Low Cost 3D Flow Estimation in Medical Ultrasound

by

Siyuan Wei

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2018 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Chair
Antonia Papandreou-Suppappola
Umit Ogras
Thomas F. Wenisch

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Medical ultrasound imaging is widely used today because of it being non-invasive and cost-effective. Flow estimation helps in accurate diagnosis of vascular diseases and adds an important dimension to medical ultrasound imaging. Traditionally flow estimation is done using Doppler-based methods which only estimate velocity in the beam direction. Thus when blood vessels are close to being orthogonal to the beam direction, there are large errors in the estimation results. In this dissertation, a low cost blood flow estimation method that does not have the angle dependency of Doppler-based methods, is presented.

First, a velocity estimator based on speckle tracking and synthetic lateral phase is proposed for clutter-free blood flow. Speckle tracking is based on kernel matching and does not have any angle dependency. While velocity estimation in axial dimension is accurate, lateral velocity estimation is challenging due to reduced resolution and lack of phase information. This work presents a two tiered method which estimates the pixel level movement using sum-of-absolute difference, and then estimates the sub-pixel level using synthetic phase information in the lateral dimension. Such a method achieves highly accurate velocity estimation with reduced complexity compared to a cross correlation based method. The average bias of the proposed estimation method is less than 2% for plug flow and less than 7% for parabolic flow.

Blood is always accompanied by clutter which originates from vessel wall and surrounding tissues. As magnitude of the blood signal is usually 40-60 dB lower than magnitude of the clutter signal, clutter filtering is necessary before blood flow estimation. Clutter filters utilize the high magnitude and low frequency features of clutter signal to effectively remove them from the compound (blood + clutter) signal. Instead of low complexity FIR filter or high complexity SVD-based filters, here a power/subspace iteration based method is proposed for clutter filtering. Excellent

clutter filtering performance is achieved for both slow and fast moving clutters with lower complexity compared to SVD-based filters. For instance, use of the proposed method results in the bias being less than 8% and standard deviation being less than 12% for fast moving clutter when the beam-to-flow-angle is 90^o.

Third, a flow rate estimation method based on kernel power weighting is proposed. As the velocity estimator is a kernel-based method, the estimation accuracy degrades near the vessel boundary. In order to account for kernels that are not fully inside the vessel, fractional weights are given to these kernels based on their signal power. The proposed method achieves excellent flow rate estimation results with less than 8% bias for both slow and fast moving clutters.

The performance of the velocity estimator is also evaluated for challenging models. A 2D version of our two-tiered method is able to accurately estimate velocity vectors in a spinning disk as well as in a carotid bifurcation model, both of which are part of the synthetic aperture vector flow imaging (SA-VFI) challenge of 2018. In fact, the proposed method ranked 3rd in the challenge for testing dataset with carotid bifurcation. The flow estimation method is also evaluated for blood flow in vessels with stenosis. Simulation results show that the proposed method is able to estimate the flow rate with less than 9% bias.

ACKNOWLEDGMENTS

I am very grateful to my advisor Dr. Chaitali Chakrabarti for her technical guidance and patience. There were difficult times in my PhD journey when Dr. Charkrabarti encouraged me and inspired me with valuable suggestions. I also appreciate her integrity in both academics and teaching. I learned from her not only how to be a good scholar but also how to be a good person.

I would like to give thanks to my committee members: Dr. Antonia Papandreou-Suppapola, Dr. Umit Ogras and Dr. Thomas F. Wenisch. They provided valuable inputs to my research work. I am also grateful to the University of Michigan team, Dr. Thomas F. Wenisch, Dr. Brian Fowlkes and Dr. Oliver Kripfgans, for their valuable technical advice over long weekly phone metings, for providing constructive criticism of my research work and help in reviewing my papers.

I am also grateful to the National Science Foundation CCF-1406810 and the School of Electrical, Computer and Energy Engineering at ASU for funding my graduate study.

Over the years, many colleagues have graduated and many are continuing their journey in academics. Ming Yang, who introduced me into the medical ultrasound field and devoted a lot of time training me, was the trail blazer in the project. The work in this dissertation would not have been possible without his help. I also want to thank all that have helped and encouraged me in my PhD career: Jian Zhou, Richard Sampson, Brendan West, Rungroj Jintamethasawat, Chengen Yang, Qian Xu, Lifeng Miao, Manqing Mao, Shunyao Wu, just to name a few.

Last but not least, I am grateful for my family's support, especially my wife, who gave up her job and came to accompany me in the US.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

In recent years, non-communicable diseases (NCDs) have been known to cause more deaths than all other causes of deaths combined. The four major NCDs are cardiovascular diseases, cancer, chronic respiratory diseases and diabetes. Together they account for more than 80% of the total NCD deaths. Of the four categories, a study in [1] showed that in 2012, cardiovascular diseases caused 17.5 million deaths which is 31.2% of the total NCD deaths worldwide and twice the number of deaths caused by cancer [1].

To help doctors diagnose cardiovascular diseases at an early stage and improve the prognosis, full 3D blood flow field with sufficient spatial and temporal resolution is highly necessary. MRI and ultrasound imaging are both used in measuring blood flow. They help monitor anomalies in the flow volume and also nature of the flow. Although MRI is believed to have better accuracy and thus regarded as the gold standard, it suffers from long acquisition time, high cost and low portability [2]. This long acquisition time makes MRI not suitable for measuring the peak velocity, which provides crucial information for determining the extent of stenosis. Recent developments in high frame rate ultrasound imaging have enabled accurate blood flow estimation. Also, in vivo experiments show that ultrasonic flow estimation accuracy has become comparable to that of MRI [3, 4].

In traditional medical ultrasound systems, only velocities along ultrasound directions (1D) are estimated and these velocities are angle compensated based on the beam-to-flow angle. However, beam-to-flow angle is often hard to estimate for human vessels. This prompted the development of methods that can estimate 2D or 3D

velocity vectors reliably [5]. Methods which overcome the angle dependency of the 1D methods include the cross-beam vector Doppler method [6], transverse oscillation (TO) method [7, 8], and speckle tracking [9].

3D velocity vectors contain significantly higher diagnostic information than its 2D counterpart, as expected. For example, 3D flow imaging can facilitate precise calculation of volumetric flow across an arbitrary plane through a vessel [10]. In contrast, in 2D flow imaging, blood velocity estimation is less accurate when there is blood flowing out of the plane. Furthermore, collecting a complete velocity vector field within a 3D region of interest enables derivation of other important flow parameters, such as pressure drops associated with flow restrictions [5, 11, 12].

3D flow estimation requires high volumetric frame acquisition rate, which is usually in the order of kilo Hertz. Such a high frame rate results in significant increase in the computational complexity due to beamforming. In addition to the already high complexity of beamforming for 3D images [13–16], 3D velocity estimation typically requires many more firings and an order of magnitude more computations than a 2D system.

Our goal is to develop a low-cost 3D scheme for accurate blood flow estimation. Reducing the number of computations helps reduce the power consumption, making it possible to implement the proposed scheme in a portable platform. Portability is particularly important since nearly three quarters of the NCDs deaths happen in low- and middle- income countries [1], and a portable ultrasound machine could facilitate fast access to medical help.

1.1 Blood Velocity Estimation in Medical Ultrasound

3D blood flow estimation using ultrasound has recently become a subject of active research [17, 18]. It has a high clinical value in diagnosing vascular diseases such

as stenosis, thereby reducing the risk of vascular deaths [19]. However, 3D flow estimation is quite challenging, as its computational complexity is drastically higher than its 2D counterpart [18].

Traditionally, blood velocity estimation methods have been based on autocorrelation methods, such as Kasai estimator and 2D autocorrelation estimator [20]. These methods only estimate axial velocity (along the ultrasonic beam direction) and thus the estimation accuracy of the primary flow becomes poor when the beam-to-flow angle approaches 90^o.

Several methods have been proposed to overcome this angle dependency and enable accurate estimation of 2D or 3D velocity vectors. For example, the 2D cross-beam vector Doppler method uses multiple sub-apertures and combines multiple Doppler measurements to obtain both lateral and axial velocities [6]. Recently, a 3D cross beam multiple receiver scheme has been used for estimation of low velocity flows [21]. Complex receive schemes that introduce transverse oscillation (TO) [7, 8] have been proposed that enable 2D and even 3D [22] velocity vectors to be estimated using the basic Doppler principle. The TO method has been combined with plane-wave imaging to provide high frame rate 2D flow imaging [23]. SARUS, a synthetic aperture 3D ultrasound system [24] that supports flow estimation is based on the TO method.

Other velocity estimation methods use time-delay estimators, such as cross correlation, to estimate motion. The directional beamforming method generates beamforming lines in the flow direction and estimates velocity using cross correlation [25]. However, this method requires apriori knowledge of the beam-to-flow angle.

Speckle tracking is among the earliest motion estimation methods used in ultrasound. It does not require prior knowledge of the flow angle [26]. Since the blood speckle is correlated with blood movement, kernel matching techniques can be used to track blood velocity. For example, a 2D blood velocity estimation based on speckle

tracking with plane wave imaging has been shown to achieve a frame rate of 100 vector flow images (VFIs) per second with a packet size of 40 [9]. Estimation robustness can be improved by compounding the speckle tracking results in dual angle plane-wave imaging [27]. Speckle tracking methods have also been used for estimation of complex flows generated through computational fluid dynamics (CFD) simulations [28, 29].

With the development of fast 3D imaging, speckle tracking can be naturally extended to handle 3D motion estimation. In fact, fast 3D imaging [30] has been actively investigated and several systems have been developed in recent years. For instance, [31] presents a customized 3D ultrasound imaging system, with a 32×32 matrix probe, that supports flow imaging with high frame rates. Despite challenges due to large computational complexity, these developments have made speckle tracking a promising technique for real time flow estimation [18]. There are other flow estimation techniques that utilize microbubble contrast agents to enhance the contrast-to-noise ratio between the flow containing microbubbles and surrounding static tissue responses [32]. Because many of these techniques are also based on use of autocorrelation or cross correlation or otherwise rely on pulse to pulse signal coherence, the instability of the ultrasonic response of microbubbles puts a constraint on the ensemble length, resulting in lower accuracy [33, 34]. A Fourier-based velocimetry that uses estimates over a range of frequencies for higher estimation accuracy is proposed in [35]. It has shown improvement when using microbubble as contrast agents.

In this work, we focus on improving the accuracy of 3D blood flow estimation with minimum increase in the computational complexity. We present a low complexity 3D flow esimtaitor and show that it achieves high estimation accuracy for simple (plug, parabolic) as well as complex (spinning disk, carotid bifurcation, vessel with stenosis) flows.

1.2 Clutter Filter Design for Blood Flow Estimation

Blood signals acquired in vivo are accompanied by clutter signals. These originate from the vessel wall and the surrounding tissue and usually have high amplitude and low frequency [20]. For accurate flow estimation, it is important to first remove the clutter before estimating flow velocity. Clutter filters work on the slow time signal of a sample volume within a packet. These filters have traditionally been high-pass filters. However, when the tissue surrounding the blood vessel moves, the Doppler frequency of the clutter is no longer centered at zero. In such cases, traditional filters that have fixed frequency response tend to remove part of the blood signal, which decreases the accuracy of flow estimation. Thus, while the traditional high pass filters have low computational cost, they are not robust and their performance is not acceptable.

In a realistic scenario, clutter and blood velocities change over time, resulting in changes in the spectrum of blood and clutter signals. This makes adaptive clutter filters necessary for good clutter removal performance. One approach [36] is to downmix the signal to move the mean Doppler frequency of the clutter to zero before performing high pass filtering. However, the filter performance degrades if the stopband is not chosen properly [37].

Another important class of adaptive clutter filter is the eigen-based clutter filter. Eigen-based clutter filter has high adaptivity to the slow time signal. Bjaerum et al. [38] proposed a eigen regression filter that uses the eigen decomposition on the slow time signal. Yu and Cobbold [39] proposed an approach called Hankel-SVD which uses Singular Value Decomposition (SVD) on the Hankel matrix constructed using the slow time signal. Despite their excellent filter performance, eigen-based filters suffer from the high complexity of large SVD computations, which are of the order $O(n^3)$ [40]. Our goal is to design clutter filters that have significantly lower

computational complexity while still producing comparable performance to existing high-end systems.

1.3 Problems Addressed

Our flow estimation scheme is developed for a plane wave imaging system such as the one in [16]. The block diagram of our system is shown in Fig 2.1. In each firing, a 32×32 receive aperture is used to emulate an unfocused plane wave that propagates through the imaging region. The beamforming process used here is based on separable decomposition of delay calculation into two separate directions. It can generate RF data with a frame rate of up to 6,000 frames/s. This data is then processed in packets by flow estimation units (shaded in the block diagram). Since the input is blood contaminated with clutter, first the clutter filter removes the clutter. Next, the clutter-filtered data is used to estimate 3D velocity vectors. The velocity vectors obtained with motion estimation are then used in the flow rate estimation. In the rest of this section, we summarize the specific problems that were addressed in this work along with the findings.

1.3.1 Low Cost Blood Velocity Estimator with Sub-pixel Accuracy

Blood flow estimation is traditionally done using the Doppler method. This method estimates the velocity only in beam direction and estimates the lateral velocity based on prior knowledge of the beam-to-flow angle. There are "real" 2D velocity estimation schemes that obtain axial and lateral velocities independently and thus are able to estimate the beam-to-flow angle. Moreover, 3D velocity estimation takes the out-of-plane velocity into consideration and thus provides full velocity profiles which enables flow rate estimation at arbitrary cross-section planes of the vessel.

Speckle tracking is a velocity estimation technique which estimates the velocity by tracking the movement of a kernel by using a goodness-of-match algorithm. Among the goodness-of-match schemes used in speckle tracking, cross correlation is much more computationally complex compared to sum-of-absolute-difference (SAD) but usually has higher accuracy. Fortunately, as a 3D kernel includes more samples than a 2D kernel, the performance difference between SAD and cross correlation based methods is largely reduced and so SAD is usually sufficient for pixel level estimation. However, in 3D speckle tracking with plane-wave imaging, lateral image resolution remains poor and sub-pixel estimation is usually needed. Sub-pixel estimation can be obtained by locating the zero-crossing positions of the phases of the cross correlation functions. Unfortunately, there is no natural phase in the lateral dimension and synthetic phase has to be generated through spectrum separation.

We proposed a two-tiered scheme to combine the low complexity SAD estimator and a 3D version of the synthetic lateral phase method [41] to provide accurate blood velocity estimation with sub-pixel accuracy. This method was presented in [42]. We were able to achieve an average bias of less than 2.1% and average standard deviation of less than 9.3% for plug flow. We also achieved less than 9% standard deviation for parabolic flow. We further improved the sub-pixel accuracy by correcting phase of cross correlation based on autocorrelation in [43].

1.3.2 Low Cost Eigen-based Clutter Filter

In medical ultrasound, after beamforming, the clutter signal which originates from vessel wall and surrounding tissues usually has 40-60 dB higher magnitude than the blood signal. Therefore, a clutter filter is needed before any blood velocity estimation can be done. Clutter is conventionally removed by high pass filter as clutter usually

has lower frequency. However, a high pass filter with fixed stopband does not account for the changes in clutter and blood movement.

Eigen-based clutter filter, on the other hand, utilizes both the high magnitude and low frequency features of clutter and adapts to the data. It has better clutter removal performance than traditional FIR or IIR filter but has much higher complexity due to the computationally expensive singular value decomposition (SVD) [37, 39]. The computational cost of SVD can be reduced if only the first several largest singular values and corresponding vectors (i.e. subspaces) are needed. These subspaces can then be removed by subtraction from the original signal.

