Modelling and Extraction of Pulsatile Radial Distension and Compression Motion for Automatic Vessel Segmentation from Video

Alborz Amir-Khalili\textsuperscript{a,}\textsuperscript{*}, Ghassan Hamarneh\textsuperscript{b}, Rafeef Abugharbieh\textsuperscript{a}

\textsuperscript{a}Biomedical Signal and Image Computing Lab, University of British Columbia, Vancouver, BC, Canada
\textsuperscript{b}Medical Image Analysis Lab, Simon Fraser University, Burnaby, BC, Canada

Abstract

Identification of vascular structures from medical images is integral to many clinical procedures. Most vessel segmentation techniques ignore the characteristic pulsatile motion of vessels in their formulation. In a recent effort to automatically segment vessels that are hidden under fat, we motivated the use of the magnitude of local pulsatile motion extracted from surgical endoscopic video. In this article we propose a new approach that leverages the local orientation, in addition to magnitude of motion, and demonstrate that the extended computation and utilization of motion vectors can improve the segmentation of vascular structures. We implement our approach using four alternatives to magnitude-only motion estimation by using traditional optical flow and by exploiting the monogenic signal for fast flow estimation. Our evaluations are conducted on both synthetic phantoms as well as two real ultrasound datasets showing improved segmentation results with negligible change in computational performance compared to the previous magnitude only approach.

Keywords: Motion Analysis, Vessel Segmentation, Ultrasound, Video

1. Introduction

Identification of blood vessels from medical images is important to many clinical procedures. Common applications of vascular imaging range from routine non-invasive diagnostic procedures to complex surgical interventions. Vascular imaging is routinely used to assess the risk for cardiovascular morbidity by (i) directly imaging and analyzing the coronary arteries with intravascular ultrasound (US), magnetic resonance (MR), or computed tomography (CT) imaging; (ii) quantifying arteriosclerosis from color images of the retina (Pedersen et al., 2000); (iii) segmenting atherosclerotic plaque from US (Bots et al., 1997), MR (Duivenvoorden et al., 2009), or CT images (Manningsing et al., 2010) of the common carotid artery (CCA); or (iv) monitoring changes in vascular distensibility from MR images of the aorta (Cavalcante et al., 2011) and CT angiography images of the CCA (Hameeteman et al., 2013) – all of which have been identified as independent predictors of stroke (Hansson, 2005). Moreover, vascular imaging is used regularly during preoperative planning and screening of surgical interventions like kidney and liver transplants (Haijpern et al., 2000; Kamel et al., 2001). Finally, in addition to the traditional applications of X-ray fluoroscopy and CT angiography during image-guided cardiac catheterization (Grossman, 1986) and aneurysm surgery (Kaiabe et al., 2006), vascular imaging is finding new applications in intraoperative guidance during robot-assisted prostate and kidney cancer surgeries (McLeod et al., 2015a; Amir-Khalili et al., 2014, 2015b; Tobis et al., 2011).

Extraction of vascular structures are of such importance that many acquisition techniques and imaging modalities have been specifically developed to enhance the appearance of vasculature in medical images. Such techniques include contrast enhanced CT or MR angiography, laser speckle imaging (Murari et al., 2007), near infrared fluorescence imaging (Tobis et al., 2011), color Doppler US and optical coherence tomography (OCT) (Izatt et al., 1997). Although these modalities and techniques enhance the appearance of the imaged vasculature, many of the aforementioned clinical applications stand to benefit from a fully automatic vessel localization algorithm. The need for automated vessel localization, or segmentation, has in...
Table 1: Categorization and comparison of existing automatic vessel segmentation methods.

<table>
<thead>
<tr>
<th>Modality</th>
<th>Application</th>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D DSA &amp; 3D MRA</td>
<td>Vessel enhancement</td>
<td>Vesselness</td>
<td>Frangi et al. (1998)</td>
</tr>
<tr>
<td>3D MRA &amp; 3D CT</td>
<td>Vessel segmentation</td>
<td>Active contours</td>
<td>Lorigo et al. (2003)</td>
</tr>
<tr>
<td>RGB image</td>
<td>Segment retinal vessels</td>
<td>Vesselness</td>
<td>Staal et al. (2004)</td>
</tr>
<tr>
<td>3D image</td>
<td>Segment retinal vessels</td>
<td>AM</td>
<td>Vermeer et al. (2004)</td>
</tr>
<tr>
<td>3D MRA &amp; 3D CT</td>
<td>Vessel segmentation</td>
<td>Vesselness</td>
<td>Mcintosh and Hamarneh (2006)</td>
</tr>
<tr>
<td>3D MRA</td>
<td>Detect carotid/arteries</td>
<td>OF + MO + HT</td>
<td>Law and Chung (2008)</td>
</tr>
<tr>
<td>DUS</td>
<td>Segment carotid artery</td>
<td>Supervised AM</td>
<td>McLeod et al. (2015)</td>
</tr>
<tr>
<td>2D RGB</td>
<td>Segment retinal vessels</td>
<td>Random forests</td>
<td>Rigamonti and Lepetit (2012)</td>
</tr>
<tr>
<td>2D RGB &amp; 3D RM</td>
<td>Vessel segmentation</td>
<td>CNN</td>
<td>Becker et al. (2013)</td>
</tr>
<tr>
<td>2D US</td>
<td>Segment carotid artery</td>
<td>OF + MO + HT</td>
<td>Amir-Khalili et al. (2014, 2015b)</td>
</tr>
<tr>
<td>2D US</td>
<td>Vessel segmentation</td>
<td>PBRMS</td>
<td>Hennersperger et al. (2015)</td>
</tr>
<tr>
<td>DUS</td>
<td>Detect dural pulsation</td>
<td>EKF + PS</td>
<td>Gastounioti et al. (2015)</td>
</tr>
<tr>
<td>DUS</td>
<td>Segment carotid artery</td>
<td>PBMS</td>
<td>Amir-Khalili et al. (2015a)</td>
</tr>
<tr>
<td>3D MR</td>
<td>Segment carotid artery</td>
<td>PBRMS</td>
<td>Gao et al. (2017)</td>
</tr>
</tbody>
</table>

AM: Appearance Model; BM: Brightfield Microscopy; CNN: Convolutional Neural Networks; CT: Computed tomography; DSA: Digital Subtraction Angiography; DUS: Dynamic Ultrasound; EK: Extended Kalman Filter; EV: Endoscopic Video; FS: Frequency Smoothing; GBS: Genetic B-spline Segmentation; GWR: Group-wise Registration; HFS: Hessian Features; MO: Morphological Operations; MRA: Magnetic resonance angiography; NURBS: Non-uniform rational B-spline; PRMS: Phase-Based Motion Segmentation; PS: Phase-based Segmentation; PRMM: Pulsatile Radial Motion Model; US: Ultrasound; USCM: Ultrasound Confidence Maps; WF: Wavelet Features; Y: Yes; N: No

Table 1: Categorization and comparison of existing automatic vessel segmentation methods.

The reader is referred to comprehensive surveys of vessel segmentation techniques (Suri et al., 2002; Kirbas and Quek, 2003; Lesage et al., 2009; Kirisli et al., 2013) for more information on other existing methods.

