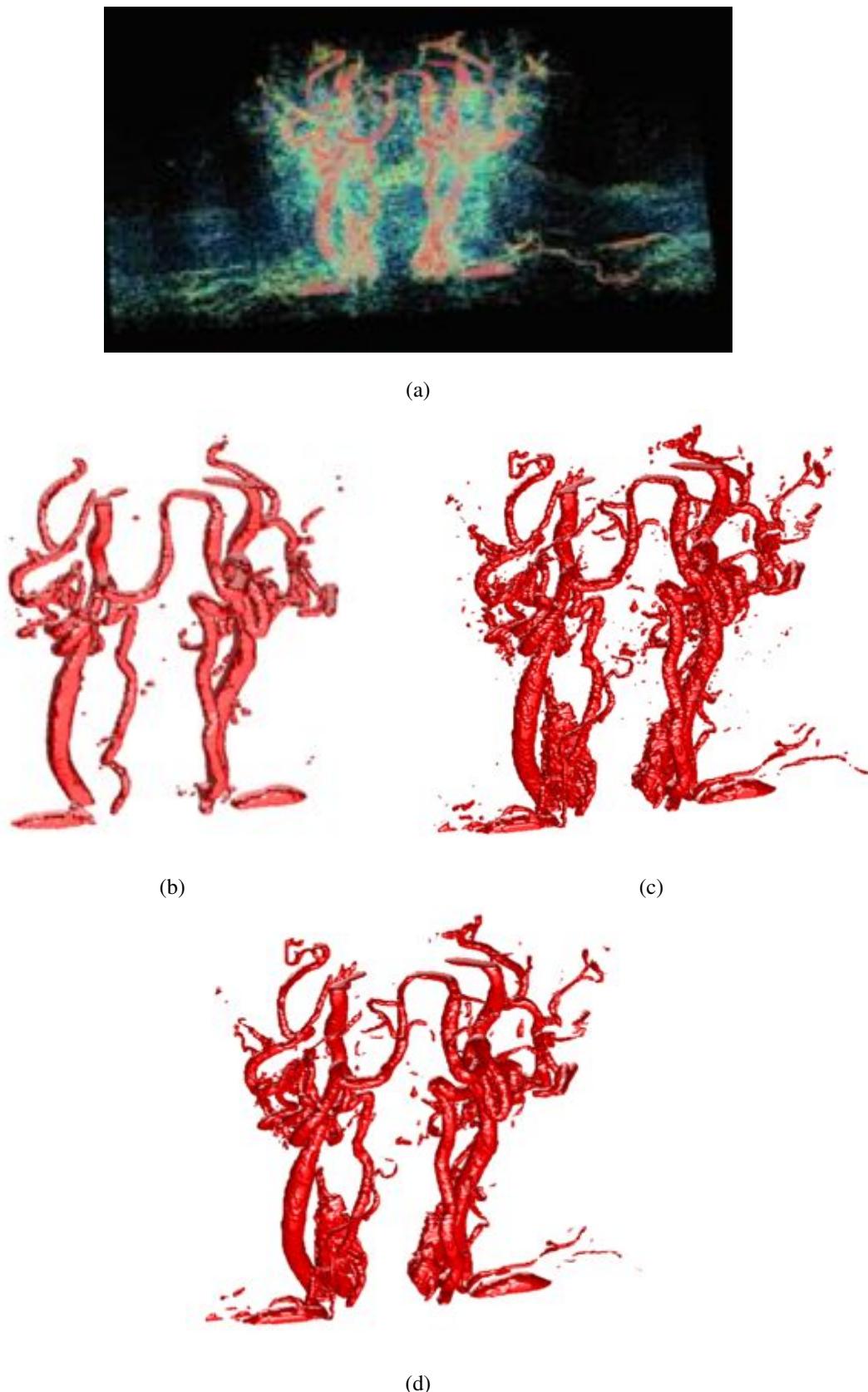


Fig. 5. Segmentation of the brain volume data set. (a) Given MRA image. (b) Result by the method in [23]. (c) Result by our method. (d) Result of our method after smoothing by (4) once.

0 or 1 during the iterations, by taking advantage of this, one can construct a sparse data structure to accelerate the method. Moreover, one can use different tight-frame systems such as those from contourlets, curvelets or steerable-wavelet [19], [11], [25] to capture more directions along the boundary. Though we have proved that our algorithm will always converge to a binary image, it will be interesting to see what functional the binary image is minimizing. The framework for proving convergence for tight-frame algorithms, as developed in [7], may be useful here. These are the directions we will explore in the future.

Acknowledgments.: This work was supported by HKRGC Grant CUHK 400510 and CUHK Direct Allocation Grant 2060408.