We proposed a low-cost eigen-based clutter filter method [43], which can remove one or more subspaces. This method can handle both slow moving clutter caused by respiratory motion and fast moving clutter caused by pulsatile motion. Slow moving clutter corresponds to the case when the clutter velocity is 0.5% of the peak blood velocity and fast moving clutter corresponds to the case when the clutter velocity is 10% of the peak blood velocity. We validated our techniques through Field-II [44] simulations for both slow and fast moving clutter, and for 60^o and 90^o beam-to-flow angles. Our simulation results showed that the proposed clutter filters attenuated the clutter significantly with excellent post-filter clutter-to-blood ratio (CBR). It also provided good blood velocity estimation results, comparable to the case where clutter was not included in the simulations. For both slow and fast moving clutters, the average velocity estimation bias was within 7.5% and the average standard deviation was within 15.7% for parabolic flow. The clutter filter work was presented in [43, 45].

1.3.3 Flow Rate Estimation for Parabolic Flow

Flow rate can be calculated by integrating the estimated velocities on a cross section plane. As speckle tracking is a kernel based method, some of the kernels that

are close to the vessel boundaries are not fully inside the vessel, and the velocity esimation is noisier for such kernels. Therefore, for accurate flow rate estimation, we use a weighting of kernel power that assigns fractional weights to kernels that are not fully inside the vessel, thereby improving overall flow rate estimates.

We consider estimation of blood flow through vessels that can be represented by a straight tube. The flow through such vessels is parabolic in nature. Field-II simulation results show that our blood velocity estimation scheme is quite accurate, with less than 8% average bias for both slow and fast moving clutter. The average standard deviation of the estimation is smaller for 90^o scenario ($< 12\%$) than that of the 60^o scenario ($< 16\%$). Volumetric flow rate estimation is also quite accurate. For a beam-to-flow angle of 90^o, the estimation bias is 8.2%, and the standard deviation is 5.6% for slow moving clutter; the bias is 8.8% and the standard deviation is 3.1% for fast moving clutter. The proposed flow rate estimation method appeared in [43].

1.3.4 Flow Estimation in Challenging Models

As the blood vessel is never a perfect cylinder, it is important to test our flow estimation method on more challenging flow models. Therefore, we validated the performance of our method with spinning disk and carotid bifurcation models. These models are have complex geometry and require the flow estimation to be accurate over a wide range of angles.

Spinning disk and carotid bifurcation models were obtained from the synthetic aperture vector flow imaging (SA-VFI) challenge held as part of the International Ultrasonic Symposium 2018 (IUS18) conference. Since this was a 2D velocity estimation challenge with synthetic aperture imaging, we had to modify our 3D method designed for plane wave imaging. The key changes were deriving a 2D version of the two-tiered method, accounting for aliasing in the phase of the correlation function

and then application of median filtering on the final estimates. The velocity profiles estimated with our method closely matched the ground truth for all datasets in the challenge. For instance, the average magnitude bias was less than 10% and the average angle bias was less than 10^o for spinning disk.

We also considered velocity estimation in vessels with stenosis. We generated these models using COMSOL. For both single stenosis and double stenosis models, the velocity estimations were quite accurate except when the velocity gradient was large near the stenosis boundary. We showed that by reducing the kernel size and increasing the packet size, the bias can be reduced at the stenosis boundary without increasing the standard deviation. The flow rate estimation for this model was also quite accurate, with less than 6% bias.

1.3.5 Post-processing in Plane Wave Imaging Challenge in Medical Ultrasound (PICMUS)

Plane wave imaging is a promising imaging modality for medical ultrasound as it has fast acquisition rate. In fact, our 3D flow estimation is based on plane wave imaging. The imaging quality of plane wave imaging can be improved by coherent compounding [30]. We participated in the PICMUS challenge held as part of International Ultrasonic Symposium 2016 (IUS16). The goal of the challenge was to compare the performance of different beamforming methods for plane wave imaging with or without compounding.

We chose delay-and-sum (DAS) beamforming and applied a post processing method based on edge detection. The imaging results showed that a proper choice of parameters in beamforming (f-number, apodization window, etc.) along with our post processing method helped improve the contrast ratio of the cysts without affecting other

parts of the image. Compared to the results posted by other groups, our method had superior contrast ratio for cysts but lower resolution for point scatterers.

1.4 Outline

This dissertation is organized as follows.

In Chapter 2, we describe our two-tiered motion estimation method SAD+SLP-3D, designed to lower the complexity of our sub-pixel accurate scheme while improving the accuracy. Techniques to further reduce the computational complexity are described and the overall complexity is analyzed.

In Chapter 3, we introduce different approaches to reduce the complexity of eigen-based clutter filter. Their performance based on post-filter clutter-to-blood ratio (post-CBR) and computational complexities are compared.

In Chapter 4, we use flow rate calculation method based on power weighting to estimate overall flow rate estimation. The performance of this system is evaluated in detail.

In Chapter 5, we apply 2D or 3D version of our two-tiered velocity estimator to challenging models, including spinning disk, carotid bifurcation and vessels with stenosis. Performance of our flow estimation system when applied to each model is presented.

In Chapter 6, we describe our performance in the PICMUS challenge of 2016. We show that the post processing method improved the contrast ratio of cysts.

In Chapter 7, we conclude this work and discuss future challenges.

Chapter 2

3D BLOOD VELOCITY ESTIMATION

As mentioned in Chapter 1, there are two main approaches for 3D blood flow estimation. One is based on the traditional Doppler approach with special beamforming schemes to create Doppler frequency in lateral dimensions [22, 46] and the other is based on time delay estimators usually coupled with kernel based algorithms. Regardless of the methods used, blood flow estimation requires high image acquisition rate, in order to ensure coherence between consecutive frames. Plane wave imaging is a suitable candidate, especially for kernel based algorithms, as it generates full 3D images at a high frame rate.

Previously, our group had developed a low cost plane wave imaging system which can support 6000 frames/s within the 5 W power constraint of a handheld device [16]. The low complexity was achieved by using separable beamforming [15] and 3D stacking architecture [14]. In this work, we make use of this high frame rate plane wave imaging system.

In this chapter, we focus on the motion estimation module based on speckle tracking, as shown in Fig. 2.1. The challenges of speckle tracking in 3D include poor sub-pixel accuracy and high computational complexity. We propose to use synthetic lateral phase (SLP) technique and a two tiered approach to achieve low cost blood velocity estimation with sub-pixel accuracy.

2.1 3D Speckle Tracking

Speckle tracking estimates motion across frames by searching for the best match of a kernel in the reference frame with candidate kernels in the search region (usually

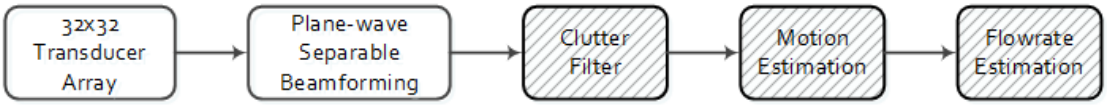


Figure 2.1: System Block Diagram

centered around each kernel location) in a previous frame. Candidate kernel locations are evaluated with a goodness-of-match cost function; typically, this cost function is a time-delay estimator (TDE), such as Sum-of-Absolute-Difference (SAD), Sum-of-Squared-Difference (SSD), or normalized cross correlation (NCC) [47]. Among these, SAD has the lowest computational complexity and NCC is the most computational demanding. Their performances in pixel level are comparable in most cases [47], but NCC can preserve phase information which allows more accurate sub-pixel estimation [41].

3D speckle tracking has advantages over its 2D counterpart because it does not suffer from speckle decorrelation caused by out-of-plane motion [48]. Assume $x_0(i, j, k)$ is a pixel in the kernel region X_0, and $x_1(i + \alpha, j + \beta, k + \gamma)$ is a pixel in the candidate region X_1 that is displaced by (α, β, γ) compared to X_0. SAD, SSD and NCC estimates are computed using equations (2.1), (2.2) and (2.3), respectively. Here $\bar{X}_0$ and $\bar{X}_1$ are the average pixel values, and $\sigma(X_0)$ and $\sigma(X_1)$ are the standard deviation of pixels in kernel regions X_0 and X_1. The estimated motion vector $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$, corresponding to the best match candidate across two frames, is averaged across the frames in a packet to give the motion estimation of the kernel of interest.

When using speckle tracking for flow estimation in a single transmit plane-wave system, the TDE estimators provide limited accuracy in lateral dimensions as the space between scanlines is quite large (λ) and the lateral resolution is poor due to lack of transmit focusing. To improve lateral accuracy, methods based on linear interpolation [49], and polynomial fitting [50] have been studied for 2D systems.

$$\varepsilon_{SAD}(\alpha,\beta,\gamma) = \sum_i \sum_j \sum_k |x_0(i,j,k) - x_1(i+\alpha, j+\beta, k+\gamma)| \tag{2.1}$$

$$\varepsilon_{SSD}(\alpha,\beta,\gamma) = \sum_i \sum_j \sum_k (x_0(i,j,k) - x_1(i+\alpha, j+\beta, k+\gamma))^2 \tag{2.2}$$

$$\rho_{NCC}(\alpha,\beta,\gamma) = CC(X_0, X_1)/(\sigma(X_0)\sigma(X_1)), \quad \text{where} \tag{2.3}$$

$$CC(X_0, X_1) = \sum_i \sum_j \sum_k (x_0(i,j,k) - \bar{X}_0)(x_1(i+\alpha, j+\beta, k+\gamma) - \bar{X}_1)$$

$$\sigma(X_0) = \sqrt{(\sum_i \sum_j \sum_k (x_0(i,j,k) - \bar{X}_0)^2)};$$

$$\sigma(X_1) = \sqrt{(\sum_i \sum_j \sum_k (x_1(i+\alpha, j+\beta, k+\gamma) - \bar{X}_1)^2)}$$

However, the improvement in sub-pixel accuracy is quite limited. The Synthetic Lateral Phase (SLP) technique [41] estimates the sub-pixel movement by locating the fractional position where the phase of the correlation function is zero. This is determined by finding the location where the magnitude of the cross correlation function is maximized. In lateral dimension, the spectrum is centered at DC, and no phase information is present. Artificial phases must be created by splitting the spectrum into halves so that the spectrums are no longer centered at DC. Next, we describe a 3D extension of this method, named SLP-3D, for estimating the phases of the cross correlation functions and using these to derive a motion vector with sub-pixel granularity.

2.2 Synthetic Lateral Phase in 3D

In this section, we extend the Synthetic Lateral Phase algorithm [41] designed for 2D speckle tracking to the 3D case (SLP-3D). In the SLP-3D algorithm, the

pixel-level estimates of 3D motion vectors are calculated using the cross-correlation functions, while the sub-pixel level estimates are obtained by locating the zero phase positions. This algorithm is substantiated by the fact that the magnitude of the cross-correlation function is maximum at the point where its phase is zero. To estimate the sub-pixel level motion using the phase-based zero crossing methods, we need phase information, which is present only in the analytical signal. In the axial dimension, the analytical signal is obtained by taking only the positive half of the 3D FFT along that dimension (4 out of 8 quadrants) and phase-based zero crossing is used to determine the sub-pixel motion. However, there is no phase information in lateral dimensions as there are no carrier frequencies in those dimensions. Thus, phase has to be generated artificially using Synthetic Lateral Phase method.

The proposed SLP-3D algorithm is shown in Algorithm 1. Note that Step 1 is done for each volumetric frame while Steps 2 and 3 are done for each frame pair. In Step 1, first 3D FFT of the beamformed RF data is computed. Then the spectrum is split in both lateral dimensions, creating four spectrum quadrants as shown in Fig. 2.2. The four sets of frame-pairs are used to compute complex cross correlation functions ($\rho_{x+,y+}$, $\rho_{x+,y-}$, $\rho_{x-,y+}$, $\rho_{x-,y-}$) based on equation (2.3). Each of these functions can be represented by $\Gamma e^{-i\omega}$, where Γ is the magnitude and ω is the net frequency. The axial, lateral-x and lateral-y cross-correlation functions, ρ_{axial}, ρ_{lat-x}, ρ_{lat-y}, are computed by multiplying the four cross correlation functions (or their conjugates) as shown in equation (2.4). Here, * stands for complex conjugate.

$$\begin{aligned}
\rho_{axial} &= (\rho_{x+,y+})(\rho_{x+,y-})(\rho_{x-,y+})(\rho_{x-,y-}) \\
\rho_{lat-x} &= (\rho_{x+,y+})(\rho_{x+,y-})((\rho_{x-,y+})(\rho_{x-,y-}))^* \\
\rho_{lat-y} &= (\rho_{x+,y+})(\rho_{x-,y+})((\rho_{x+,y-})(\rho_{x-,y-}))^*
\end{aligned} \tag{2.4}$$

Algorithm 1: Procedure of SLP-3D algorithm

Step 1 Preprocessing

- Obtain the spectra of the frame by computing 3D FFT
- Split each analytical spectrum into halves in both lateral dimensions, resulting in four spectra
- Compute the Inverse FFT of the spectra to obtain four volume pairs: $V_{x+,y+}$, $V_{x+,y-}$, $V_{x-,y+}$, $V_{x-,y-}$, where $V_{x+,y+}$ corresponds to the spectrum with positive frequencies in both x and y dimensions, $V_{x+,y-}$ corresponds to the spectrum with positive frequencies in x dimension and negative frequencies in y dimension, and so on.

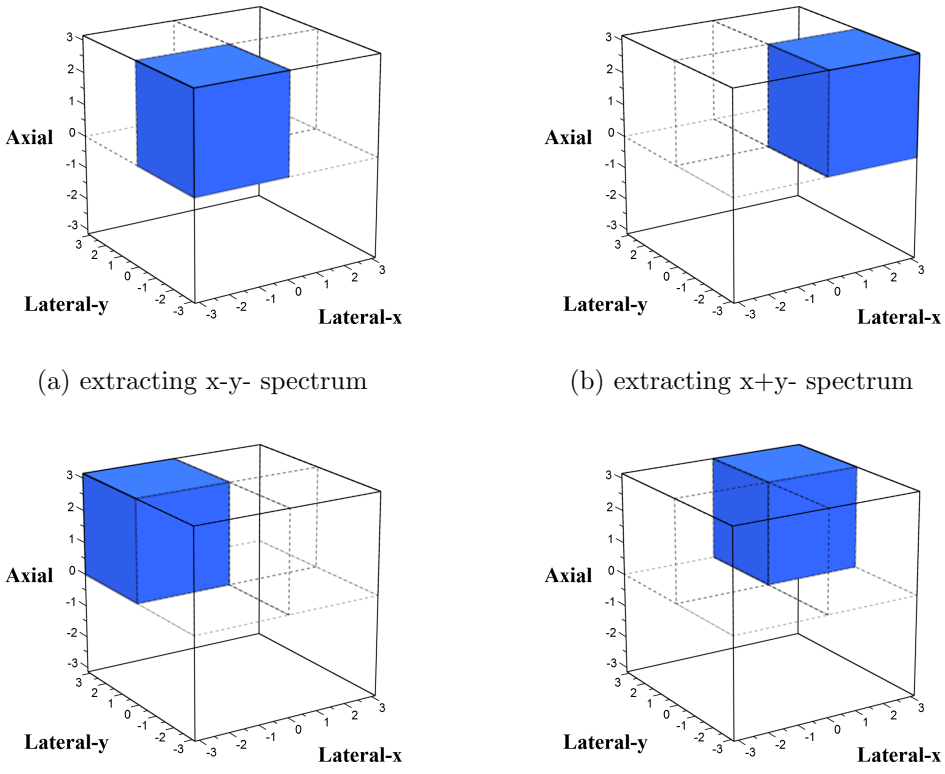
Step 2 Pixel level estimation

- Compute cross correlations of four volume pairs, resulting in $\rho_{x+,y+}$, $\rho_{x+,y-}$, $\rho_{x-,y+}$, $\rho_{x-,y-}$
- Compute ρ_{axial}, ρ_{lat-x}, ρ_{lat-y} based on equation (2.4)
- Search for the peak using one of the three cross correlation functions

Step 3 Sub-pixel level estimation

- Correct phases of cross-correlation functions based on autocorrelations
- Locate the zero-phase position in each dimension

Utilizing the relation between the frequencies in $\rho_{x+,y+}$, $\rho_{x+,y-}$, $\rho_{x-,y+}$, $\rho_{x-,y-}$ in a way that is very similar to the 2D method in [41], we ensure that ρ_{axial}, ρ_{lat-x}, ρ_{lat-y} only have nonzero phase in one dimension. For instance, ρ_{axial} only has nonzero



(a) extracting x-y- spectrum

(b) extracting x+y- spectrum

(c) extracting x-y+ spectrum

(d) extracting x+y+ spectrum

Figure 2.2: Separating the Spectrum into Four Quadrants in SLP-3D

phase in the axial dimension and zero phases in both lateral dimensions. For pixel-level estimation, it is sufficient to search for the peak of the cross-correlation function using only one of ρ_{axial}, ρ_{lat-x}, ρ_{lat-y}, as they have the same magnitude. The sub-pixel level motion vector is then estimated using the synthetic phase information. The phase variation is approximated using a linear function model [41] and the zero-phase location along each dimension is the zero-crossing point of the fitted line.