The early attempts at automatic vessel segmentation focus on applying advanced low-level pixel based image analysis techniques to static intensity information acquired from the aforementioned imaging modalities. Such attempts include the exploitation of ridge-like features in the image (Staal et al., 2004), Hessian-based vesselness features (Frangi et al., 1998; Hennersperger et al., 2015), Law and Chung (2008), and model/physics based approaches (Vermeer et al., 2004; Hennersperger et al., 2015). Other high-level techniques have also been proposed by embedding these low-level features in broader frameworks, which include: vessel trackers (Mcintosh and Hamarneh, 2006), deformable 3D cylindrical non-uniform rational B-spline surface models (Gao et al., 2017), a combination of wavelet-based features and machine learning (Soares et al., 2006), active contours (Lorigo et al., 2001), and supervised machine learning techniques (Schaap et al., 2011; Rigamonti and Lepetit, 2012; Becker et al., 2013). With the exception of Doppler US and OCT, the techniques listed above and in cited survey papers (Suri et al., 2002; Kirbas and Quek, 2003; Lesage et al., 2009; Kirisli et al., 2013) focus on extracting low- and high-level features from static information alone, ignoring the most characteristic feature of a pulsating vessel, i.e., its kinematics or temporal behavior.

On the other hand, US and OCT exploit the pulsatile flow kinematics of blood inside the vessels to facilitate localization. Such modalities are capable of measuring the directionality and relative velocity of structures (usually blood) by leveraging the Doppler effect. The flow of blood, however, is not the only temporal characteristic of vascular structures. The pulsatile radial distension and compression of the vascular walls (from the lumen to tunica externa) is another characteristic that can be observed and measured using almost any imaging modality so long as the temporal and spatial resolutions are adequate.

The first use of temporal features for the purpose of vessel segmentation did not explicitly model the kinematics (Riha and Beneš, 2010). In their paper, the authors simply assumed that the only meaningful movement in dynamic US (DUS) scan of the CCA imaged along the transverse axis is the pulsatile movement of a circular pattern. Based on this assumption the authors propose to use an optical flow (OF) algorithm to extract motion vectors from adjacent frames in a sequence and simply average the absolute value (magnitude) of motion across the entire sequence to generate features. These features are then processed with median filtering and morphological operations (MO) to generate a binary mask. High-level features are finally extracted from the Hough transform (HT) of the binary mask and the resulting features, along with the last frame of the sequence, are fed into a Bayesian classifier to compute the center and radius of the CCA.

We initially proposed to exploit the kinematics of pulsating vessels in the context of kidney cancer surgery to identify major vessels that are hidden under layers of connective tissues (Amir-Khalili et al., 2014, 2015b). Rather than a simple computation...
of the average magnitude of motion using OF, we proposed the use of a temporal bandpass filter to isolate features that are in sync with the heart-rate. In our approach, we reformulated the Eulerian video magnification (Wadhwaa et al., 2013) pipeline into a multi-scale phased-based motion segmentation (PBMS) algorithm to detect the motion of renal vessels by analyzing the magnitude of temporal change in the local phase information of an endoscopic video (EV) sequence. Our PBMS method, although novel in application, only operates on the magnitude of local pulsatile motion and is consequently prone to false positives when tested on other applications and imaging modalities, failing to differentiate between the motions specific to vasculature versus neighboring structures that happen to move at the same frequency as blood vessels. In a recent publication (Amirkhalili et al., 2015a), to reduce the number of false positives and to extend the application of our method to more challenging imaging modalities, such as DUS sequences of the CCA, we proposed a novel kinematic model-based vessel segmentation (KMVS) pipeline that couples a pulsatile radial motion model (PRMM) with a more detailed computation of motion characteristics that entails an estimation of the local magnitude and orientation of motion. We showed that this updated pipeline increases the accuracy of kinematics-based vessel segmentation and that by reconstructing the monogenic signal (Felsberg and Sommer, 2001) and computing the motion vectors using a monogenic flow (MF) technique, the local orientation of motion may be estimated in a more computationally efficient manner compared to the previous PBMS method.

Concurrent with our efforts, other novel methods have been proposed to address similar challenges with the help of pulsatile kinematics models. In a recent publication, it was demonstrated that a kinematic model of periodic low velocity out-of-plane motion of structures in DUS using extended Kalman filters (EKF) and frequency smoothing (FS) can localize dural pulsation for spine needle interventions (McLeod et al., 2015b). The proposed method operates in real-time and is capable of detecting subtle motions that are imperceptible in Doppler US. Furthermore, the proposed visualizations were shown to reduce the normalizing path length and number of attempts required to perform a mock epidural procedure on a spinal phantom model. Although this method was shown to be effective in the novel application presented, similar to our PBMS method, it will likely not be able to distinguish between vascular structures and others that happen to translate at the same frequency as vessels. The FS aspect of the proposed method may allow the EKF approach to perform better than PBMS, but this method cannot benefit from an advanced kinematic model of vasculature due to the lack of a mechanism to account for the spatial orientation of motion.

In the context of CCA atherosclerosis assessment, another method was proposed to learn the kinematic dependencies between atherosclerotic and healthy vascular tissue in DUS by combining group-wise image registration (GWIR) with a graph-based segmentation (GBS) scheme (Gastounioti et al., 2015). Rather than implementing a physics-based kinematic model, the authors proposed a data-driven approach to learn a complex discriminative model. To do this, the magnitude of total vertical and horizontal displacements (MTD) are first computed for every pixel throughout the sequence using GWIR. Then, independent component analysis is used to identify the dominant and independent motion classes, which are used as a basis to which the MTD of each pixel is mapped using mutual information. A final mutual information value is assigned to a given pixel through majority voting. The likelihood of a pixel belonging to a binary class (healthy or atherosclerotic) given the final map is first learned and then used as the data term to perform GBS and generate contiguous contours around the atherosclerotic regions. Segmenting atherosclerotic plaque from DUS is challenging and the proposed pipeline performs well. It can be argued that the pipeline may be modified to segment vascular structures in addition to the plaque regions. Even though real-time performance is not a strict requirement for diagnostic clinical applications, the speed of the algorithm is of clinical value. The authors do not mention the runtime of their pipeline and the GWIR method used in the paper was projected to take minutes to complete, at best, if optimized and implemented in C++ (Sotiras et al., 2009). It is thus unlikely that the proposed method would be able to perform in real-time.

1.2. Contributions

Our objective in this paper is to assess the performance of vessel localization using kinematic pulsatile radial motion models without the aid of learning or appearance models to motivate the use of motion-based features within a broad range of vessel segmentation applications including image-guided therapies and routine diagnostic procedures. Specifically, we aim to disseminate a fast and effective low-level motion based segmentation pipeline that ultimately may be incorporated, alongside complementary low-level intensity based features, into existing high-level discriminative segmentation frameworks (Becker et al., 2013; Hennersperger et al., 2015). To that effect, we elucidate the intuition behind our KMVS pipeline (Amirkhalili et al., 2015a) in detail and extend our original formulation with alternative optimized and off-the-shelf techniques for performing the motion estimation stage of the pipeline. We start by motivating the Eulerian motion estimation method employed in our pipeline in order to link our current approach with our previous PBMS method. We then delve into the different ways that complete motion vectors may be estimated from a sequence of images, i.e., OF and MF, and the trade-offs associated with each method. We evaluate the performance of all proposed methods with our original synthetic and clinical DUS scans of the CCA as well as an additional large external dataset of publicly available DUS sequences of the CCA imaged along the transverse plane. Through the analysis of our results on the DUS datasets, we conclude that a tuned OF motion estimation method increases the segmentation accuracy and computational performance of the KMVS pipeline beyond that of our original formulation (Amirkhalili et al., 2015a). Finally, we highlight the future prospects of our kinematic approach; namely, the extension of the KMVS pipeline to high dimensional, 3D+time, data and into the native radio frequency representation of DUS.
2. Methodology