REFERENCES

- [1] S. Arivazhagan and L. Ganesan. Texture segmentation using wavelet transform. *Pattern Recognition Letters*, 24, 3197–3203, 2003.
- [2] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. Math. Imaging Vision*, 28(2), 151–167, 2007.
- [3] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion, and shape *Int. J. Comput. Vis.*, 72(2), 195–215, 2007.
- [4] J.F. Cai, R.H. Chan, L.X. Shen, and Z.W. Shen. Simultaneously inpainting in image and transformed domains. *Numer. Math.*, 112(4), 509–533, 2009.
- [5] J.F. Cai, R.H. Chan, L.X. Shen, and Z.W. Shen. Restoration of chopped and nodded images by framelets. *SIAM J. Sci. Comput.*, 30, 1205–1227, 2008.
- [6] J.F. Cai, R.H. Chan, L.X. Shen, and Z.W. Shen. Convergence analysis of tight framelet approach for missing data recovery. *Adv. Comput. Math.*, 31, 87–113, 2009.
- [7] J.F. Cai, R.H. Chan, and Z.W. Shen. A framelet-based image inpainting algorithm. *Appl. Comput. Harmon. Anal.*, 24, 131–149, 2008.
- [8] J.F. Cai, S. Osher, and Z.W. Shen. Split Bregman methods and frame based image restoration. *Multiscale Modeling and Simulation*, 8(2), 337–369, 2009.
- [9] X.H. Cai, R.H. Chan, S. Morigi, and F. Sgallari. Framelet-Based Algorithm for Segmentation of Tubular Structures. SSVM 2011, LNCS6667. Springer, 2011.
- [10] E. Candes and D. Donoho. New tight frames of curvelets and optimal representations of objects with C2 singularities. *Comm. Pure Appl. Math.*, 56, 219–266, 2004.
- [11] E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling and Simulation*, 5(3), 861–899, 2006.
- [12] R.H. Chan, S. Setzer, and G. Steidl. Inpainting by flexible Haar-wavelet shrinkage. *SIAM J. Imaging Sci.*, 1(3), 273–293, 2008.
- [13] R.H. Chan, T.F. Chan, L.X. Shen, and Z.W. Shen. Wavelet algorithms for high-resolution image reconstruction. *SIAM J. Sci. Comput.*, 24(4), 1408–1432, 2003.
- [14] S.H. Cai and K.Y. Li. Matlab Implementation of Wavelet Transforms. <http://taco.poly.edu/WaveletSoftware/>
- [15] T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *Technical Report*, 54, UCLA, 2004.
- [16] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2), 266–77, 2001.
- [17] I. Daubechies. Ten lectures on wavelets. vol. CBMS-NSF Lecture Notes, SIAM, nr., 61, 1992.
- [18] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames. *Appl. Comput. Harmon. Anal.*, 14(1), 1–46, 2003.
- [19] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.*, 14(12), 2091–2106, 2004.
- [20] B. Dong, A. Chien, and Z.W. Shen. Frame based segmentation for medical images. *Technical Report*, 22, UCLA, 2010.
- [21] B. Dong and Z.W. Shen. MRA based wavelet frames and applications. *IAS Lecture Notes Series, Summer Program on The Mathematics of Image Processing, Park City Mathematics Institute*, 2010.
- [22] D.L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3), 613–627, 1995.
- [23] E. Franchini, S. Morigi, and F. Sgallari. Segmentation of 3D tubular structures by a PDE-based anisotropic diffusion model. M. Dæhlen et al. (eds.): MMCS 2008, LNCS5862, pp. 224–241, 2010, Springer-Verlag Berlin Heidelberg, 2010.
- [24] E. Franchini, S. Morigi, and F. Sgallari. Composed segmentation of tubular structures by an anisotropic PDE model. X.-C. Tai et al. (eds.): SSVM 2009, LNCS5567, pp. 75–86, 2009, Springer-Verlag Berlin Heidelberg, 2009.
- [25] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9), 891–906, 1991.

- [26] A. Gooya, H. Liao, et al. A variational method for geometric regularization of vascular segmentation in medical images. *IEEE Trans. Image Process.*, 17(8), 1295–1312, 2008.
- [27] R.C. Gonzales and R.E. Woods. *Digital Image Processing*, 3rd Ed., PrenticeHall, 2008.
- [28] X.Q. Gao, T.Q. Nguyen, and G. Strang. Theory and lattice structure of complex paraunitary filterbanks with filters of (hermitian-) symmetry/antisymmetry properties. *IEEE Trans. Signal Processing*, 49(5), 1028–1043, 2001.
- [29] H. Hassan and A.A. Farag. Cerebrovascular segmentation for MRA data using levels set. *Proc. CARS*, pp. 246–252, 2003.
- [30] C. Kirbas and F. Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, 36, 81–121, 2004.
- [31] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trouset. Model-based detection of tubular structures in 3D images. *CVIU*, 80,130–171, 2000.
- [32] N. Kingsbury. Image processing with complex wavelets. *Philos. Trans. R. Soc. London A, Math. Phys. Sci.*, 357(1760), 2543–2560, 1999.
- [33] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Appl. Comput. Harmon. Anal.*, 10(3), 234253, 2001.
- [34] L.M. Lorigo, O.D. Faugeras, E.L. Grimson, et al. Curves: Curve evolution for vessel segmentation. *Medical Image Analysis*, 5, 195–206, 2001.
- [35] T. Mcinerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2), 91–108, 1996.
- [36] A. Ron and Z.W. Shen. Affine systems in L2(Rd): the analysis of the analysis operator. *J. Funct. Anal.*, 148, 408–447, 1997.
- [37] I. Selesnick, R.G. Baraniuk, and N.G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Process. Mag.*, 22(6), 123–151, 2005.
- [38] B. Sandberg and T.F. Chan. A logic framework for active contours on multi-channel images. *J. Vis. Commun. Image R.*, 16, 333–358, 2005.
- [39] F. Shi and I.W. Selesnick. Video denoising using oriented complex wavelet transforms. *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)* 2, 949–952, 2004.
- [40] H. Scherl, et al. Semi automatic level set segmentation and stenosis quantification of internal carotid artery in 3D CTA data sets. *Medical Image Analysis*, 11(1), 21–34, 2007.
- [41] J.-L. Starck, E.J. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Processing*, 11(6), 670–684, 2000.
- [42] K. Sum and P. Cheung. Vessel extraction under non-uniform illumination: A level set approach. *IEEE Trans. Biomed. Eng.*, 55(1), 358–360, 2008.
- [43] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Process.*, 4(11), 1549–1560, 1995.
- [44] Z. Ye and C.-C. Lu. A complex wavelet domain Markov model for image denoising. *Proc. IEEE Int. Conf. Image Processing, Barcelona*. 3, 365–368, 2003.