2.2.1 Computational Complexity

To provide a single, simple complexity measure for each method, we use a normalized computation metric: multiplication is considered the reference operation with a

weight of 1, addition has a weight of 1/8, and division and square root operations both have weight of 6. The weights have been chosen to roughly match the relative costs of the different arithmetic operations in terms of energy and area in our hardware design.

For a 3D frame of size 512×32×32 samples, computing the FFT and IFFTs (Step 1) requires 2×10^8 multiplications and 3×10^8 additions per frame. Assuming that there are 100 kernels in each 3D frame and for each kernel in the reference frame there are 387 candidate kernels, computing the correlation functions (Step 2) requires 387×100 correlation functions, resulting in 2.38×10^9 multiplications, 4.17×10^9 additions and 1.55×10^5 divisions. In the sub-pixel level estimation step (Step 3), each estimate requires 18 additions, 3 divisions and 3 square root operations. If there are 100 kernels in each 3D frame, this corresponds to 3,825 normalized computations. In total, about 3.14×10^9 normalized computations are needed per frame, and Step 2 is dominant with more than 90% of the total computations.

Despite the excellent performance of SLP-3D in estimating motion at sub-pixel resolution, this method suffers from high computational complexity. In the next section, we describe optimizations that drastically lower complexity without compromising on the accuracy of naive SLP-3D.

2.3 Two-tiered Approach

As the cross correlations consume most of the computations in SLP-3D, we first consider techniques to reduce the number of cross correlations. We propose a two-tiered approach [42] that uses SAD for coarse-grained motion estimation and SLP-3D for fine tuning the estimate. This approach combines the advantages of low complexity SAD with the sub-pixel accuracy of SLP-3D. Here, SAD is used to locate the pixel-level movement, so that the search region for cross correlation is narrowed. Only the

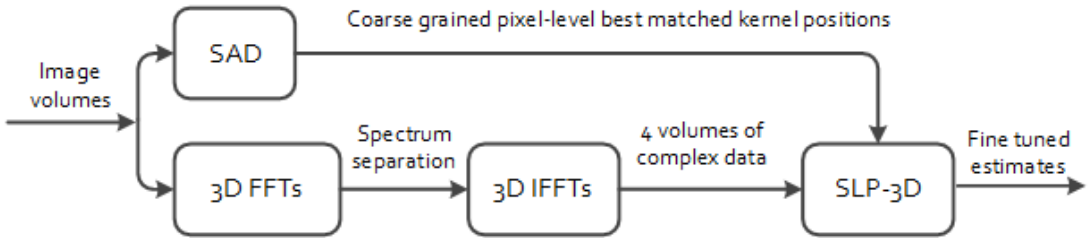


Figure 2.3: Block Diagram of the Two-tiered Velocity Estimation Method (SAD+SLP-3D)

nearest-neighbor candidates around the SAD best match are used in the sub-pixel estimation. A block diagram of the proposed method is shown in Fig. 2.3.

2.3.1 Computational Complexity

For the setup given in Table 2.1, if there are 100 kernels in the reference frame of size 512×32×32, the two-tiered method (SAD+SLP-3D) requires 2.23×10^8 additions for SAD, 2×10^8 multiplications and 3×10^8 additions for computing FFT and IFFTs, and 1.66×10^8 multiplications, 2.94×10^8 additions, 1.11×10^4 divisions and 1.11×10^4 square root operations for computing the cross correlation functions. This results in a total of 6.03×10^8 normalized computations per frame. While this method has significantly lower complexity compared to naive SLP-3D, it is still quite high. Next, we describe techniques to reduce the computational complexity of SLP-3D with minimal degradation in estimation performance. We refer to this method as SAD+SLP-3Dopt.

2.3.2 Reducing Complexity of FFT & IFFT

We reduce transform complexity by exploiting the fact that many samples used in the FFT and IFFT calculations are zeroes. For 3D FFT, we compute 1D FFT of 512 points along the z dimension first, followed by a 2D FFT along the x-y plane

Table 2.1: Simulation Environment

Parameter	**Slow moving clutter**	**Fast moving clutter**
Speed of sound	1540 m/s	1540 m/s
Transmit aperture pitch	0.5λ	0.5λ
Transmit aperture size (2D)	128×128	128×128
Receive aperture pitch	λ	λ
Receive aperture size (2D)	32×32	32×32
Center frequency (f_0)	4 MHz	4 MHz
6 dB Bandwidth	2 MHz	2 MHz
Sampling frequency (f_s)	40 MHz	40 MHz
Vessel radius (R)	3.5 mm	3.5 mm
Vessel center depth	15 mm	15 mm
Packet size	32	32
Beam-to-flow angle	$60^o/90^o$	$60^o/90^o$
Blood peak velocity (v_0)	1 m/s	0.3 m/s
Clutter velocity (v_c)	5 mm/s	30 mm/s
Pulse repetition frequency (f_{prf})	5 kHz	2 kHz
Clutter-to-blood ratio	40 dB	40 dB
Vibration frequency (f_{vib})	1 Hz	1.5 Hz

(1D FFT along x and 1D FFT along y). Since the spectrum corresponding to the negative frequencies has to be set to zero, half of the 3D volume of intermediate data is 0. Thus, we only need to compute 2D FFT of 256 x-y planes instead of 512 x-y planes. Similarly, when we compute the IFFTs, we compute the 1D IFFT along the z dimension first. Since in each case, three quarters of the data are zeroes, the 1D IFFT are also zeroes and need to be computed. Therefore, we can save half of the computations in the second stage where we compute the 1D IFFT along x dimension. As a result of these reductions, the total number of multiplications for computing FFT and IFFTs is reduced from 2×10^8 to 1.11×10^8, while the total number of additions is reduced from 3×10^8 to 1.67×10^8. This corresponds to a 45% reduction in normalized computations, from 2.38×10^8 to 1.32×10^8.

2.3.3 Reducing Complexity of Correlation Function

In order to reduce the number of computations for correlation functions, we first reduce the kernel size from 77×5×5 to 25×3×3. This helps achieve about 8×savings in the number of computations for correlation functions at the cost of increasing the standard deviation by 2-3%. Furthermore, we omit computing $\sigma(X_0)$ in Eqn. 2.3, since it is a common factor for all the candidates, and compute only once the normalized factors for correlation functions ρ_{axial}, ρ_{lat-x}, ρ_{lat-y}, since they are the same. Finally, we avoid the square root operations required to compute the normalization factor and replace it by finding the maximum of the square of the correlation function. Since squaring destroys the phase information required to locate the zero phase position in Step 3 of SLP-3D, we first calculate and store the phase via the arctan function.

Fig. 2.4 illustrates the overall reduction in complexity in terms normalized computations. The complexity of beamforming and SAD is the same for the three methods.

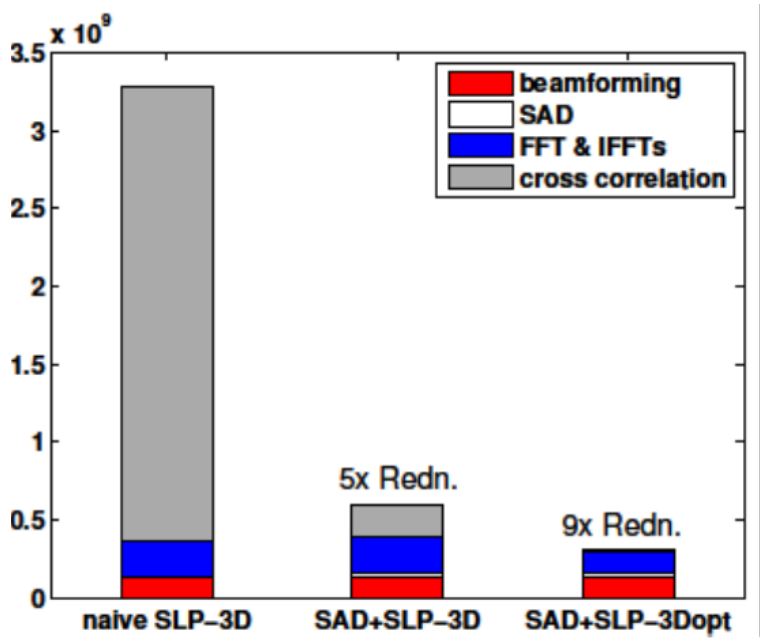


Figure 2.4: Average Number of Normalized Computations for 100 Velocity Estimates per Frame.

We were able to save about 48% if the normalized computations through the techniques described above. Together with the two-tiered approach, the total complexity reduction is about 9×. The complexity numbers were derived assuming that of the four main computation blocks, beamforming and FFT & IFFTs are done for each volume, while SAD and NCC are done for every motion vector estimate.

2.4 Phase Correction

While the SLP-3D method results in fairly accurate velocity estimates for plug flow [42], for parabolic flow, the bias can be as large as 12.77% as shown in Section 2.5. We found that the bias is due to the phase errors introduced by speckle tracking. Speckle tracking is a kernel based method, and thus, for parabolic flow or any other flow conditions in which the velocities of each pixel within a kernel are not the same, decorrelation between the kernels results in phase error [26].

Phase error also exists in autocorrelation, namely correlation between the kernel and its neighboring kernels in the same frame. In the spatial domain, in theory, the phase of autocorrelation of lag 0 should be zero, while the phases of autocorrelation of lag ± 1 should be symmetric (same value, different sign). However, in practice, the phase of autocorrelation of lag ± 1 are not always symmetric, especially in lateral dimensions. Assume the phases of autocorrelation of lag ± 1 are α and β, compensation of $\Delta = ||\alpha| - |\beta||/2$ is needed to make the phases symmetric. This compensation, Δ, can then be applied to the phases of the corresponding cross correlation functions to reduce the errors. Note that in the axial dimension the phases are quite accurate, and so we propose to correct only the phases of the cross-correlation functions (between frame i and frame $i + 1$) in lateral dimensions as follows:

- Compute autocorrelation for frame i, at lags of ± 1.
- Compute the phase compensation amount (Δ) needed to make the phases symmetric at lags of ± 1 for autocorrelations.
- For the pair of frames(i,$i + 1$), if the peak of the cross-correlation functions corresponds to lag 0, then apply Δ to the cross-correlation functions at lags of ± 1.

Phase correction is done for each pair of frames to ensure accuracy. For each kernel, four autocorrelations are needed in addition to the 27 cross-correlation functions. Since the complexity of autocorrelation is the same as cross correlation, this results in a 15% increase in complexity. Considering the packet size of 32, the total computation breakup for a packet is as follows. SAD requires 886M (million) normalized computations (15%), FFT and IFFTs needed 4224M normalized computations (75%) and cross correlation and autocorrelation need 558M normalized computations (10%). Therefore, the number of additional computations contributes to only 0.5%

additional complexity of the full system. The overall complexity of SAD+SLP-3Dopt is about 1.91× of the complexity of our B-mode imaging system [16] which can achieve a frame rate of 6,000 3D frames/s. In theory, this enables us to support flow imaging with frame rate up to 95 velocity estimations/s when packet size is 32.

2.5 Simulation Results

In this section we evaluate the performance of the optimized two-tiered method (SAD+SLP-3Dopt). We use Field II [44, 51] in MATLAB with the parameters listed in Table 2.1. Note that clutter is not included in this set of simulations. The effects of clutter and clutter filter performance are discussed in Chapter 3.

2.5.1 Plug Flow

We fist consider blood flow in a straight vessel with plug velocity profiles. We simulate two sets of cases. In the first set, the beam-to-flow angle θ is varied from 0^o to 90^o, in steps of 15^o, while the out-of-plane angle ϕ is kept at 0^o. In the second set, the out-of-plane angle ϕ is varied from 0^o to 90^o, in steps of 15^o, while the beam-to-flow-angle θ is kept at 90^o. Detailed bias and standard deviation measurements are presented in Tables 2.2 and 2.3. The standard deviations of lateral motion components v_x and v_y are generally larger than the standard deviation of axial motion component v_z. The higher standard deviation is to be expected since the spatial resolution in the lateral dimensions is 20× coarser than the axial dimension.

For the second set of simulations, where ϕ is varied, the estimated velocity magnitude has an average bias of about 0.02 m/s and an average standard deviation of about 0.08 m/s, while the estimated flow angle ϕ has an average bias of about 0.5^o and an average standard deviation of about 4.2^o. Compared to the results presented in [11], which simulate the same flow conditions, the performance of our method

Table 2.2: Estimation Performance for Different θ with Plug Flow.

	Avg. Bias(%)			Avg. Std.(%)		
θ	V_x	V_y	V_z	V_x	V_y	V_z
0^o	0.93	0.91	0.21	3.28	3.39	0.46
15^o	0.73	1.00	0.18	3.48	4.17	0.34
30^o	1.02	1.29	0.13	4.47	6.73	0.51
45^o	0.82	1.50	0.15	3.80	6.60	0.54
60^o	0.82	1.95	0.56	5.06	8.34	1.06
75^o	0.65	1.01	0.11	2.81	5.96	0.38
90^o	1.06	1.46	0.12	4.22	7.11	0.53

(with a 32×32 array) is better than the performance of the TO (transverse oscillation) method with a 32×32 array, and yet close to the performance of the TO method with a 64×64 array.

A pictorial summary of the results is shown in Fig. 2.5. Black arrows (solid) indicate the true velocities, while the red lines (dashed) are the estimated velocities, with ellipses (solid) showing the standard deviations for two velocity components (v_x and v_z for the first set of simulations; v_x and v_y for the second set of simulations). The estimated velocities match the true velocities well in both scenarios, demonstrating that the SAD+SLP-3Dopt produces accurate 3D velocity vector estimates.