As motivated in the Introduction section, the goal is to leverage an understanding of the kinematics of vascular structures to perform image segmentation. These kinematics are observed through the change in intensity information of anatomical structures captured in a video or dynamic sequence of frames. In this paper, each $j^{th}$ frame of such sequence is defined as a scalar-valued (grayscale) function $f : (x \in \mathbb{R}^2, j \in \mathbb{R}) \rightarrow \mathbb{R}$ mapping a pixel $x = (x_1, x_2)^T$ in the 2D spatial domain of each frame to an intensity value. Depending on the spatiotemporal resolution of the sequence and the specific vascular anatomy being imaged, some kinematics of the vasculature may be observed by the naked eye, the most notable characteristic being the periodic motion of the vascular walls induced by the pulsatile flow of blood in the vessel. The visibility of this phenomenon—or the magnitude of observable displacement—varies depending on the radius, thickness, and viscoelastic properties of the vascular walls as well as the flow rate and pressure of blood inside the vessel (Warriner et al., 2008). In previous publications (Amir-Khalili et al., 2014; 2015b), we demonstrated that advanced Eulerian motion estimation techniques may be used to observe these phenomena, even in situations where the motions are subtle and imperceptible to the naked eye, by observing the temporal change in $f$ at every pixel $x$.

There are different ways to identify periodic motions occurring at a given pixel. For blood vessels, these periodic pulsations are within a temporal passband centered on the heart-rate of the patient. A trivial way to identify this periodic motion is to apply a temporal bandpass filter to the raw intensity information $f$ at every pixel $x$ independently. This naïve approach is prone to error as it does not consider the motion of neighboring pixels, is sensitive to noise, and it cannot estimate the magnitude of motion, which is required to attenuate the effects of noise. In our previous phase-based motion segmentation (PBMS) approach (Amir-Khalili et al., 2014; 2015b), we overcame these limitations by (i) estimating the magnitude of motion via the change in local (spatial) phase information and (ii) measuring this change over multiple spatial scales and orientations to encode the motion information from neighboring pixels.

2.1. Phase-Based Motion Segmentation

The shift property of Fourier transform states that motion is directly related to the change in phase. Our aforementioned PBMS method is built on this foundation and relies on a complex “analytic” steerable pyramid to extract local motion information from a sequence of grayscale images. The estimation of motion from local phase information is straightforward when there is only one spatial dimension. In 1D, local phase can be measured by constructing the analytic signal. The analytic signal is constructed from quadrature filters, i.e., 1D Hilbert pair of bandpass filters. The estimation of local phase is more complex in 2D images and there are thus many approaches to extend the analytic signal to 2D. One approach is to use a steerable complex pyramid decomposition (Portilla and Simoncelli, 2000) to extract local motion information from a sequence of grayscale images.

2.1.1. Multi-Scale Steerable Analytic Decomposition

With the complex steerable pyramid, the analytic signal is estimated at different scales $s = 1..S$, along $n = 1..N$ different orientations from the complex response $\mathbf{h}(\mathbf{x}, j; s, n) : \mathbb{R}^3 \rightarrow \mathbb{C}$ to a set of steerable filters $\mathbf{b}(\mathbf{x}, s, n)$. The real and imaginary parts of $\mathbf{h}(\mathbf{x}, j; s, n)$ correspond to a pair of even- and odd-symmetric filter responses that are analogous to a one-dimensional Hilbert transform along the given orientation. The orientations are sampled evenly such that the local orientation $\theta_n$ (where $\Delta x = (\cos{\theta_n}, \sin{\theta_n})^T$ is determined by $\theta_n = \pi n/N$, where $N$ is the total number of orientations used in the pyramid. The steerable PBMS method measures the magnitude of local phase $\phi(\mathbf{x}, j; s, n) = \text{arg}(\mathbf{h}(\mathbf{x}, j; s, n))$ projected onto $N$ different angles $\theta_n, n = 1..N$.

2.1.2. Spatiotemporal Filtering

To identify locations where $\phi$ exhibits periodic motion, we filter the local phase measurements using an ideal bandpass filter:

$$ z(j) = 2\tau_L \text{sinc}(2\tau_L j) - 2\tau_T \text{sinc}(2\tau_T j), $$

where $\tau_L$ is the temporal low frequency cut-off and $\tau_T$ is the high frequency cutoff and the sinc functions are the time domain representations of rectangular functions in the temporal frequency domain that construct an ideal bandpass. The response of the temporal bandpass filter is $\hat{\phi}(\mathbf{x}, j; s, n) = \phi(\mathbf{x}, j; s, n) \ast z(j)$.

Noise is suppressed by multiplying the bandpassed phases and the normalized magnitude of the spatial filter response vectors to obtain $\hat{\phi}_s(\mathbf{x}, j; s, n) = \mathbf{h}(\mathbf{x}, j; s, n)\hat{\phi}_s(\mathbf{x}, j; s, n)$. A spatiotemporal median filter is also used to remove the impulse noise introduced in the temporal filtering step.

2.1.3. Multi-Scale Motion-Based Segmentation

This denoised product $\hat{\phi}_s(\mathbf{x}, j; s, n)$ is averaged across all scales $s$, orientations $n$, and frames $j$ to obtain our final fuzzy labels

$$ I_P(\mathbf{x}) = \frac{1}{M_P} \sum_{j, s, n} \left| \hat{\phi}_s(\mathbf{x}, j; s, n) \right| \frac{1}{2\pi \omega_s}, $$

where $\omega_s$ is the spatial frequency of scale $s$ and $M_P$ is a normalizing factor to fix the range of PBMS labels $I_P$ to $[0, 1]$. The simple averaging across all orientations only considers the weighted mean magnitude of motion. An alternative approach, presented initially in our kinematic model-based vessel segmentation (KMVS) paper and described in detail in the following Section 2.2 is to compute the motion vectors entirely (magnitude and orientation) and to implement a better kinematic model that only detects vessel-like structures, which radially distend and contract in time.

2.2. Kinematic Model-Based Vessel Segmentation

The PBMS method, described in the previous section 2.1 leverages the pulsatile temporal motion of structures to approximate the location of blood vessels. In addition to pulsatile motion, the geometry of the vessel is also an integral part of its distinguishing kinematics. Blood vessels are tubular structures and...
when these structures are subjected to a pulsatile flow, the vascular walls undergo radial and longitudinal displacements (Warnier et al., 2008). The radial displacement component manifests as the expansion and contraction of vessel walls in medical images. These motions are unique to vasculature, unlike the longitudinal motions that also occur in the surrounding tissues. The complete computation of local motion vectors, that encode longitudinal motions that also occur in the surrounding tissues.

The major components and differences between PBMS and KMVS are presented in Fig. 1. The same filter parameters are used to create the temporal bandpass filters in both PBMS and KMVS pipelines but the dimensionality of inputs and outputs to the filter are different.