2.5.2 Parabolic Flow

Next, we evaluate SAD+SLP-3Dopt with parabolic flow. Fig. 2.7 shows the estimated velocity components with $\theta = 90^o$ and $\phi = 0^o$. For this case, the estimations are fairly accurate with average standard deviations of the three velocity components

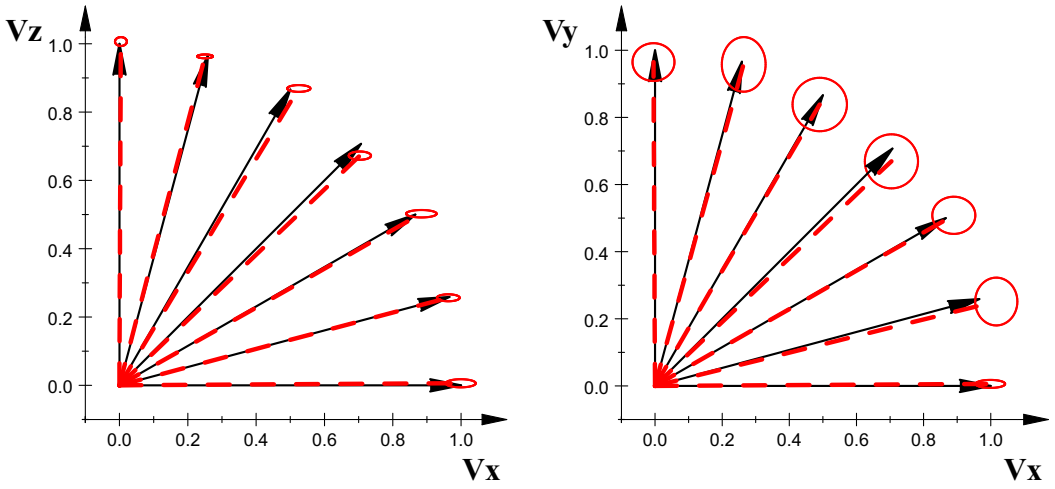
Table 2.3: Estimation Performance for Different ϕ with Plug Flow.

	Avg. Bias(%)			Avg. Std.(%)		
θ	V_x	V_y	V_z	V_x	V_y	V_z
0^o	1.06	1.46	0.12	4.22	7.11	0.53
15^o	0.99	1.02	0.08	4.63	6.30	0.32
30^o	0.96	1.96	0.14	7.06	6.58	0.50
45^o	2.09	1.98	0.25	9.29	7.98	1.06
60^o	1.91	1.48	0.15	9.29	8.09	0.69
75^o	1.92	0.96	0.12	6.76	4.89	0.43
90^o	1.59	1.50	0.13	7.21	5.05	0.53

Table 2.4: Estimation Performance Comparison with or without Phase Correction. No Clutter is Included. The Peak Velocity is 1 m/s

	Avg. Bias(%)			Avg. Std.(%)		
Case	V_x	V_y	V_z	V_x	V_y	V_z
w/o Ph. Corr.	-12.77	-0.26	0.03	3.99	7.11	0.36
Ph. Corr.	-6.88	-0.28	0.03	3.71	8.08	0.42

v_x, v_y and v_z being around 4.0%, 7.1% and 0.4% respectively. While the biases of v_y and v_z are quite small, v_x is underestimated with an average bias of about 13%. The underestimation is mainly due to the decorrelation between the kernels and the candidates because of the flow gradients within the kernel.



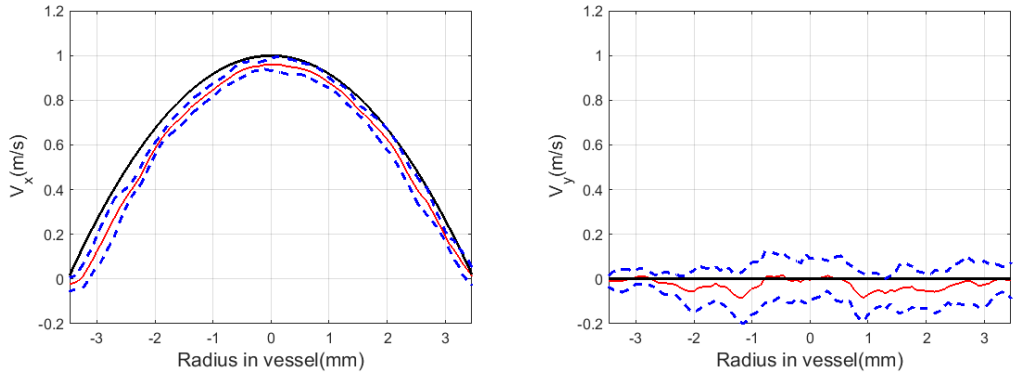
(a) Primary flow in x dimension

(b) Primary flow in y dimension

Figure 2.5: Velocity Estimates for Plug Flow as a Function of (a) Beam-to-flow-angle θ; (b) Out-of-plane-angle ϕ. Black Arrows (Solid) Indicate the True Velocities, while the Red Lines (Dashed) Represent the Estimated Velocities with Ellipses (Solid) Showing the Standard Deviations for the Two Velocity Components.

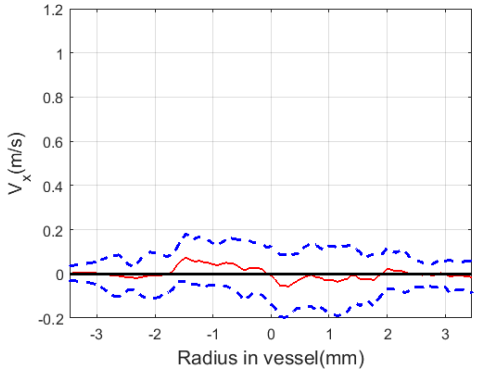
2.5.3 Phase Correction

To demonstrate the improvement due to use of phase correction in the blood velocity estimation, as mentioned in Section 2.4, we present the motion estimation results with phase correction and without phase correction. The results are shown in Fig. 2.7 and Table 2.4. We can see that phase correction reduced the bias from 13% [42] to below 7% in the primary flow direction. When θ is 60^o, the biases are smaller compared to the case when θ is 90^o, though the improvement due to phase correction is less significant. In general, the estimation performance is better with considerably smaller standard deviation when θ is 90^o than when θ is 60^o. We also see that when beam-to-flow angle is 90^o, the standard deviation of the y dimension is higher than in the x dimension. To verify that this is not an artificial systematic

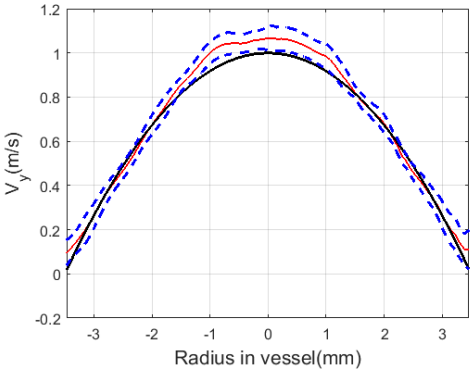


(a) Primary flow in x dimension

(b) Primary flow in y dimension



(c) Primary flow in x dimension



(d) Primary flow in y dimension

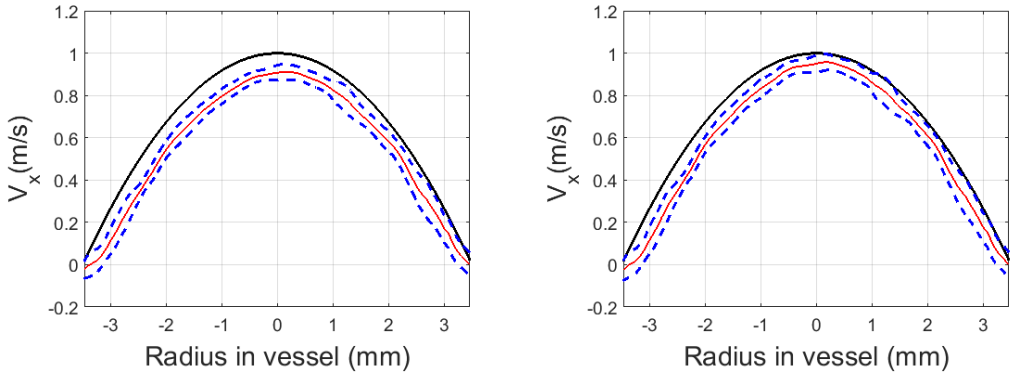
Figure 2.6: Parabolic Flow. (a)(c): Flow Direction is Lateral-x; (b)(d): Flow Direction is Lateral-y. Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean±1 Standard Deviation.

bias between the two lateral dimensions, we changed the primary flow direction to be lateral-y instead of lateral-x. The results are compared in Fig. 2.6. We can see that the standard deviation of x dimension is now higher than in the y dimension. We conclude that the motion estimation in the lateral dimensions tends to have higher standard deviation in the dimension where the actual flow is smaller (or even zero).

2.6 Summary

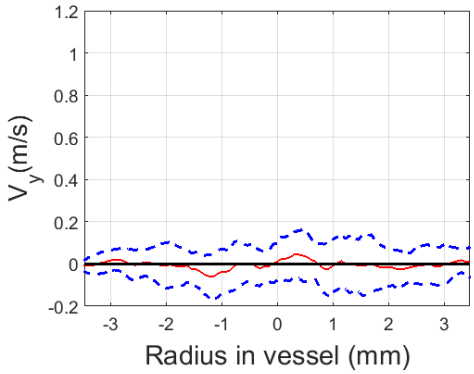
In this chapter, we proposed a two-tiered blood velocity estimator which uses SAD for pixel level estimation and SLP-3D for sub-pixel estimation. By replacing the cross correlations with SAD in the pixel level estimation and reducing the computations in FFT and IFFTs, we achieved about 9× computational complexity reduction without compromising on the estimation accuracy.

We evaluated the performance of the velocity estimator with both plug flow and parabolic flow. For plug flow, the estimation bias is less than 2.1% and the standard deviation is less than 9.3%. For parabolic flow, the estimation bias is less than 6.9% and the standard deviation is less than 8.1%.

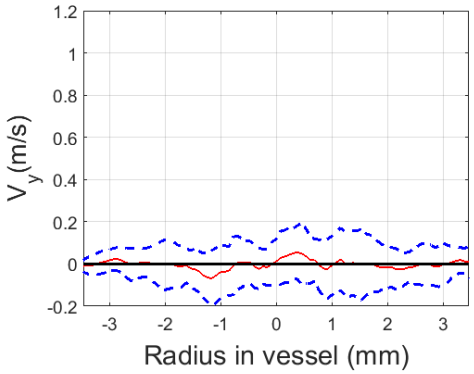


(a) Without phase correction: x dimension

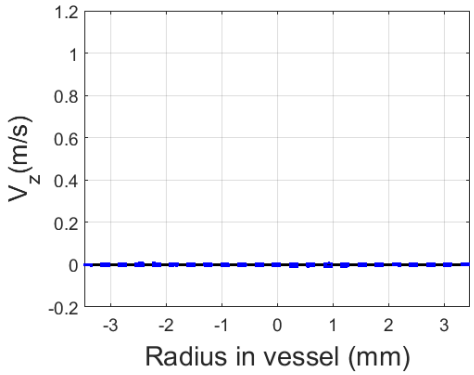
(b) Phase correction: x dimension



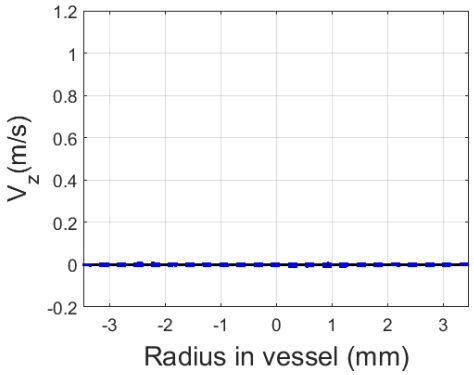
(c) Without phase correction: y dimension



(d) Phase correction: y dimension



(e) Without phase correction: z dimension



(f) Phase correction: z dimension

Figure 2.7: Effect of Phase Correction on Parabolic Flow. Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean±1 Standard Deviation.

Chapter 3

CLUTTER REMOVAL

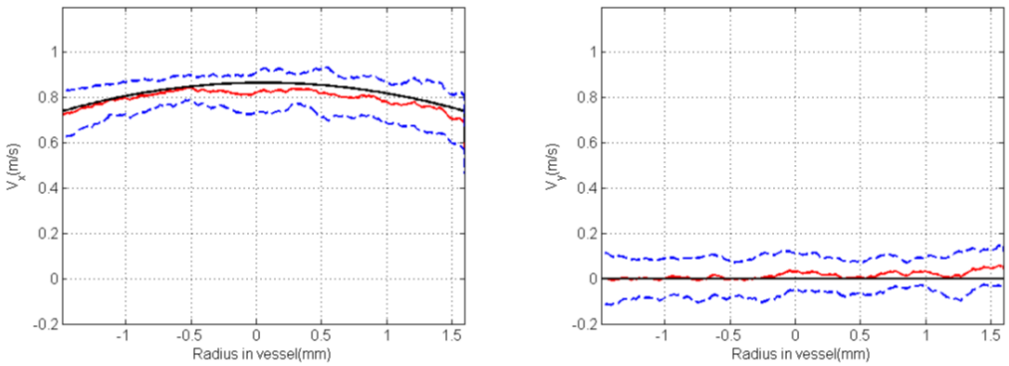
The low cost motion estimation method for blood flow estimation presented in Chapter 2 assumed that there was no clutter mixed in the blood signal. However, clutter is always present, and if not removed sufficiently, may cause a large bias in velocity estimation.

Consider motion estimation of a system where blood has slow moving clutter. Fig. 3.1 shows how presence of clutter degrades the accuracy of velocity estimation (3.1c and 3.1d). Clearly, the clutter signal has to be removed! Fig. 3.1e and 3.1f show the velocity estimation results when the clutter has been removed by an eigen-based clutter filter. These results are almost as good as the case where there is no clutter (Fig. 3.1a and 3.1b).

In this chapter, we describe clutter removal techniques that are able to sufficiently filter out both slow- and fast- moving clutters with reduced computational complexity. The clutter-free blood signal is then processed by motion estimation unit as shown in Fig 2.1.

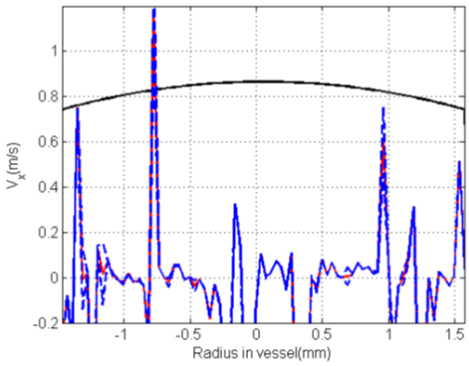
3.1 Clutter Characteristics

In blood flow estimation, the term clutter refers to the high-amplitude signals from the vessel wall and surrounding tissues. The amplitude of the clutter signal is usually 40 to 60 dB higher than the amplitude of blood signal [20]. However, since vessel wall and surrounding tissues are either stationary or moving very slowly compared to the blood, clutter has relatively low frequency. As shown in Fig. 3.2, clutter usually corresponds to the first (several) eigen/singular values, and has low frequency. These

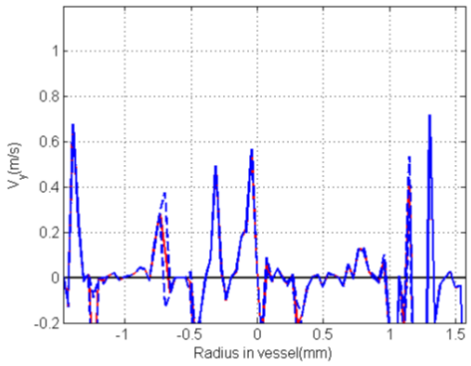


(a) No clutter: x dimension

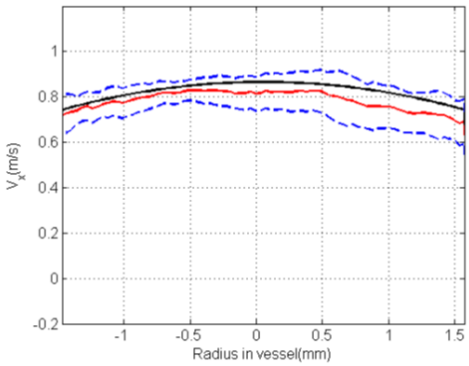
(b) No clutter: y dimension



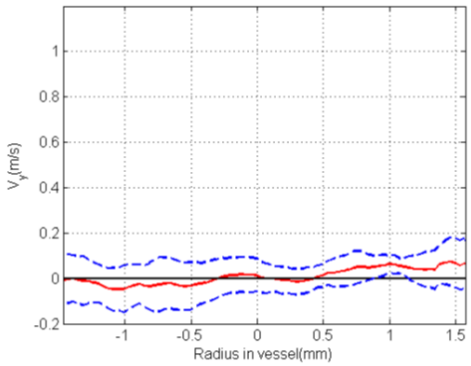
(c) With clutter: x dimension



(d) With clutter: y dimension

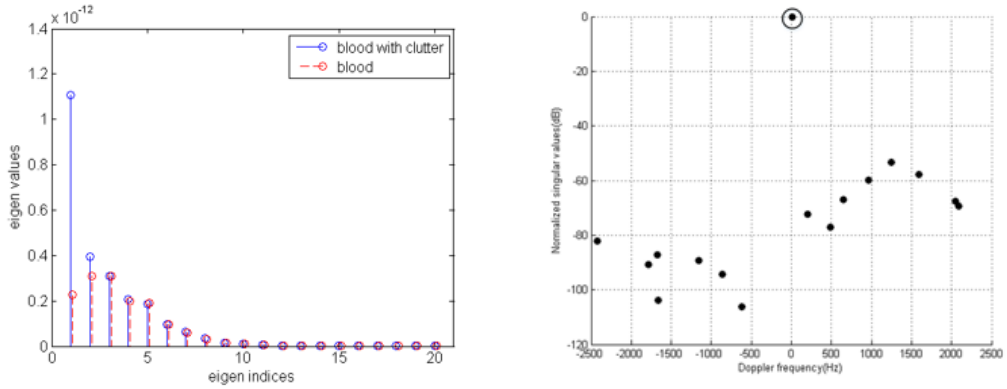


(e) Clutter removed: x dimension



(f) Clutter removed: y dimension

Figure 3.1: Flow Estimation with Beam-to-flow-angle $\theta = 60^o$. Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean$\pm$1 Standard Deviation.



(a) High amplitude

(b) Low frequency

Figure 3.2: Clutter Has High Amplitude and Low Frequency. (a) Singular Value Amplitudes for both Blood and Clutter signal. (b) Singular Value vs. Doppler Frequency. The Singular Value at Doppler frequency of ≈0 Hz (circled) Corresponds to Clutter.

two basic characteristics of clutter, namely high amplitude and low frequency, are key to identifying it and filtering it from the signal.

Traditionally, high pass finite-impulse response (FIR) or infinite-impulse response (IIR) fiters have been used to suppress the clutter with a fixed stopband. However, clutter characteristics vary for different blood vessels, and blood flow is pulsatile. As shown in Fig. 3.3, we can see that when blood velocity varies, the spectrum is different. Therefore, filters with fixed stopband result in poor filtering performance, which could hamper the blood velocity estimation [37, 38].

3.2 Eigen-based Clutter Filter

To improve the clutter filter performance, some studies have proposed adaptive filters, such as eigen-based methods. In [37], two different approaches, namely, the single-ensemble formulation and the multi-ensemble formulation, are presented. The

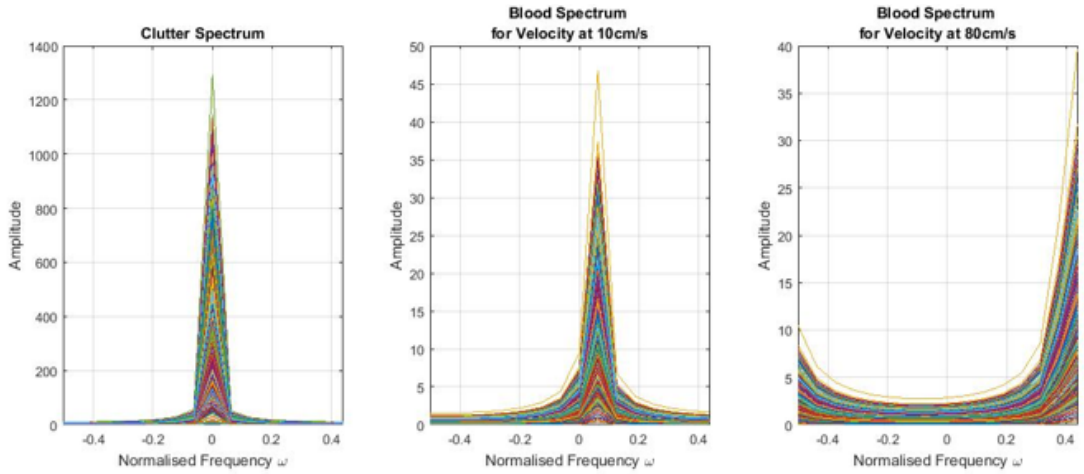


Figure 3.3: Clutter Spectrum and Blood Spectrum with Different Velocities.

single-ensemble formulation works on the slow time signal of a signal sample volume at a time, while the multi-ensemble formulation considers several neighboring sample volumes at the same time. The single-ensemble formulation has better performance when the clutter characteristics vary locally in a spatial domain. An important step in eigen-based clutter filters is to determine whether a certain subspace is due to clutter. In [39], only the Doppler frequency is used to distinguish whether a certain subspace is due to clutter or blood, whereas in [52], both Doppler frequency and relative eigenvalue amplitude are used to make this decision.

In this work, we consider both slow and fast moving clutter based on the clutter model in [39]. If the clutter velocity is 0.5% of the peak blood velocity, we refer to it as slow moving clutter, and if clutter velocity is 10% of the peak blood velocity, we refer to it as fast moving clutter. For slow moving clutter only the most significant subspace is due to clutter, while for fast moving clutter, usually the first two subspaces are due to clutter. In [39], the Hankel-SVD approach is shown to have good performance for various scenarios. However, since this approach involves singular value decomposition

(SVD), the computational demand is quite high. In this section, we consider several approaches to reduce the computational complexity.

The core of the Hankel-SVD based clutter filter method is briefly described as follows:

- Step 1: Form the data matrix A of size $m \times n$ for a given voxel in a Hankel structure [39]; the packet size is $m + n - 1$.

- Step 2: Calculate the SVD of matrix A: $A = USV$, where S is a diagonal matrix with singular values $\lambda_1, \lambda_2, ...$, U and V are matrices whose column vectors are the singular vectors.

- Step 3: Reconstruct the subspace matrix that corresponds to clutter, by $A_c = \sum_i \lambda_i U_i V_i'$, where U_i (of size $m \times 1$) and V_i (of size $n \times 1$) are the singular vectors corresponding to λ_i, and V_i' is the conjugate transpose of V_i.

- Step 4: Transform the subspace matrix to signal vectors based on the Hankel structure by taking the average of the elements on the inverse diagonal of the matrix. This process is shown in Fig. 3.4. Finally, subtract this signal vector that corresponds to clutter from the original signal packet, to reconstruct the blood signal.

3.3 Reduce Complexity of Eigen-based Clutter Filter

For full SVD (Golub-Reinsch SVD), the complexity of Step 2 in terms of floating point operations (flops) is $4m^2n + 8mn^2 + 9n^3$, where m and n are the dimensions of the matrix. Since the input to the clutter filter is the demodulated signal, which is complex, the complexity increases by $4\times$. Step 3 takes $(6mn+6m)\times m$ flops, and Step 4 takes $2(mn - m - n - 1)\times m$, which makes the total complexity $24m^2n + 32mn^2 +$

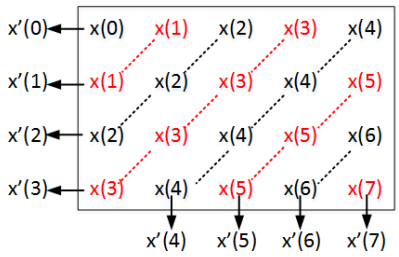


Figure 3.4: Signal Reconstruction from a Hankel Matrix in Step 4. $x'(i)$ is the Average of $x(i)$s.

$36n^3 + 4mn - 2m^2 + 2m$ flops. In our flow estimation system, the packet size is 32, and so the size of matrix A ($m \times n$) is 16×17. The total number of flops per SVD calculation is 429,892, which translates to 2.25×10^{11} flops for processing a volume that has 512×32×32 voxels. This is clearly excessive and so we derive methods that exploit blood-clutter characteristics to reduce complexity.

In clutter filtering, usually only a small number of subspaces represent clutter, the computational complexity can be reduced if we only find the singular values that are large. Power iteration and subspace iteration methods [40] are effective ways to find large singular values. Although these methods are iterative, since clutter subspaces usually have much higher power than the blood subspaces, the convergence of these algorithms is quite fast. Another way of reducing the complexity is by reducing the size of matrix size for SVD. This leads to small degradation in the filter performance especially at the edge of the vessel, as will be shown in Section 3.4.

In the Hankel-SVD method, we need to first find the SVD for matrix A. To use power iteration or subspace iteration, we convert the SVD into eigenvalue decomposition (EVD) first. If the SVD of A gives $A = USV'$, then the EVD of $A*A'$ gives $A*A' = US_0U'$, where $S_0 = S*S'$. Therefore, a matrix multiplication is needed

before using power iteration or subspace iteration to replace SVD. This step can be computationally costly if the matrix size is large.

While power iteration is used for finding the most significant subspace in [45], it can also be used to find the second most significant subspace by removing the most significant subspace from the correlation matrix $A*A'$ and applying power iteration again. The signal power of the residual signal is calculated and compared with a threshold to determine whether it is necessary ro calculate the second subspace. The threshold is dynamic and is dependent on the total signal power and clutter-to-blood ratio expectation (usually 40-60 dB, varies with application). The extended versions of both power iteration and subspace iteration methods can be used to find more than the two largest singular values. The power iteration and subspace iteration based methods are described in Algorithms 2 and 3, respectively.

In our scheme, the packet size is 32, resulting in a Hankel matrix A of size 16×17. In both the full SVD and its substitutes, the computational complexity is $O(n^3)$, where the matrix is of size $n \times (n+1)$. Therefore, reducing the matrix size for SVD is another effective way to reduce the computational complexity. To this end, we propose methods which split a packet of 32 samples into 4 subgroups with 8 samples per subgroup.

We evaluate the performance of the five methods which have varying degrees of complexity. Methods 1-4 are based on Hankel-SVD clutter filter and Method 5 is FIR filter (baseline). The FIR filter is a minimum phase filter, which has a frequency response that is similar to the poly regression filter in [53].

- Method 1(SVD32): Hankel-SVD clutter filter, with SVD fully implemented, for packet size of 32

- Method 2(SVD8): Hankel-SVD clutter filter, with SVD fully implemented, for 4 subgroups with 8 samples in each subgroup
- Method 3(PIter8): Hankel-SVD clutter filter, with power iteration (twice), for 4 subgroups with 8 samples in each subgroup
- Method 4(SIter8): Hankel-SVD clutter filter, with subspace iteration, for 4 subgroups with 8 samples in each subgroup
- Method 5(FIR): FIR filter, order 16 (33 taps), with cutoff frequency of 0.15 Nyquist rate.

Algorithm 2: Power iteration method for finding the largest singular value

Data: Matrix A (m×n)

Result: The largest singular value S(1) and corresponding singular vectors u and v (m×1, n×1) for matrix A

Initialization: v_0 is arbitrarily given (randomly generated or unit vectors);

$R = A \times A'$;

while $abs(\lambda_{k+1} - \lambda_k) > threshold$ **do**

$\lambda_{k+1} = norm2(R \times v_k)$;

$v_{k+1} = R \times v_k / \lambda_{k+1}$

end

$S(1) = sqrt(\lambda)$;

$u = A \times v / S(1)$

Table 3.1 compares the complexities of the five methods in terms of number of flops. Method 5 has only 2048 flops but its performance is nowhere as good as Methods 1-4. Note that Method 1 operates on a matrix of size 16×17 and Methods

Algorithm 3: Subspace iteration method for finding the two largest singular values

Data: Matrix A (m×n)

Result: The two largest singular value S(1), S(2) and corresponding singular vectors u and v (m×2, n×2) for matrix A

Initialization: v_0 is arbitrarily given so that $v_0' \times v_0 = I$;

$R = A \times A'$;

while $abs(\lambda_{k+1} - \lambda_k) > threshold$ **do**

$z_{k+1} = R \times v_k$;

$z_{k+1} = v_{k+1}/\lambda_{k+1}$

end

$S(1) = sqrt(\lambda(1,1)), S(2) = sqrt(\lambda(2,2))$;

$u(:,1) = A \times v/S(1), u(:,2) = A \times v/S(2)$

Table 3.1: Computational Complexity (in Flops) of the 5 Methods, for a Single Voxel with Packet Size of 32.

	Method 1	Method 2	Method 3	Method 4	Method 5
No. of flops	429,892	38,704	4,160	3,984	2,048

2-4 divide the packet into four subgroups and work on matrices of size 4×5. As a result, Methods 2-4 are much less computationally complex than Method 1.

Methods 3 and 4 use power iteration and subspace iteration, respectively, and have significantly lower number of flops than Method 2. For Method 3, assuming the number of iterations for calculating the first and second subspace are N_1 and N_2 respectively, the iterative process takes $(63(N_1+N_2)-112)$ additions, $(72(N_1+N_2)-128)$ multiplications, $(N_1 + N_2)$ square root operations, and $7(N_1 + N_2)$ divisions. There

is another 56 additions and 80 multiplications to remove the first subspace from the correlation matrix R. As the first subspace is usually dominant, N_1 is usually 3 and N_2 is usually 4 or 6. So we choose $N_1 = 3$ and $N_2 = 5$ for our estimates. For Method 4, the average number of iterations is $N = 3$. The iteration process takes $112(N-1) + 58N$ additions, $128(N-1) + 64N$ multiplications, $2N$ square root operations, and $14N$ divisions. In addition, for Methods 3 and 4, 160 multiplications and 208 additions are needed for calculating the correlating matrix R, and 80 multiplications and 70 additions are needed for removing each subspace. Following the normalized computation metric, Method 3 takes 2065 (or 5326) normalized computations to remove one (or two) largest subspace and Method 4 takes 4597 normalized computations. We can see that when two subspaces need to be removed, Method 4 is less computationally complex than Method 3. However, Method 3 needs about 48% fewer normalized computations if only one subspace is to be removed. Therefore, Method 3 and 4 can be used for different scenarios–when the region of interest contains mainly fast-moving clutter, Method 4 is less complex; when it contains mainly slow-moving clutter, Method 3 is less complex.

3.4 Post-filter Clutter-to-blood Ratio (CBR)

To analyze the performance of each clutter filter technique, we measure the clutter-to-blood ratio (CBR) after filtering [39]. CBR is calculated as the ratio of the post-filter power of the clutter-only signal and the power of the slow-time clutter+blood signal. If the filter works properly, this ratio should be negative on a dB scale. If the same level of white noise is added to both the clutter-only and normal slow-time signals, the CBR can be expected to be the negation of the blood signal-to-noise ratio.

Table 3.2: CBR for Different Clutter Removal Methods (Slow Moving Clutter).

	CBR(dB)				
Method	*Min*	25%	*Median*	75%	*Max*
1:SVD32	-27.58	-23.32	-22.04	-20.81	-16.26
2:SVD8	-27.99	-22.96	-21.55	-20.12	-15.64
3:PIter8	-27.99	-22.95	-21.54	-20.11	-15.64
4:SIter8	-27.99	-22.95	-21.55	-20.12	-15.33
5:FIR	-26.93	-20.33	-18.22	-15.86	-7.14

Table 3.3: CBR for Different Clutter Removal Methods (Fast Moving Clutter).