Figure 1: Overview of the PBMS and KMVS segmentation pipelines presented in this article. In KMVS, the simple multi-scale motion-based segmentation (red) step is replaced with a complex pulsatile radial motion model (cyan). This radial model is built atop a multi-scale motion computation step (blue) instead of the simpler steerable phase pyramid decomposition (orange) approach used in PBMS. Although the parameters of the temporal bandpass filter (green) used in both approaches are the same, the dimensionality of inputs and outputs to the filter are different.

which can also be expressed as a continuous problem

\[
\frac{df(x, j)}{dj} = \frac{\partial f}{\partial x} \frac{dx}{dj} + \frac{\partial f}{\partial j} = \nabla_x f \mathbf{v} + \nabla_j f = 0, \tag{5}
\]

where \(\frac{\partial f}{\partial x} = \nabla_x f\) is the spatial gradient of the frame, \(\frac{\partial f}{\partial j} = \nabla_j f\) is the difference between the intensities of two frames, and \(v\) is the velocity, or flow, of motion.

A simple solution to this problem, initially proposed in the context of computational stereopsis, is to assume that the motion is constant over a local window. This assumption leads to an overcomplete system of equations that can be solved using least squares iteratively by warping the image at each iteration (Lucas and Kanade, 1981). Additionally, large displacements may be accounted for by performing the iterative optimization over multiple spatial scales as well, by subsampling the frames in the spatial domain. Computational stereopsis problems comprise a subset of OF problems, where correspondence between two rectified stereo images are constrained to only one dimension (Hartley and Zisserman, 2004). Once the motion of objects within the scene are generalized to 2D, the simple localized least squares solution becomes ill-conditioned and the problem of motion estimation becomes more difficult due to the aperture problem. An alternative way to overcome this problem, proposed by Horn and Schunck (1981), is to impose a global smoothness constraint over the flow vectors \(\mathbf{v}\) and solve the problem globally.

Both of these local and global approaches of computing flow have drastically advanced since their inception and have also been successfully combined into a unified framework (Bruhn et al., 2005), which combines the advantages of both approaches, i.e., robustness to noise and ability to yield dense flow fields. In our analyses, we opted to use two modern implementations of OF (Liu, 2009; Sun et al., 2010) implemented in MATLAB for the purpose of computing the dense motion vector \(\mathbf{d}(x, j)\).

One of the major drawbacks of OF lies in the brightness consistency assumption. This assumption is sensitive to smooth contrast variations (temporal changes in lighting conditions) and other similar situations where pixel intensities cannot be considered as reliable features. These scenarios are abundant in medical image sequences; examples include: moving light sources in endoscopic video, specular noise or non-Lambertian reflections, and local brightness variations caused by complex acoustic beam propagation in dynamic ultrasound (DUS). In context of 3D DUS imaging, Alessandri et al. (2012) attribute brightness consistency violations (temporal variations in the local echo strength) in part to changes in the angle between connective tissue fibers and beam propagation direction, and the limited acquisition frame rates of 3D DUS. 2D DUS benefits from higher frame rates, but suffers from artefacts caused by the out-of-plane motion of structures within the 2D field of view. Structures with varying thickness and acoustic properties may travel through the field of view during acquisition and cast a time varying acoustic shadow on surrounding tissues. This results in local attenuation or amplification of 2D B-mode intensity values which are not necessarily correlated to the rela-
vant in-plane motions. On the other hand, it has been argued that phase-based computation of motion vectors is more robust to this type of noise and has the added advantage of producing subpixel accuracy without explicit subpixel reconstruction or feature localization (Fleet and Jepson, 1990; Wadhwa et al., 2013). This was the original motivation behind our choice to utilize a phase-based (Wadhwa et al., 2013) approach in our PBMS pipeline compared to seminal gradient-based Eulerian video magnification approaches (Wu et al., 2012).

2.2.2. Monogenic Flow

The monogenic (MON) signal is another 2D extension of the analytic signal (similar to the steerable phase-based method presented in section 2.1.1) and it provides an efficient framework for extracting the local orientation $\theta$ and the local phase $\phi$ features from an image. By measuring the temporal change in $\phi$, we can estimate motion (Felsberg, 2007; Alessandri et al., 2013). The MON signal is constructed from a trio of bandpass filters with finite spatial support. This trio is commonly referred to as spherical quadrature filters (SQF) (Felsberg and Sommer, 2001). To estimate the motion of both small and large structures in each frame, we generate different SQF by tuning the spatial passband of the filters to varying scales.

Each set of SQF comprises an even (symmetric) radial bandpass filter and two odd (antisymmetric) filters. The odd filters are computed from the Riesz transform, a 2D generalization of the Hilbert transform, of the radial bandpass filter (Felsberg and Sommer, 2001). In the literature, many different bandpass filters have been proposed to construct the SQF including: first order Gaussian (Boukerroui et al., 2004), Cauchy (Boukerroui et al., 2004), Poisson (Felsberg and Sommer, 2004), and difference of Poisson filters (Wietzke et al., 2009). We employ Log-Gabor (Kovesi, 1996) bandpass filters as they suit the natural statistics of an image (Soares et al., 2006; Field, 1987) and maintain zero DC gain at lower spatial scales. For every scale $s$, the even Log-Gabor component of the SQF is expressed as

$$B_x(u; s) = \exp \left( -\frac{[\log|\omega_s|]^2}{2[\log k]^2} \right),$$

in the frequency domain $u = (u_1, u_2)^T$, where $k$ and $\omega_s$ are parameters of the filter. The parameter $k = \sigma/\omega_s$ is a fixed constant representing the ratio of the standard deviation $\sigma$ of the Gaussian describing the Log-Gabor filter’s transfer function in the frequency domain to the filter’s center frequency $\omega_s$. At each scale $s$, the center frequency is defined as $\omega_s = (\lambda_0 2^{s-1})^{-1}$, where $\lambda_0$ is an initial minimum wavelength. The radial bandpass filter $B_x$ is symmetric as it is only a function of the magnitude of the frequency $u$. Using the Riesz transform, we compute the two odd components ($B_{x1}$ and $B_{x2}$) associated to this SQF as

$$B_{x1}(u; s) = \frac{i u_1}{|u|} B_x; \quad B_{x2}(u; s) = \frac{i u_2}{|u|} B_x.$$  

In the spatial domain, the components of the MON signal $(h_x, h_y)$ are obtained by convolving the SQF with a given frame of the sequence such that

$$h_x(x, j; s) = \mathcal{F}^{-1} \left[ B_x(u; s) F(u, j) \right]$$

$$h_{x1}(x, j; s) = \mathcal{F}^{-1} \left[ B_{x1}(u; s) F(u, j) \right]$$

$$h_{x2}(x, j; s) = \mathcal{F}^{-1} \left[ B_{x2}(u; s) F(u, j) \right]$$

$$h_y(x, j; s) = (h_{x1}(x, j; s) h_{x2}(x, j; s))^T,$$

where $F(u, j) = \mathcal{F}[f(x, j)]$ is the frequency domain representation of the frame.

From the SQF responses (8), the phase vector $r$ is then defined as the continuous representation of local orientation $\theta$ and local phase information $\phi$ such that

$$r(x, j; s) = \phi(c \cos \theta, s \sin \theta)^T = \frac{h_x}{|h_o|} \arg(h_x + \rho h_y).$$

Local motion may then be calculated by first computing the components of a 3D rotation that relates the response of two adjacent frames in the video

$$\Delta h_x = h_x(x, j; s) h_x(x, j + 1; s) + h_y(x, j; s)^T h_y(x, j + 1; s)$$

$$\Delta h_y = h_y(x, j; s) h_y(x, j + 1; s) - h_x(x, j + 1; s) h_x(x, j; s)$$

and then computing the phase differences $\Delta r$ by substituting (10) into (9). Given a local neighborhood $N$, the local displacement $d_N(x, j; s)$ is calculated from

$$\sum_{x \in N} [\nabla^T r(x, j; s)] d_N(x, j; s) = \sum_{x \in N} \Delta r(x, j; s)$$

where $\nabla^T$ is the divergence operator. The derivation of (9), (10), and (11) from the response to the SQF falls outside of the scope of this paper and can be found in the original MF paper (Felsberg, 2007). To improve the estimate for the displacement vector $d_N(x, j; s)$, we compute the mean of this value across all scales $s$. The computed $d_N(x, j)$ is an estimate of the true motion vectors $d(x, j)$ that relate two frames in a sequence $f(x, j + 1) = f(x - d(x, j), j)$. This is the same motion vector that is computed by traditional OF techniques.

2.2.3. Pulsatile Radial Motion Model

Let $d(x, j)$ define a motion field containing the motion vectors estimated for all adjacent frames inside a given sequence using either OF or MF. We first isolate the motions that are in sync with the heart-rate by applying the same ideal temporal bandpass filter (Fig. 1) described in [1]. We define the temporally bandpassed motion vectors as $d_t(x, j) = d(x, j) * z(j)$.

Temporal filtering alone does not distinguish between structures that distend radially and tissues that translate at pulsatile frequency. Pulsating vessels are subject to periodic expansion and contraction, and the key insight is that the orientation of motion vectors are opposing each other along the center-line of the vessel during both contraction and expansion. Such vector fields thus exhibit high divergence along the center-line of the structure as illustrated in Fig. 2.

In physical terms, divergence measures the extent to which a point source in the vector field behaves as a sink or source and
it is defined as the sum of the partial derivatives of the vector field

$$
\nabla^T \mathbf{d}(x, j) = \frac{\partial \mathbf{d}}{\partial x_1} + \frac{\partial \mathbf{d}}{\partial x_2}. 
$$

(12)

Due of the tubular geometry of vessels, the radial motion along the center-line of the vessels are weaker compared to the regions that are along the walls. We account for this by computing the divergence across multiple spatial scales; the motion field at each scale denoted \( \mathbf{d}(x, j, s) \). At each scale, we downsample the vector field by a factor of two using bilinear interpolation. The resulting vessel labels are computed to be

$$
I_k(x) = \frac{1}{M_k} \sum_{j,s} |\nabla^T \mathbf{d}(x, j, s)|,
$$

(13)

where \( M_k \) is a normalizing factor to fix the range of KMVS labels \( I_k \) to [0, 1].

3. Experiments and Results

In this section we present two experiments to evaluate the performance of the different approaches presented in the methodology section. In each experiment, we compare the original phase-based motion segmentation (PBMS) implementation against four different implementations of kinematic model-based vessel segmentation (KMVS). The four implementations of the KMVS consist of different motion estimation techniques, which includes two optical flow (OF) techniques \( \text{OF1, OF2} \) \cite{Li2009} and the simple monogenic flow (MF) method presented in section 2.2.3 and a more robust alternative to compute MF \( \text{MF1, MF2} \) \cite{Alessandrini2013}.

3.1. Implementation Details

The different implementations of PBMS and KMVS are denoted PB, OF1, OF2, MF1, and MF2. A summary of the different parameters used in each implementation is provided in Table 2. In our experiments, we use the default parameters of PB \( \text{Amir-Khalili et al. (2015b)} \) and the \text{classic+nl-fast} setting of OF1 \( \text{Sun et al. (2010)} \). In the OF2 \( \text{Li2009} \) implementation, we manually tuned the parameters to increase the speed and performance of the algorithm. The minimum wavelength was set to \( \lambda_0 = 8 \), weight of the regularization \( \alpha \) was decreased to 0.022, the scale multiplier (inverse of downsampling ratio) was set to match other implementations \( \delta = 2 \), number of outer \( N_o \) and inner \( N_i \) fixed point iterations were reduced to 1, and the default number of successive over relaxation iterations \( N_{\text{ iterate}} = 20 \) was found to be sufficient for our purposes. The parameters for the MF1 \( \text{Alessandrini et al. (2015a)} \) method were set to values used in the original publication: \( \lambda_0 \) is set to 2, ratio \( k \) for the Log-Gabor filter was set to 0.05, and a 7×7 box filter was used to average the displacements over the neighborhood of \( N \). Note that the 7×7 box filter imposes a constant flow assumption over small overlapping local neighbourhoods and does not adversely impact segmentation results in our experiments. The number of scales are set such that \( s = \lfloor \log_2(L) \rfloor - 1 \) where \( L \) is the smallest image dimension. MF2 \( \text{Alessandrini et al. (2013)} \) was setup to use the same filter parameters as MF1 but the frequency and scale computation mode was set to robust. The motion estimation mode was changed to \text{lucas kanada} as the spatially affine transformation model was not performing well on our dataset. The remaining parameters were kept at their default values.

All of the methods described were implemented in MATLAB \( \text{MATLAB} \) \footnote{MATLAB executables are publicly available for download from \url{https://bisicl.ence.ubc.ca/software/radialDistension.html}} running on a workstation with an Intel 3.7 GHz Xeon E5-1620 processor and 8 GB of RAM. The source code for OF1, OF2, and MF2 is publicly available online.

<table>
<thead>
<tr>
<th>Name</th>
<th>Motion Estimation</th>
<th>Parameters</th>
<th>Kinematic Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>Amir-Khalili et al. (2014, 2015b)</td>
<td>( \text{classic+nl-fast} )</td>
<td>Multi-Scale Motion-Based Segmentation</td>
</tr>
<tr>
<td>OF1</td>
<td>Sun et al. (2010)</td>
<td>( \lambda_0 = 8 ), ( \alpha = 0.022 ), ( \delta = 2 ), ( \lambda_0 = 1 ), ( N_{\text{ iterate}} = 20 )</td>
<td>PBMM ( s = \lfloor \log_2(L) \rfloor - 1 )</td>
</tr>
<tr>
<td>OF2</td>
<td>Liu (2009)</td>
<td>( \lambda_0 = 2 ), ( \lambda = 0.05 ), ( \alpha = \lfloor \log_2(L) \rfloor - 1 ), ( N_{\text{ iterate}} = 7 \times 7 ) box</td>
<td></td>
</tr>
<tr>
<td>MF1</td>
<td>Amir-Khalili et al. (2015a)</td>
<td>( \lambda_0 = 2 ), ( \lambda = 0.05 ), ( N_{\text{ iterate}} = 20 )</td>
<td></td>
</tr>
<tr>
<td>MF2</td>
<td>Alessandrini et al. (2013)</td>
<td>( \text{filter type = logGabor, iter mode = robust, iter mode = robust, } \lambda_0 = 2 ), ( \lambda = 0.05 ), ( N_{\text{ iterate}} = 20 )</td>
<td></td>
</tr>
</tbody>
</table>

3.2. Materials and Experimental Setup

Although our methods are applicable to other imaging modalities, ultrasound (US) is ideal for validation as it can im-