	CBR(dB)				
Method	*Min*	25%	*Median*	75%	*Max*
1:SVD32	-29.48	-24.16	-22.87	-21.65	-16.76
2:SVD8	-32.52	-25.98	-24.35	-22.35	-13.22
3:PIter8	-33.16	-26.23	-24.42	-22.49	-13.22
4:SIter8	-33.16	-26.56	-24.86	-22.93	-13.21
5:FIR	-24.20	-5.23	-2.97	-1.61	-0.09

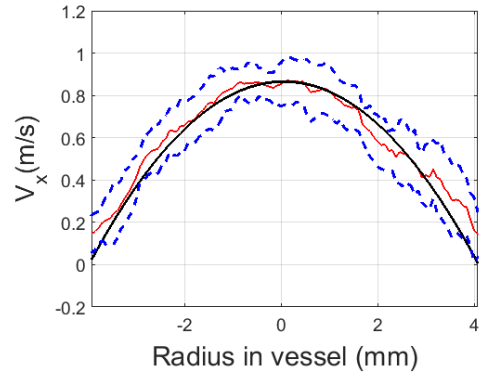
To account for simulation variation across packets, we randomly select roughly 40,000 packets and calculate CBR for each packet. The blood signal-to-noise ratio (BSNR) is 20 dB for both the slow and fast moving clutter cases, and hence the median CBR is expected to be around -20 dB. The statistics of the results for slow moving and fast moving clutter are presented in Tables 3.2 and 3.3, respectively. Methods 1 through 4 have comparable performance in terms of median CBR while Method 5, which is based on the FIR filter, has poor performance. Method 1 results in

the smallest range of CBR, which results in smaller deviations in velocity estimations as shown in Section 4.2.

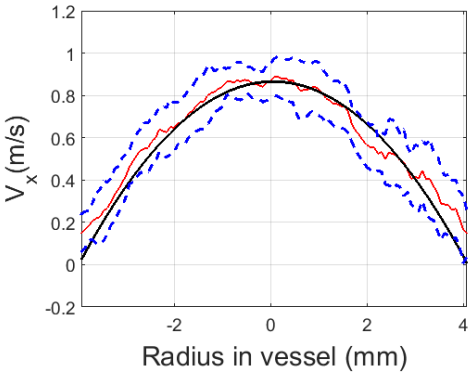
3.5 Summary

In this chapter, we presented different types of clutter filters and compared them with respect to post-filter CBR results. The proposed methods based on power iteration and subspace iteration have similar performance with the full eigen-based method. Our simulation results agree with the conclusion in [39] that the eigen-based clutter filter outperforms traditional high pass FIR or IIR filters.

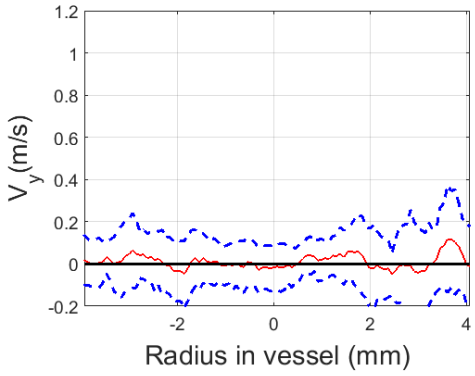
To examine whether the blood signal gets attenuated by a clutter filter, we consider a case where the blood-only signal is filtered followed by velocity estimation. We conduct an experiment similar to that in [53], with a beam-to-flow angle of 60^o. The results in Fig. 3.5 show that the clutter filter detects no clutter in the blood-only signal and that the blood signal is not attenuated. In fact, the estimation result is almost as good as the case where the clutter-free blood signal is directly sent to the velocity estimation unit. Thus, the proposed eigen-based clutter filter introduces no attenuation to the blood signal when clutter does not exist.



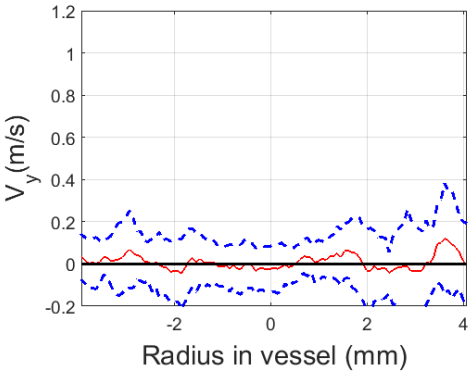
(a) Unfiltered: x dimension



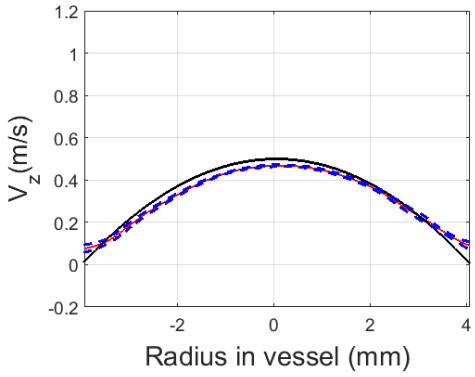
(b) Filtered: x dimension



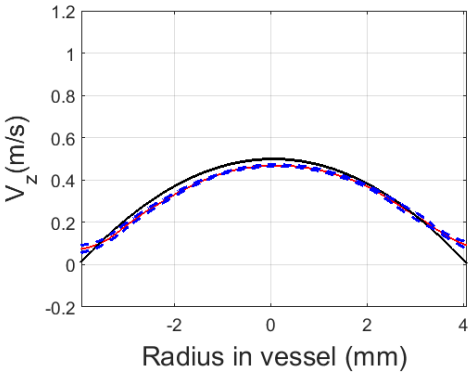
(c) Unfiltered: y dimension



(d) Filtered: y dimension



(e) Unfiltered: z dimension



(f) Filtered: z dimension

Figure 3.5: Flow Estimation (Slow Moving Clutter) for Blood Only Signal: Unfiltered (Left) and Filtered (Right). Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean±1 Standard Deviation.

Chapter 4

FLOW RATE ESTIMATION FOR PARABOLIC BLOOD FLOW

In Chapter 2 and Chapter 3, we described our low-cost methods for motion estimation and clutter removal. We presented the simulation results for both modules separately and showed that our methods achieve excellent performance with significantly computational complexity reduction. In clinical practice, the estimated velocity field is used to calculate the flow rate. Abnormal flow rate often indicates vascular disease [54, 55]. As mentioned in Chapter 1, 3D velocity field enables flow rate estimation on any arbitrary plane. In this chapter, we first describe the flow rate calculation method, and then present the performance of our blood flow estimation for the case then the flow is parabolic. We also present computational complexity of the whole system.

4.1 Flow Rate Calculation

To obtain accurate flow rate estimates, accurate velocity estimates are needed throughout the vessel. Velocity estimates at the edge of the vessel are less accurate than those in the center due to poor clutter filtering performance. This is because the blood velocity is usually low at the edge, making it harder to distinguish its spectrum from that of clutter. Furthermore, a sample volume at the vessel edge may intersect the vessel wall, leading to estimation error. Therefore, we must treat velocity estimation at the edge of the vessel carefully.

Flow rate estimation is usually calculated by integrating the velocity vectors over a certain cross section of the vessel. Considering the use of plane wave imaging and the fact that the beam-to-flow angle is more likely to range from 45^o to 90^o, it is reasonable to use the cross sections in which the scanlines lie (y-z plane). However,

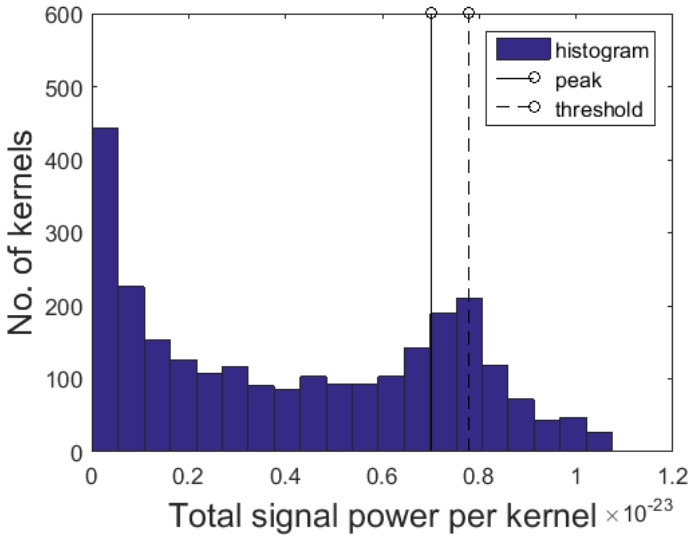


Figure 4.1: Power Weighting Method: Finding Threshold Based on the Kernel Power Histogram

since sample volumes abutting the vessel walks include components both inside and outside the blood vessel, it is not accurate to directly integrate the velocity estimates corresponding to them.

Here we propose to use power weighting in flow rate calculation. In [10], the Doppler power histogram was used to find the threshold P_T for partial weighting: the pixels that have Doppler power larger than the threshold were given a weight of 1, while the pixels that have Doppler power smaller than the threshold were given a weight ranging from 0 to 1, depending on the ratio between their Doppler power and the threshold. Instead of Doppler power, here we calculate the kernel power defined as the sum of signal power of voxels within the kernel. The threshold for partial weighting is given as $P_T = 0.9 \times P_M$, where P_M is the peak of the kernel power histogram. Note that since the signal power is calculated after clutter filtering, the kernels that are fully associated with blood signal have larger power than the kernels that have partial association. Thus, the peak P_M should not be located in the region

where the kernel power is small. We illustrate this method in Fig. 4.1, where we treat kernels with power larger than the threshold as containing blood signal only and give them a weight of 1. The power weighting method corrects the overestimation of the flow rate due to the fact that the area ascribed to the velocity vector is not all blood near the boundary. The results obtained by both direct integration and power weighting are shown in Section 4.2.2.

4.2 System-level Clutter Filter Performance

To construct a comprehensive demonstration of the functionality of our methods, we consider both slow moving clutter and fast moving clutter for beam-to-flow angles of 60^o and 90^o. The slow moving clutter case represents clutter movement caused by respiratory motion, while the fast moving clutter case represents clutter movement caused by pulsatile motion. The fast moving clutter case has lower blood velocity, lower pulse repetition frequency, and higher clutter velocity and vibration frequency, resulting in a more difficult scenario for the clutter filter. We only consider the primary flow in our simulations.

A single angle plane wave with 128×128 aperture size and a pitch of 0.5λ is used in transmit to ensure sufficient transmit power. A 32×32 aperture size with a doubled pitch (λ) is used in receive, and the separable delay-and-sum is used for beamforming [56]. The beamformed RF data is used in the flow estimation. The corresponding simulation settings are listed in Table 4.1.

We use Field II to generate the blood signal [44, 51]. The clutter signal, based on the clutter model in [37, 39], is added to the blood signal to form the clutter+blood signal. The clutter+blood signal is then processed by the clutter filters, followed by motion and flow rate estimation. We simulate a cylindrical phantom with a length

Table 4.1: Simulation Environment

Parameter	**Slow moving clutter**	**Fast moving clutter**
Speed of sound	1540 m/s	1540 m/s
Transmit aperture pitch	0.5λ	0.5λ
Transmit aperture size (2D)	128×128	128×128
Receive aperture pitch	λ	λ
Receive aperture size (2D)	32×32	32×32
Center frequency (f_0)	4 MHz	4 MHz
6 dB Bandwidth	2 MHz	2 MHz
Sampling frequency (f_s)	40 MHz	40 MHz
Vessel radius (R)	3.5 mm	3.5 mm
Vessel center depth	15 mm	15 mm
Packet size	32	32
Beam-to-flow angle	$60^o/90^o$	$60^o/90^o$
Blood peak velocity (v_0)	1 m/s	0.3 m/s
Clutter velocity (v_c)	5 mm/s	30 mm/s
Pulse repetition frequency (f_{prf})	5 kHz	2 kHz
Clutter-to-blood ratio	40 dB	40 dB
Vibration frequency (f_{vib})	1 Hz	1.5 Hz

of 10 mm and a radius of 3.5 mm containing blood scatterers. The density of the scatterers is 10 per mm^3, to ensure Gaussian distributed speckle signal amplitudes.

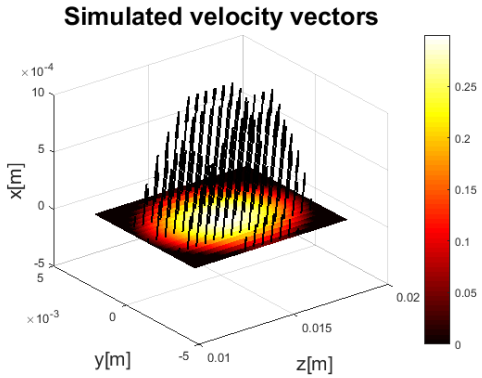
We consider several clutter filters that are described in Section 3.4. We evaluate the performance of our system based on the following three metrics. We first present the blood velocity estimation results, and compare the effect of different clutter filters on the estimation accuracy in Section 4.2.1. We then consider the flow rate estimation on a cross section of the vessel in Section 4.2.2.

4.2.1 Blood Velocity Estimation with Clutter

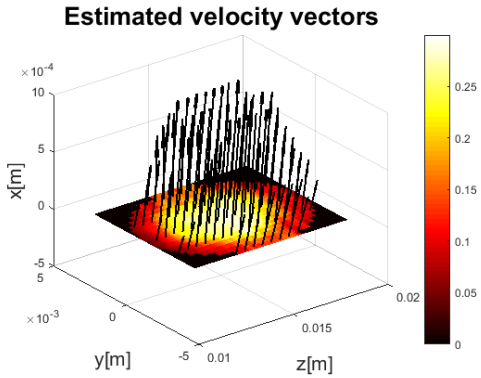
To evaluate system-level clutter filter performance, we compare the corresponding velocity estimation results. In Fig. 4.2, the 3D velocity estimates and the simulated velocity vectors (ground truth) on a cross section plane are presented. The estimated velocity vectors are close to the simulated velocity vectors in the 2D parabolic profile with small differences in velocity direction and amplitude.

The estimation results with beam-to-flow angle of 60^o and 90^o for Methods 1-4 are shown in Tables 4.2, 4.3, 4.4 and 4.5 respectively. Different clutter filters are applied to both the slow and fast moving clutter cases, and the performance is compared. Results for FIR filter based clutter filter are very poor and so have not been listed.

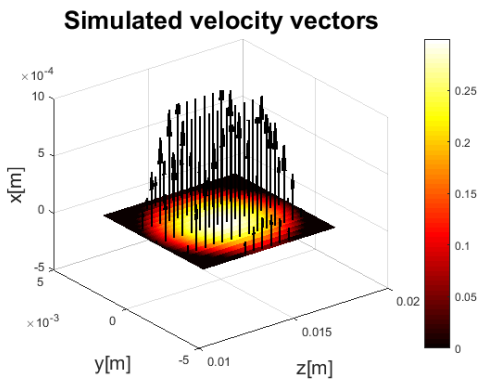
Our results show that the eigen-based clutter filters clearly outperform the FIR filter. Among the eigen-based clutter filters, the filter based on full SVD (Method 1) has the best performance. In fact, it has performance that is closest to the clutter-free case, which indicates best possible reconstruction of the blood signal. Methods 2-4 have similar performance, with comparable standard deviations to Method 1. In terms of average bias, Method 1 is closer to that of the clutter-free case compared to Methods 2-4. The latter methods lose some accuracy in estimation of the subspaces due to the reduced number of samples. However, in some cases (slow moving clutter



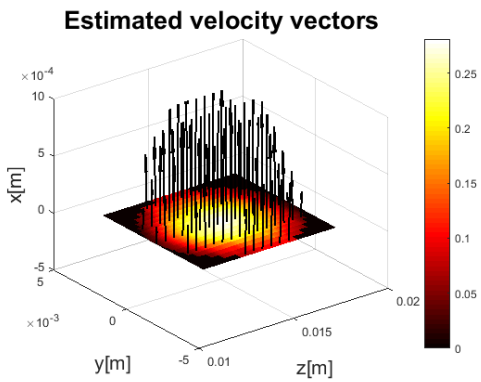
(a) Sim. velocity vectors (60^o)



(b) Est. velocity vectors (60^o)



(c) Sim. velocity vectors (90^o)



(d) Est. velocity vectors (90^o)

Figure 4.2: Flow Estimation Results for Fast Moving Clutter Case, with a Beam-to-flow-angle of 60^o in (a),(b) and 90^o in (c),(d). The Vectors are Obtained from a Cross Section Perpendicular to the Lateral-x Direction.