![Figure 2: Depiction of simple 2D motion vector fields (blue arrows) and corresponding scalar divergence value plotted on an orthogonal axis (red arrows). The occurrence of divergent vascular wall motions are illustrated on longitudinal (b) and transverse (c) cross sections of an artery. Vector fields that purely translate in one direction (a) are incompressible or divergence free, whereas radial distension motion (b and c) exhibit high divergence along the center of expansion due to the tubular shape of the vessel. (For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image-url)
age vessels in the transverse and longitudinal axes, it has high temporal resolution, and the vessels can be manually delineated with accuracy and regarded as ground truth for evaluation. We validate the performance of PBMS and different implementations of KMVS on a set of synthetic computational phantoms and two dynamic ultrasound (DUS) datasets of common carotid artery (CCA) scans. The phantom dataset is designed to mimic the kinematics of pulsating vascular structures imaged along the transverse and longitudinal slices and is described in further detail in Section 3.3. The first DUS dataset, hereby referred to as the UBC dataset, was acquired in-house and consists of eight sequences from three volunteers with six scans acquired along the transverse and two along the longitudinal axis of the CCA captured at 30 frames per second. The first frame of each sequence was manually segmented for quantitative analysis. These eight cases and the synthetic dataset were previously presented in our KMVS paper (Amir-Khalili et al., 2015a) and are used as training dataset to select the parameters presented in Table 2. A secondary publicly available DUS transverse scans of CCA (Riha et al., 2008; Riha and Potuček, 2009; Riha and Benes, 2010; Riha et al., 2013), referred to as the SPLab dataset, is used to further corroborate our findings. We selected a total of 35 sequences from the dataset with each sequence containing four to eight frames captured at an estimated three frames per second. Every frame in the SPLab dataset is accompanied by image coordinates that define a tight bounding box (Fig. 3a) around the CCA. These coordinates are used to generate an ellipsoidal mask (Fig. 3b) for the first frame of the selected sequences to serve as an approximate ground truth (Fig. 3c), in lieu of manual segmentations.

### 3.3. Phantom Experiment

We use three computational phantoms, in a two-frame matching experiment, to compare the effectiveness of our MF and OF based segmentation techniques to the PB method. Temporal filtering was not used in this experiment. Our phantoms consist of: pulsating and translating circles (top row in Fig. 4 and 5), pulsating and translating tubular structures (middle row in Fig. 4 and 5), and a noise pattern that undergoes a combina-

Figure 3: Processing steps used to generate ground truth data for the SPLab dataset. Each frame in the dataset is accompanied by a set of coordinates that define a bounding box, overlayed in yellow, around the CCA (a). We generate an ellipsoidal mask (b) inside this bounding box and use it as an approximate ground truth overlayed in red (c). (For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Figure 4: Qualitative phantom experiments illustrating the results of PB, OF1, OF2, MF1, and MF2 implementations. Top row: circle phantom. Bottom row: tubular phantom. Columns (b-f): the estimated 2D motion vectors of each implementation applied to the phantom dataset. The final segmentation results are presented in column (c). Overall, the qualitative results favor the MF1 implementation of KMVS in column (c). Refer to Section 3.3 for detailed explanation of annotations ⊙ to ⊙. (For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Figure 5: Intermediate results of the motion estimation stage of different KMVS implementations applied to the phantom dataset. The final segmentation results are presented in Fig. 4. The PBMS method is not included in this comparison as it does not explicitly compute the flow vectors. Column (a): the first frame of each phantom sequence with red contours around structures that distend or contract radially and yellow contours for structures that are subject to translation only. Columns (b-e): the estimated 2D motion vectors of each implementation is color-coded such that hue represents the direction and saturation represents the relative magnitude of motion. The color-bar to the right shows the mapping of hue to the orientation of motion measured in degrees. Refer to Section 4.2 for detailed explanation of annotations ⊙ to ⊙. (For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
tion of pulsating and translating motions in shape of circles and tubes (bottom row in Fig. 4 and 5). The fuzzy automatic segmentation results obtained with all five implementations presented in section 3.1 are presented in Fig. 4, and the intermediary motion computation results of the KMVS implementations are presented in Fig. 5.

In our phantom experiments (Fig. 4), the PB implementation is only capable of detecting the translating structures (C) in Fig. 4c and the edges of the larger pulsating structures (C) in Fig. 4a that appear as local translations. Among the proposed KMVS implementations, MF2 (d) fails to localize the large pulsating structure in the circle phantom (C) in Fig. 4i and any of the structures in the tube phantom. This is due to the fact that the MF2 method fails to estimate the motion of large circular structures in the circle phantom (C) in Fig. 5c and any of the tubular structures in the tube phantom (Fig. 5a). The algorithm cannot detect the motion of the larger circles due to the fact that default window sizes, over which the motion is computed, are too small. Increasing the window size improves the performance when computing the motion of the larger circles, but it increases the computation time and adversely affects the overall performance on the real data presented in the following section, thus the default values were kept. As for the poor performance on the tube phantom, upon inspecting the algorithm, we noted that this may be due to a stability check performed by the algorithm during motion estimation. We found that relaxing the stability threshold results in some motions being correctly measured on the tube phantom but we ultimately decided to leave the threshold unchanged as lower limits resulted in more errors on other phantoms.

The results of motion estimation presented in Fig. 5 are visibly different, especially for the circle and tube phantoms, as both of the OF codes used rely on Horn and Schunck (1981) style regularization (flow field smoothness constraints) in regions that do not contain salient textures (C) in Fig. 5. This regularization sometimes results in falsely occurring divergent behaviour and, as a result, false positives in the final segmentation (C) in Fig. 4i. Although similar false positives are observed on the tube phantom results for MF1 (Fig. 5g), the strength of the response is lower compared to the response at the centerline. We thus chose to use the third phantom to perform further quantitative comparison between the OF and MF estimation modules (presented in Fig. 5) by computing the error in flow endpoint $e_E$ to the ground truth $d_{GT}$ defined as

$$e_E(x, j) = \|d(x, j) - d_{GT}(x, j)\|_2$$

and the constituting errors in flow orientation $e_O$ and magnitude $e_M$ defined as

$$e_O(x, j) = \cos^{-1}\left(\frac{d(x, j) \cdot d_{GT}(x, j)}{\|d(x, j)\| \cdot \|d_{GT}(x, j)\|}\right)$$

$$e_M(x, j) = \|d(x, j)\| - \|d_{GT}(x, j)\|.\tag{15}$$

In the quantitative comparison of flow estimation methods, the OF2 method is the fastest method and ranks as second best in motion computation performance for all three error metrics. However, combined with the pulsatile radial motion model (PRMM), the motions extracted from OF2 fail to detect the small pulsating tubular structure (C) in Fig. 4i present in the noise phantom. Reducing the regularization weight $\alpha$ to 0.01 improves the detection of the small pulsating tubular structure, however we observed that regularization weights outside the range of $0.015 < \alpha < 0.1$ consequently result in noticeable reduction in motion estimation performance on synthetic data and segmentation accuracy on UBC dataset experiments presented in the following section.