Table 4.2: Velocity Estimation Performance when the Beam-to-flow-angle is 60^o. The Peak Velocity is 1 m/s and Slow Moving Clutter is Included.

Clutter	**Clutter Filter**	Avg. Bias(%)			Avg. Std.(%)		
		V_x	V_y	V_z	V_x	V_y	V_z
w.o. clutter	None	2.22	1.13	1.56	11.45	13.73	0.80
w. clutter	1:SVD32	1.41	1.60	1.12	10.84	13.83	0.79
	2:SVD8	-4.75	0.77	-1.02	11.18	12.26	0.65
	3:PIter8	-4.82	0.72	-1.01	11.12	12.25	0.65
	4:SIter8	-4.83	0.76	-1.01	11.26	12.18	0.65
	5:FIR	19.70	2.70	8.63	10.12	9.08	3.62

when $\theta = 60^o$ and fast moving clutter when $\theta = 90^o$) their biases turn out to be smaller than Method 1. For all eigen-based clutter filters, as we expect, the standard deviation of the fast moving clutter case is higher than the slow moving clutter case. When the peak blood velocity and clutter velocity differ less, they are harder to distinguish.

Similar to related work [22, 23], our approach tends to slightly underestimate the velocity in the primary flow direction. In [23], 2D velocity vectors are estimated using a phase-based block matching approach with plane wave transverse oscillations. The simulation results showed that, as the beam-to-flow angle decreases from 90^o to 60^o, the estimation performance degrades but the bias and standard deviation stay within 15%. In [9],a 2D speckle tracking method was used for velocity estimation. In general, our estimation results in lateral-x and axial dimensions are comparable with the results presented in [9], despite differences in system settings. Compared to the TO method in [22], for beam-to-flow angle of 90^o, our method has smaller average

Table 4.3: Velocity Estimation Performance when the Beam-to-flow-angle is 60^o. The Peak Velocity is 0.3 m/s and Fast Moving Clutter is Included.

Clutter	**Clutter Filter**	Avg. Bias(%)			Avg. Std.(%)		
		V_x	V_y	V_z	V_x	V_y	V_z
w.o. clutter	None	2.10	2.20	1.81	14.21	15.21	0.95
w. clutter	1:SVD32	2.63	0.59	1.15	13.97	15.39	0.90
	2:SVD8	-1.11	-0.37	0.17	14.02	15.68	0.77
	3:PIter8	-4.82	-0.35	0.17	14.06	15.67	0.77
	4:SIter8	-4.83	-0.35	0.17	14.08	15.56	0.77

Table 4.4: Velocity Estimation Performance when the Beam-to-flow-angle is 90^o. The Peak Velocity is 1 m/s and Slow Moving Clutter is Included.

Clutter	**Clutter Filter**	Avg. Bias(%)			Avg. Std.(%)		
		V_x	V_y	V_z	V_x	V_y	V_z
w.o. clutter	None	-6.88	-0.28	0.03	3.71	8.08	0.42
w. clutter	1:SVD32	-4.42	0.78	-0.03	6.39	7.66	0.43
	2:SVD8	-0.08	0.46	0.40	4.60	9.52	0.38
	3:PIter8	-1.01	0.65	-0.02	3.81	7.45	0.40
	4:SIter8	-0.88	0.89	-0.02	5.23	7.33	0.40

Table 4.5: Velocity Estimation Performance when the Beam-to-flow-angle is 90^o. The Peak Velocity is 0.3 m/s and Fast Moving Clutter is Included.

Clutter	**Clutter Filter**	Avg. Bias(%)			Avg. Std.(%)		
		V_x	V_y	V_z	V_x	V_y	V_z
w.o. clutter	None	-1.99	0.45	0.01	4.36	11.82	0.42
w. clutter	1:SVD32	-6.95	0.51	0.20	6.38	11.77	0.41
	2:SVD8	-7.43	1.06	-0.78	6.47	11.90	0.41
	3:PIter8	-7.43	1.07	-0.77	6.47	11.90	0.41
	4:SIter8	-7.48	0.48	-0.77	6.99	10.82	0.41

deviation, but larger average bias. However, for beam-to-flow angle of 60^o, the two methods have comparable bias and our method has lower standard deviation. Note that the prior works [22, 23] only considered stationary clutter and removed it with simple FIR filters. In contrast, we considered both slow and fast moving clutter and still achieved velocity estimation performance that is comparable to the existing flow estimation methods.

4.2.2 Flow Rate Estimation with Clutter

We compare flow rate estimation results, including those obtained by direct integration and our power weighting technique. The estimation results for beam-to-flow angle of 60^o and 90^o are shown in Table 4.6 and 4.7.

We see that the flow rate estimation accuracy of our method was improved by using the power weighting method. The estimation is within 10% for both bias and standard deviation. The velocity estimation near the vessel is less accurate as the kernels are partially inside the vessel and also partially outside the vessel. The

clutter filter performance is expected to slightly degrade near the vessel wall as well. By giving fractional weight to the kernels near the vessel wall, we were able to further reduce the influence of the inaccurate estimates.

Clutter filter performance is also evaluated through flow rate estimation. When beam-to-flow angle is 60^o, the eigen-based clutter filters (Methods 1-4) have much better estimation accuracy than the FIR filter. For both slow and fast moving clutter, after power weighting, Method 1 overestimates the flow rate while Methods 2-4 underestimates the flow rate, with bias less than $\pm 7\%$ and standard deviation less than 6%. Similar conclusions can be made when the beam-to-flow angle is 90^o, as shown in Table 4.7. While Method 1 has larger bias for slow moving clutter, Methods 2-4 have comparable performances for both slow and fast moving clutter, as expected. Methods 2-4 have about -8% bias and 6% standard deviation for the slow moving clutter, and about -9% bias and 3% standard deviation for the fast moving clutter. These results are consistent with the velocity estimation results in Table 4.4 and 4.5. Overall, eigen-based clutter filters have good performance for both velocity estimation and flow rate estimation with Methods 3-4 having significantly reduced computational complexity.

4.3 Computational Complexity Analysis

In our scheme, the image volume has $512 \times 32 \times 32$ voxels, and the packet size is 32. To derive the complexity of the whole system, we see that separable beamforming for a plane wave system [16] has to be done 32 times, while the motion estimation (SAD+SLP3D) has to be done 31 times for the 31 volume pairs, and clutter filtering has to be done only once. As a result, separable beamforming, clutter filter (Method 3) and motion estimation constitute 6194M (45%), 2792M (18%) and 5647M (37%) of the total complexity, as shown in Fig. 4.3. If Method 4 is used for clutter filtering, the

Table 4.6: Flow Rate Estimation (Beam-to-flow-angle is 60^o). The Real Flow Rate is 1154.04 mL/min for Slow Moving Clutter, and 346.34 mL/min for Fast Moving Clutter.

(Unit: mL/min)	Slow moving clutter		Fast moving clutter	
	Direct integ.	Power weighting	Direct integ.	Power weighting
Method	**Avg.±Std.**	**Avg.±Std.**	**Avg.±Std.**	**Avg.±Std.**
1:SVD32	1304.7±58.83	1209.3±56.05	391.26±14.42	363.16±17.58
2:SVD8	1227.1±49.81	1083.8±69.53	340.93±12.83	325.06±13.92
3:PIter8	1225.0±50.18	1089.6±68.58	339.70±13.52	324.82±13.84
4:SIter8	1231.0±51.23	1094.4±64.15	340.07±13.66	323.94±13.57
5:FIR	731.09±44.22	730.78±44.13	N.A.	N.A.

Table 4.7: Flow Rate Estimation (Beam-to-flow-angle is 90^o). The Real Flow Rate is 1154.04 mL/min for Slow Moving Clutter, and 346.34 mL/min for Fast Moving Clutter.

(Unit: mL/min)	Slow moving clutter		Fast moving clutter	
	Direct integ.	Power weighting	Direct integ.	Power weighting
Method	**Avg.±Std.**	**Avg.±Std.**	**Avg.±Std.**	**Avg.±Std.**
1:SVD32	1248.2±20.49	1039.5±58.59	338.56±5.66	327.01±10.01
2:SVD8	1342.3±17.62	1062.4±66.83	325.96±6.94	314.23±10.27
3:PIter8	1343.0±16.69	1059.6±64.55	325.95±6.87	315.82±10.83
4:SIter8	1325.0±17.48	1064.0±71.99	325.05±7.02	315.77±10.06

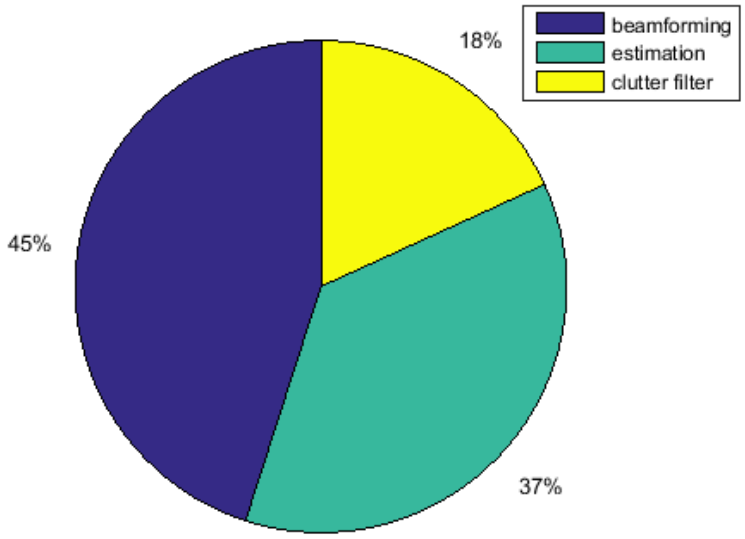


Figure 4.3: Computational Complexity Analysis for the Whole System when Power Iteration (Method 3) is used for the Clutter Filter.

change in the overall complexity is minor. Separable beamforming, clutter filter and motion estimation now contribute 6194M (46%), 2410M (16%) and 5647M (38%) to the total complexity respectively. The complexity of our flow rate estimation system is only increased by 2.22 times if Method 3 is used and 2.17 times if Method 4 is used, compared to the baseline plane wave beamforming system.

Considering both the performance and the computational complexity, we recommend using Method 3 for slow moving clutter and Method 4 for fast moving clutter. Compared to Method 4, Method 3 has about 20% more normalized computations for fast moving clutter, and about 55% less normalized computations for slow moving clutter. If the clutter characteristic is unknown, Method 3 is recommended since it can remove 1 or 2 subspaces as needed. If more subspaces have to be removed, Method 3 can be employed with minor changes.

4.4 Summary

In this chapter, we presented a weighting method based on kernel power that helped reduce the error in flow rate estimation. We showed that our flow estimation system which include a power/subspace iteration based clutter filter, a two-tiered velocity estimator using SAD for pixel level estimation and SLP-3D for sub-pixel level estimation, and a weighting method based on kernel power, achieved excellent estimation performance for both velocity vectors and flow rate. For instance, for parabolic flow with 90^o beam-to-flow angle surrounded by fast moving clutter, the velocity estimation has a bias less than 7.5% and standard deviation less than 12%, and the flow rate estimation has a bias less than 9% and standard deviation of less than 3%.

We also evaluated the clutter filter performance by comparing the velocity and flow rate estimation results with different clutter filters. The proposed power/subspace iteration based methods have comparable results with the naive implementation of SVD based methods, and are superior to the trational FIR filter. Taking the computational complexity into consideration, we concluded that power iteration based method is best for slow moving clutter while the subspace iteration based method is best for fast moving clutter.

Chapter 5

FLOW ESTIMATION IN CHALLENGING MODELS

In previous chapters, we described our two-tiered speckle tracking method that combines pixel level estimates using sum-of-absolute difference with sub-pixel level estimates using 3-D synthetic lateral phase [43]. We showed that our method produces accurate velocity vector estimates for a parabolic blood flow model with fast and slow moving clutter.

In this chapter, we extend our flow estimation method to handle challenging flow scenarios such as flow in a spinning disk (Section 5.1.2), flow in carotid bifurcation (Section 5.1.3) and flow with stenosis blockage (Section 5.2). The spinning disk and carotid bifurcation models are based on the synthetic aperture vector flow imaging (SA-VFI) challenge held in International Ultrasonic Symposium (IUS) 2018 [57]. The flow with stenosis blockage model is based on a computational fluid dynamics (CFD) simulation using COMSOL.

5.1 SA-VFI challenge

The SA-VFI challenge in 2018 [57] provided a platform to compare the performance of different estimators. It has two stages: training and testing. In the training stage the true velocity profile is known to the participants, while in the testing stage the true velocity profile is not known. Five data sets are given to the participants in the training stage. These include both simulation and experiment data for straight tubes with beam-to-flow angle of 60^o and 90^o and a spinning disc data set. The testing stage include the carotid bifurcation model. Since the SA-VFI challenge is for 2D

velocity estimation based on synthetic aperture, we had to modify our 3D velocity estimation method for plane wave imaging.

5.1.1 2D Velocity Estimation Method

The 2D velocity estimator in this chapter is based on speckle tracking and derived from our earlier work on 3D blood velocity estimation [42, 43]. In the proposed two-tiered method for 2D flow estimation, pixel-level motion is estimated with sum-of-absolute differences (SAD) and sub-pixel–level motion is estimated using the phase of the correlation function around the pixel-level peak. For the lateral dimension, we create a synthetic phase signal by operating on the spatial spectrum. Specifically, the spectrum is separated into two halves to generate up and down spectra. These two spectra are then inversely transformed into up and down images and cross correlation functions are calculated for both images. The lateral phase is obtained from $R_{lateral} = R_{up} \times R^*_{down}$, where R_{up} and R_{down} are the correlation functions obtained from the up and down images, respectively, and R^*_{down} is conjugate of R_{down}. Similarly, the axial phase is obtained from $R_{axial} = R_{up} \times R_{down}$.

Sub-pixel motion is estimated based on the phase of the correlation function at the peak location and the ± 1 lag locations in the axial or lateral dimension. Normally, the phases at these three locations form a straight line, and the zero-crossing point of this fitted line is used to determine the sub-pixel movement. However, due to aliasing, the phases at these three locations can form a V-shape. One strategy is to discard these aliased samples when combining the estimates within a packet. However, this approach would decrease the effective packet size and increase the standard deviation. So, instead, we propose to account for aliasing by compensating either the lag 1 or the lag -1 location by 2π. This ensures that the phases at the three locations fit into a line. As it is not clear whether we should compensate for the lag 1 or the lag -1

location, we consider both cases and keep records of both sub-pixel movements. In the end, when combining the results within a packet, we use samples that do not have V-shape phases to help choose which data should be used for determining the correct sub-pixel movement. As a final step, we apply median filtering in both the axial and lateral dimensions with a window size of 3.

5.1.2 Flow Estimation in Spinning Disk

Estimating flow in a spinning disk is challenging since the estimation has to be accurate over all beam-to-flow angles. The flow is modeled by spinning disk shaped scatterers with the same angular speed. Thus the further a scatterer is away from the disk center, the larger is its speed. In this work, the spinning disk model is borrowed from the SA-VFI challenge [57]. Since the straight vessel datasets from the challenge were used to tune the parameters in velocity estimation, their results are also discussed in this section.