3.4. Real Data Experiment

Initial real data evaluation is conducted on the UBC dataset, consisting of eight 30- to 40-frame DUS sequences of the CCA acquired along the transverse and longitudinal axes, where the vessel appears as a pulsating ellipsoid and tube respectively. Unlike the previous two-frame phantom experiment, the experiment with real data requires the temporal filtering stage to remove the high frequency noise and the low frequency motions (caused by breathing and small movements of the probe) that occur in the sequence. The passband of the temporal filter is tuned depending on the patient’s approximate heart-rate, denoted $\tau_r$. The parameters were set such that $\tau_L = \tau_r/2$ and $\tau_H = 2\tau_r$. The PBMS segmentation method and the four implementations of KMVS (Section 3.1) are then applied to the dataset using the same temporal filter parameters across all methods. All of the resulting fuzzy segmentation labels are shown in Fig. 7. To clarify the advantages of each approach as a trade-off between accuracy and computation time, in Fig. 8 we present quantitative analysis of segmentation error using the ground truth segmentations of the DUS sequences. The area under the receiver operating characteristics curve (AUC) for each case (thresholding the fuzzy segmentations from 0 to 1) is reported as a measure of segmentation accuracy, in which the

![Figure 6: Boxplots of endpoint errors, angular errors in degrees, and magnitude errors in pixels for each motion estimation methods in the KMVS pipeline. The corresponding mean of the errors are depicted with filled markers and black outline. The errors are computed at every pixel of the noise texture phantom (bottom row of Fig. 5) by comparing the estimated motions resulting from a two-frame matching experiment against corresponding ground truth motion vector values used to create the phantom. The mean errors and maximum errors at the 95 and 99.3 percentiles over all pixels are tabulated, with the best performing methods presented in bold. All differences in motion estimation performance are significant according to the Wilcoxon signed rank test ($p < 0.006$). The OF2 method is the fastest method in this experiment and ranks as second best in motion computation performance for all three error metrics.](image-url)
Proposed KMVS Implementations

Figure 7: Qualitative results of our experiment on the UBC dataset with yellow grid-lines superimposed to facilitate correspondence. Column (a): first frame of DUS sequences of CCA in axial and longitudinal axes including the bifurcation of internal and external carotid arteries. The corresponding US ground truth for the vessel is shown in red. Columns (b-f): color-coded fuzzy segmentation results of different implementations. The colors range from blue to red, representing weak to strong response to detected vessels. Segmentations are thresholded at 0.3 for visibility; responses below this threshold are colored black. (For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
value of 1 indicates perfect segmentation and 0.5 is the noise threshold.

Once the parameters of each implementation have been tuned on the phantom and UBC datasets, the real data experiments are repeated using the larger publicly available SPLab dataset. Only the first four frames of each sequence, recorded at an estimated three frames per second, were used to generate the results presented in Fig. 8 and 9. Due to the small number of frames and low frame rate of each sequence in this dataset, the temporal bandpass filter becomes a highpass filter that passes all of the temporal frequencies except for the zero frequency, DC, component.

In addition to the quantitative evaluation of segmentation errors using AUC, we also compare the results of each implementation by computing the Dice similarity coefficient (DSC) between the computed segmentations and corresponding ground truth of each dataset. To compute the DSC, the fuzzy segmentation labels generated using each implementation were binarized at a threshold of 0.5. The mean DSC and AUC of all cases in each dataset is computed and tabulated in Table 3. The summarized results indicate that the OF2 implementation of KMVS is the best performing method in terms of DSC and AUC. Detailed analysis of the presented results are carried out in the following section.

4. Discussions

Our experimental results show that all of the kinematic model-based vessel segmentation (KMVS) implementations outperform the phase-based motion segmentation (PBMS) approach in terms of segmentation accuracy. Compared to PB, with the addition of our proposed pulsatile radial motion model (PRMM), our monogenic flow (MF) and optical flow (OF) pipelines are more specific to motion of the common carotid artery (CCA) and resilient to motions that occur on the surrounding soft tissues. In the phantom experiments (Fig. 3), we explicitly showed how the PB implementation is only capable of detecting the translating structures (C) in Fig. 3 and the edges of the larger pulsating structures (G) in Fig. 3 that appear as local translations. On the other hand, the KVMS segmentation results presented in columns (c-f) of Fig. 3 are closer to center-line of the pulsating structures. This trend is also evident in the qualitative results of the real data experiments presented in Fig. 8 and 9 and is further substantiated by the quantitative analysis of segmentation accuracy presented in Fig. 10 and Table 3. However, contrary to our initial conclusions on real data experiments that favored MF methods over OF (Amir-Khalili et al., 2015a), we show that it is possible to outperform our proposed MF1 approach using the tuned OF2 algorithm.

The parameters selected for the MF1, MF2, and OF2 implementations were empirically tuned using only the phantom and UBC datasets originally presented in our KMVS paper (Amir-Khalili et al., 2015a) to minimize bias in the selection of parameters across different datasets within the same application domain. The automatically tuned OF1 implementation did not require any parameter tuning. These parameters were chosen based on visual assessment of the resulting segmentation and motion estimation quality, as well as the quantitative segmentation accuracy metrics used to evaluate our results. For both MF methods, we observed less than 1% change in the mean area under the receiver operating characteristics curve within 25% change in log-Gabor filter parameters k and δ. While selecting the parameters of OF2, we observed that the minimum wavelength λ_m, number of outer N_o, and inner N_i fixed point iterations mainly impacted the speed of the Lin (2009) algorithm used in OF2, while on the other hand, the regularizer weight α had a more direct impact on the results. We observed less than 1% change in the mean area under the receiver operating characteristics curve for 0.015 < α < 0.1.

Both OF implementations presented in this paper tend to perform well on dynamic ultrasound (DUS) sequences as the global flow smoothness constraint enables OF to approximate the motion of tissues in locations that are void of salient image information, i.e., the center of the vessel. This added constraint increases the computational complexity of the algorithm but, by comparing OF1 to a manually tuned OF2 (Fig. 8 and 10),
Figure 9: Eight cases presenting the best and worst results of our experiments on the publicly available SPLab dataset with yellow grid-lines superimposed to enhance correspondence. Column (a): first frame of DUS sequences of the CCA with the corresponding ground truth shown in red. Columns (b-f): color-coded fuzzy segmentation results of different implementations. The colors range from blue to red, representing weak to strong response to detected vessels. Segmentations are thresholded at 0.3 for visibility; responses below this threshold are colored black. The best and worst performing out of all 35 cases are framed in green and red respectively. For high resolution images and interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
we demonstrate that it is possible to obtain low segmentation error without increasing the run-time of OF motion computation. The motion estimation errors presented in Fig. 6 further confirm that the manually tuned OF2 implementation produces errors that are comparable to the automatically tuned OF1 implementation, while maintaining a run-time that is faster by almost two orders of magnitude across all phantom and real data experiments. The only notable difference between the two OF methods is that the mean angular error of OF2 is greater than that of OF1 and, as a result, OF1 is better at localizing the small pulsating tubular structure in our noise phantom compared to OF2 (⃝ in Fig. 4f). Regardless, OF2 outperforms OF1 on all real DUS data experiments (Fig. 8 and 10). From this observation, we conclude that although the more computationally expensive OF1 method generates better estimates of motion vectors, the much faster OF2 implementation performs better in tandem with our proposed PRMM technique and thus generates more accurate vessel localizations in DUS sequences.

The motion estimation method used in MF2 was originally developed to extract motion from medical imaging modalities, e.g. DUS and dynamic magnetic resonance (MR) imaging, in which the brightness consistency assumption does not hold (Alessandrini et al., 2013). Despite performing well on the synthetic phantom experiments, producing the lowest endpoint error among all methods (Fig. 6), MF2 does not perform as well as other KMVS methods on the real datasets. We hypothesize that this might be due to the fact that MF2 was specifically designed to compute large myocardial motion as opposed to the subtle motion of vascular walls such as the CCA.

The quantitative results presented in Fig. 8 and 10 also show that the MF1 method can, on average, achieve comparable accuracy with the OF1 method. The comparable performance of MF1 to other flow estimation methods is further confirmed by the analysis of the endpoint errors presented in Fig. 6. In terms of run-time, both OF2 and MF1 implementations are suitable for diagnostic applications, e.g., CCA segmentation, as they are projected to perform in near real-time with our imaging setup given an efficient implementation and a specialized workstation. MF1 is slower than OF2 in the two-frame phantom experiment (Fig. 6) and the experiment on the SPLab data (Fig. 10), but it performs faster on the UBC data (Fig. 8). This is attributed to the computational overhead cost of building the spherical quadrature filters (SQF) for each sequence. The MF1 method is designed in such a way that the filter bank containing the SQF pyramid is constructed only once per sequence and, as a result, this one-time computational cost dominates the overall run-time in cases where the number of frames are few, such as in the phantom and SPLab data experiments. In the UBC data experiments, where each sequence contains 30-40 frames, our MF1 implementation is faster than OF2. MF1 is also parallelizable as most of the computations performed are point-wise (pixel-wise) operations, which do not need a Horn and Schunck (1981) style global smoothness constraint. By precomputing the filter bank and porting the code to run on a graphical processing unit (GPU), it is possible to achieve further performance gains and enable the method to run in real-time on sequences with larger images (Pauwels and Van Hulle, 2008; Amir-Khalili et al., 2013).

We envision our segmentation pipeline to be deployed within an open software ultrasound platform, such as the BK Ultrasound (Analogic Corp, USA) Sonix system. GPU capabilities can be integrated directly into an OEM configured Sonix system or incorporated as part of a separate dedicated computing workstation linked to the Sonix system via available software development kits.

Our UBC dataset was recorded at 30 frames per second while the SPLab dataset was recorded at an estimated three frames per second. This translates to a recording window, and consequently a filtering lag time, of 1 to 1.3 seconds for both datasets. In our experiments we found this recording length to be empirically sufficient for localizing the CCA as it captures the periodic behavior of a patient’s resting heart-rate of 60 beats per minute (1 Hz) or above. The maximum computation time for our unoptimized MATLAB implementation of MF1 and OF2 methods brings the total lag time to approximately 10 seconds, which is reasonable in the context of CCA segmentation. With a direct implementation on a specialized ultrasound system or a dedicated workstation, this total computation time may be decreased by approximately an order of magnitude.

The proposed ideal temporal filtering scheme is suitable in the context of CCA segmentation as, in our experiments, it only contributed 1 to 1.3 seconds to the total 10 second lag time. To extend our methodology to applications that impose stricter constraints on temporal performance and lag times, e.g., intra-operative image-guidance systems, causal filters such as infinite impulse response filter used in Wu et al. (2012) and general biquad filter employed by McLeod et al. (2015a), or the extended Kalman filtering approach of McLeod et al. (2015b) may be incorporated as alternatives to ideal filtering. Tuning the pass-band of the proposed temporal filter currently requires an estimate of the patient’s heart-rate. The heart-rate can either be manually estimated by the imaging technician during acquisition, recorded using a heart-rate monitor or via the electrocardiography triggering capabilities that are built into advanced US machines, e.g., the BK Ultrasound Sonix system. However, if the probe is relatively still during acquisition and other motion artifacts are minimal, the passband of the filter may be fixed to a constant value between 0.5 - 4 Hz, which is sufficiently broad enough to encompass resting heart-rate frequencies rang-
ing from 60 - 120 beats per minute, as was demonstrated with the experiments on the SPLab dataset.

Although the OF2 pipeline outperforms both MF methods in our real data experiments, monogenic methods have other advantages that may be exploited to improve vessel localization. In the domain of US image processing, SQF have been shown to improve the extraction of structures, during radio frequency signal to B-mode conversion, by demodulating the radio frequency signal in a 2D context (Wachinger et al., 2011). This implies that MF methods may be used to extract local motion information from raw radio frequency data, allowing for a direct implementation in the native representation of acquired DUS data.

Another advantage is that the MF implementations presented may be extended to three spatial dimensions (Alessandrini et al., 2012). Our approach is not yet able to cope with gross out-of-plane motion, common during 2D+time DUS acquisitions, but such problems would not exist once the method is extended to process 3D+time sequences. A study on 3D steerable wavelets and the monogenic signal (Chenouard and Unser, 2011) concluded that the steerable approach of Portilla and Simoncelli (2000), which was used in the PBMS method, is hard to extend into 3D due to its invertibility but, on the other hand, the MF formulation can be extended to process 3D+time sequences (Chenouard and Unser, 2011; Alessandrini et al., 2012). Similarly, 3D+time extension of the OF implementations are also possible, e.g., 3D OF between two volumes followed by divergence computation of the estimated 3D vector field and multi-scale averaging. This will allow our proposed MF and OF implementations to be extended to volumetric medical images such as 3D+time CT fluoroscopy, Cine MR, and 3D DUS. However, due to the aforementioned fact that the MF1 implementation largely consists of point-wise operations, we hypothesize that a 3D MF implementation would outperform 3D Horn and Schunck (1981) style optical flow methods, on the other hand, do not scale as well in comparison as the spatial dimensionality of the data increases from 2D to 3D. As a result, point-wise motion estimation methods such as MF1 are poised to outperform competing methods as the use of volumetric dynamic imaging of vasculature becomes more prevalent in medical diagnostics and interventions.

5. Conclusions

In this paper, we assessed the performance of vessel localization using automatic segmentation algorithms that model pulsatile radial motion. We presented multiple effective implementations of a low-level motion-based segmentation pipeline that detects the characteristic pulsatile radial motion of vascular structures through the analysis of divergent motion vector fields extracted from a sequence of frames. Our proposed methods are focused on fast and accurate extraction and modeling of motion vector fields, without the aid of learning and appearance models, such that our segmentation labels may be incorporated, alongside complimentary low-level intensity based features, into existing high-level discriminative segmentation frameworks (Becker et al., 2013; Henmersperger et al., 2015).

In each implementation, we explored alternative optimized and off-the-shelf techniques for performing the motion estimation stage of the proposed pipeline. Through evaluation of all proposed methods, using synthetic and real dynamic ultrasound (DUS) sequences of the common carotid artery, we conclude that coupling a tuned optical flow motion estimation method with our pulsatile radial motion model provides the best overall performance compared to other candidates. Our experiments on two real DUS datasets show that the performance of the old phase-based motion segmentation method can be increased using the tuned OF2 implementation of kinematic model-based vessel segmentation from an average area under the receiver operating characteristics curve of 0.82 to 0.99 on the UBC dataset and from 0.83 to 0.98 on the SPLab dataset. Similarly, binarizing the resulting fuzzy segmentation labels at 0.5 yields a significant increase in Dice similarity coefficient from 0.11 to 0.72 on the UBC dataset and 0.20 to 0.67 on the SPLab dataset. Furthermore, the tuned OF2 implementation of the pipeline performs the fastest on short sequences compared to all other implementations, while our MF1 implementation boasts the fastest run-time on longer sequences due to the precomputation of the spherical quadrature filters.

The computational speed of our proposed MF1 motion estimation method can also greatly benefit from a parallel implementation that takes advantage of GPU computing (Amir-Khalili et al., 2013). The monogenic framework used to extract motions in the MF1 implementation may also be able to estimate motion from native RF data of DUS sequences (Wachinger et al., 2011) and can be extended to process volumetric dynamic (3D+time) data efficiently (Alessandrini et al., 2012; Chenouard and Unser, 2011). The Horn and Schunck (1981) style optical flow methods, on the other hand, do not scale as well in comparison as the spatial dimensionality of the data increases from 2D to 3D. As a result, point-wise motion estimation methods such as MF1 are poised to outperform competing methods as the use of volumetric dynamic imaging of vasculature becomes more prevalent in medical diagnostics and interventions.

References