Simulation Setup

Five virtual sources are used in the flow sequence to form an RF image. After every set of five firings for flow estimation, a B-mode firing is initiated. Therefore, the effective pulse repetition frequency (PRF) is one sixth of the actual PRF. Each low resolution image is beamformed with delay-and-sum (DAS) and then summed to form the final image. Since the input is RF data, clutter removal is necessary. Clutter filtering is done with simple mean subtraction, as only stationary clutter is considered here.

Some of the key parameters used in our simulations are listed in Table 5.1. The packet size is 32, which is the same as in our 3D velocity estimation work [42, 43]. We also used 32 samples in the mean subtraction process for clutter filtering. The kernel size is 1.6 mm $\times$ 0.9 mm, in the lateral and axial dimensions, respectively. The

Table 5.1: Simulation Settings

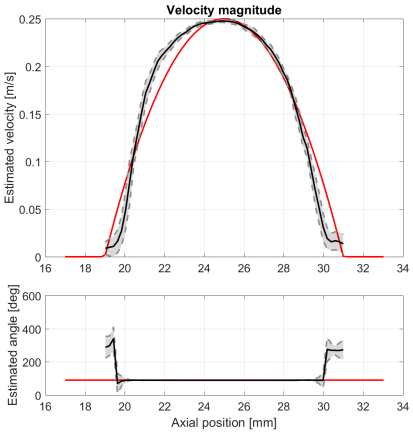
Parameter	**Value**
Number of virtual sources (flow)	5
Pulse repetition frequency (PRF)	5000 Hz
Effective PRF in estimation	5000/6 Hz
Packet size	32
Kernel size (axial)	1.6 mm
Kernel size (lateral)	0.9 mm

search region is selected based on the maximum possible velocity. Note that since the largest velocity is unknown for the testing dataset, a large search region is used to enable estimation of velocity as large as 1 m/s.

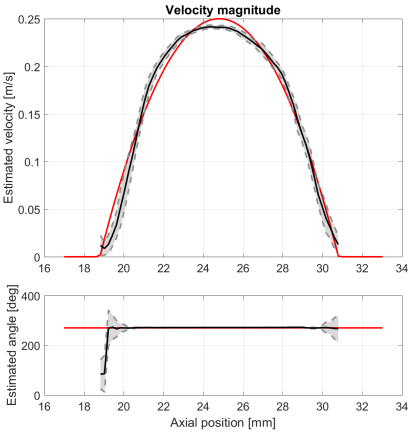
The metrics used are bias and weighted standard deviation, which are calculated over multiple estimated velocity profiles. The weighted standard deviation is given by: $\sigma_{weighted} = \sigma\sqrt{1 + \frac{N}{5}}$, where σ is the standard deviation and N is the packet size used in clutter filtering and velocity estimation. We used $N = 32$.

Straight Vessels

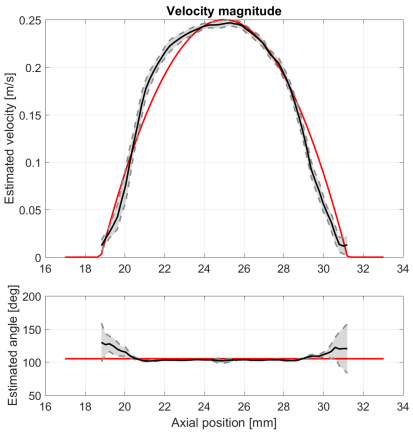
Initial estimation results for straight vessels with inter-scanline distance of 0.3 mm are shown in Fig. 5.1. Although for the most part, the estimations are accurate, there are errors at the vessel boundaries. Since the pitch of the transducer elements is quite large (0.3 mm), if we only include scanlines at the center of the elements—the distance between scanlines is large enough to affect the estimation accuracy. Therefore, we add two beamformed lines between neighboring transducer elements, thereby reducing the inter-scanline distance from 0.3 mm to 0.1 mm.



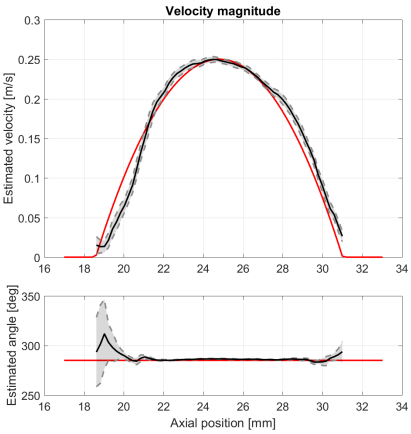
(a) 90^o, simulation



(b) 90^o, measurement



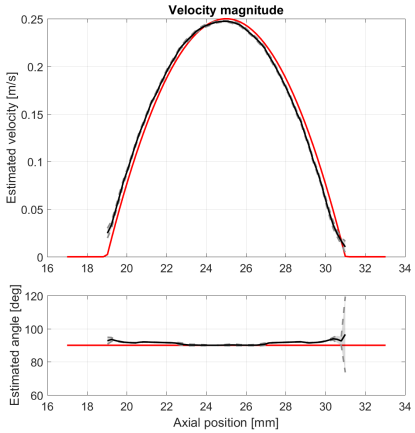
(c) 105^o, simulation



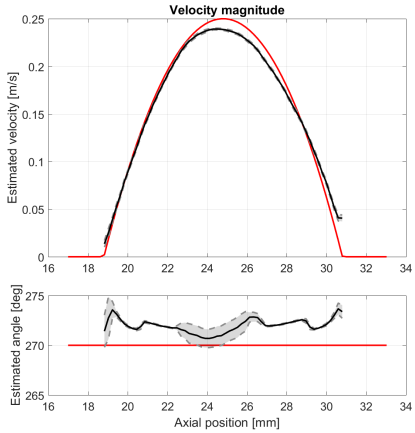
(d) 105^o, measurement

Figure 5.1: Estimation Results for Straight Vessels, with Inter-scanline Distance of 0.3 mm. The Red Line Indicates the True Velocity. The Black Solid Line is the Estimated Velocity. The Grey Dotted Lines Mark the Standard Deviation.

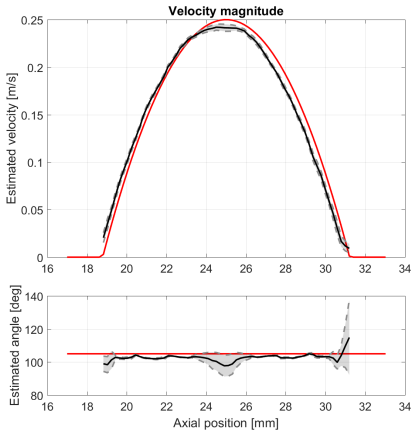
The estimation results for 0.1 mm spacing are shown in Fig. 5.2. Compared to the results for 0.3 mm spacing, the standard deviation is reduced by 52% to 87% for the straight vessel datasets. This improvement arises because reducing scanline distance improves the pixel-level estimation.



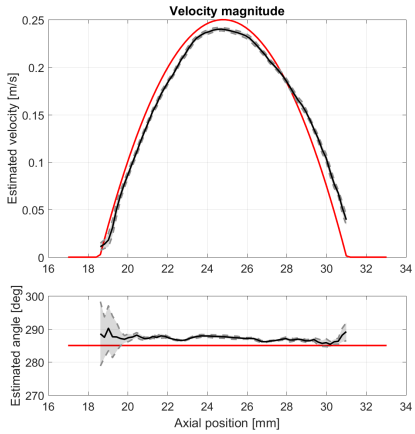
(a) 90^o, simulation



(b) 90^o, measurement



(c) 105^o, simulation



(d) 105^o, measurement

Figure 5.2: Estimation Results for Straight Vessels, with Inter-scanline Distance of 0.1 mm. The Red Line Indicates the True Velocity. The Black Solid Line is the Estimated Velocity. The Grey Dotted Lines Mark the Standard Deviation.

Detailed measurements for all training datasets are shown in Table 6.3. Estimation results for beam-to-flow angle of 90^o is slightly better than for 105^o, with smaller biases and standard deviations. This result confirms that, unlike traditional Doppler method, our method handles flow in the lateral dimension quite well. The standard

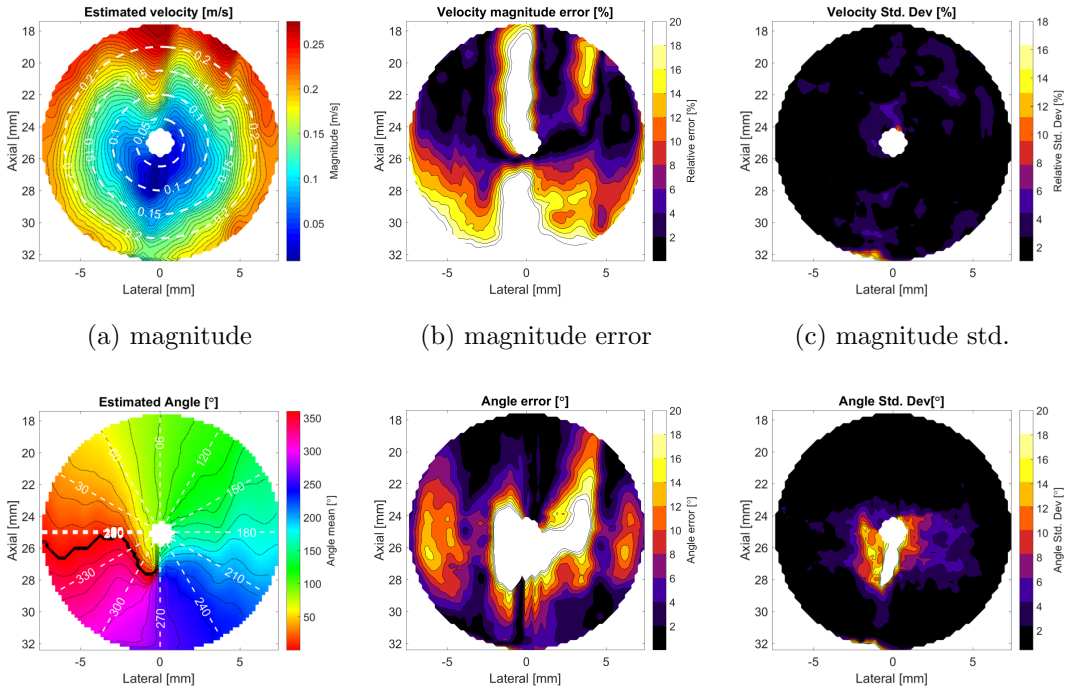
Table 5.2: Final Measurement Results Obtained from the Platform, for Training Datasets.

		Magnitude (%)		Angle (^{o})	
		Weighted std.	Bias	Weighted std.	Bias
90^o	Simulation	2.41	3.24	8.21	1.43
	Measurement	2.12	3.33	1.77	1.99
105^o	Simulation	3.62	4.24	11.55	2.77
	Measurement	3.82	5.44	5.68	2.17
Spinning disk	Simulation	9.65	10.16	16.08	10.16

deviations of the velocity magnitude are within 4%, which is quite low. The standard deviations of the angle are slightly larger for the simulation datasets due to noisy estimates at the vessel boundaries.

Spinning Disk

The estimation results for spinning disk are shown in Fig. 5.3. The estimations of both velocity magnitude and angle are reasonably accurate over all angles with the average bias being about 10% for magnitude and 10^o for angle. The velocity magnitudes have small bias except at the vertical center line of the disk. The angle estimation is also less accurate at the center of the disk. The standard deviation for magnitude is small almost everywhere, while the standard deviation for angle is slightly higher at the disk center.



(a) magnitude (b) magnitude error (c) magnitude std.

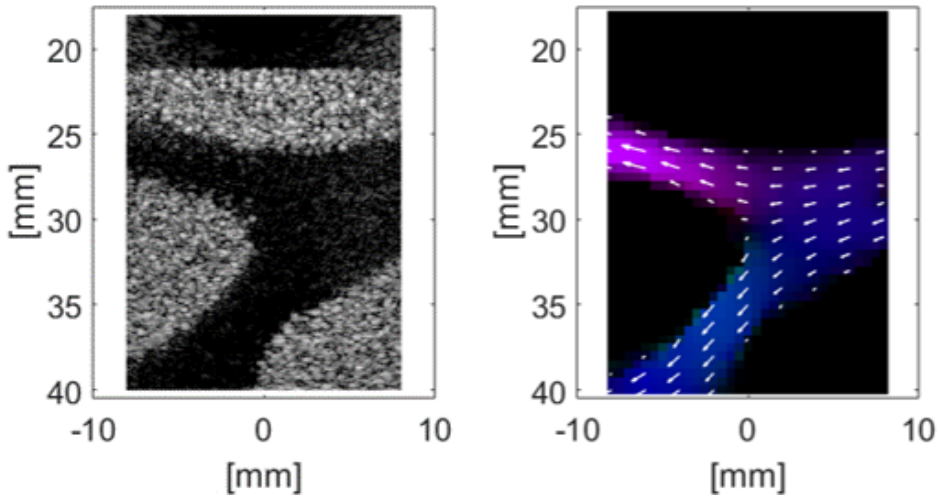
(d) angle (e) angle error (f) angle std.

Figure 5.3: Estimation Results for Spinning Disk, with Scanline Distance of 0.1 mm.

5.1.3 Flow Estimation in Carotid Bifurcation

Carotid artery plays a very important role in human circulatory system. Since it is relatively shallow and has bifurcation, it is extensively used as a modeling target for flow estimation [58]. The bifurcation introduces challenges such as different flow direction and possible reverse flows.

The PRF in the testing dataset is 15 kHz, which is larger than the 5 kHz PRF in the training datasets. Fig. 5.4a shows the B-mode image and Fig. 5.4b shows the vector flow image for this dataset. We can see that the flow pattern in the carotid bifurcation has been captured well with estimated velocity reducing at the vessel boundary and increasing when the vessel narrows (in the upper branch). The average magnitude bias is 8% and the average angle bias is 7^o.



(a) B-mode image (b) Estimation results

Figure 5.4: B-mode Image and Estimation Results for the Testing Dataset.

5.2 Flow Estimation in Vessels with Stenosis

Stenosis is believed to be related to stroke and ischemia, including asymptotic stenosis. For instance, in a study in [19], stenosis in carotid artery has been shown to result in higher stroke risk when the stenosis is greater than 75%. However, if treated properly in early stage, the risk of stroke or any cardiovascular death can be reduced [59].

While in vitro and in vivo experiments can provide real data to estimate flow in vessels with stenosis, the true velocity field cannot be extracted directly. So we make use of CFD simulation to provide the ground truth velocity profiles. In this work, the velocity field is generated using COMSOL [60], wihch is then coupled with Field II simulation with interpolation and regridding [58, 61]. In Field II simulation, the point scatterers are moved based on the velocity field in every frame, which may cause dilution of aggregation. In order to avoid this problem, we propose to reset the scatter

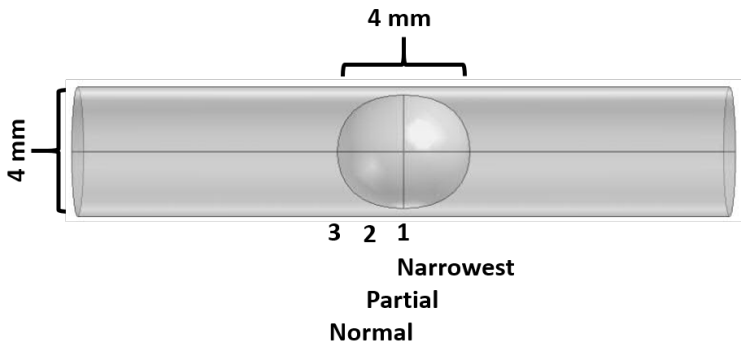


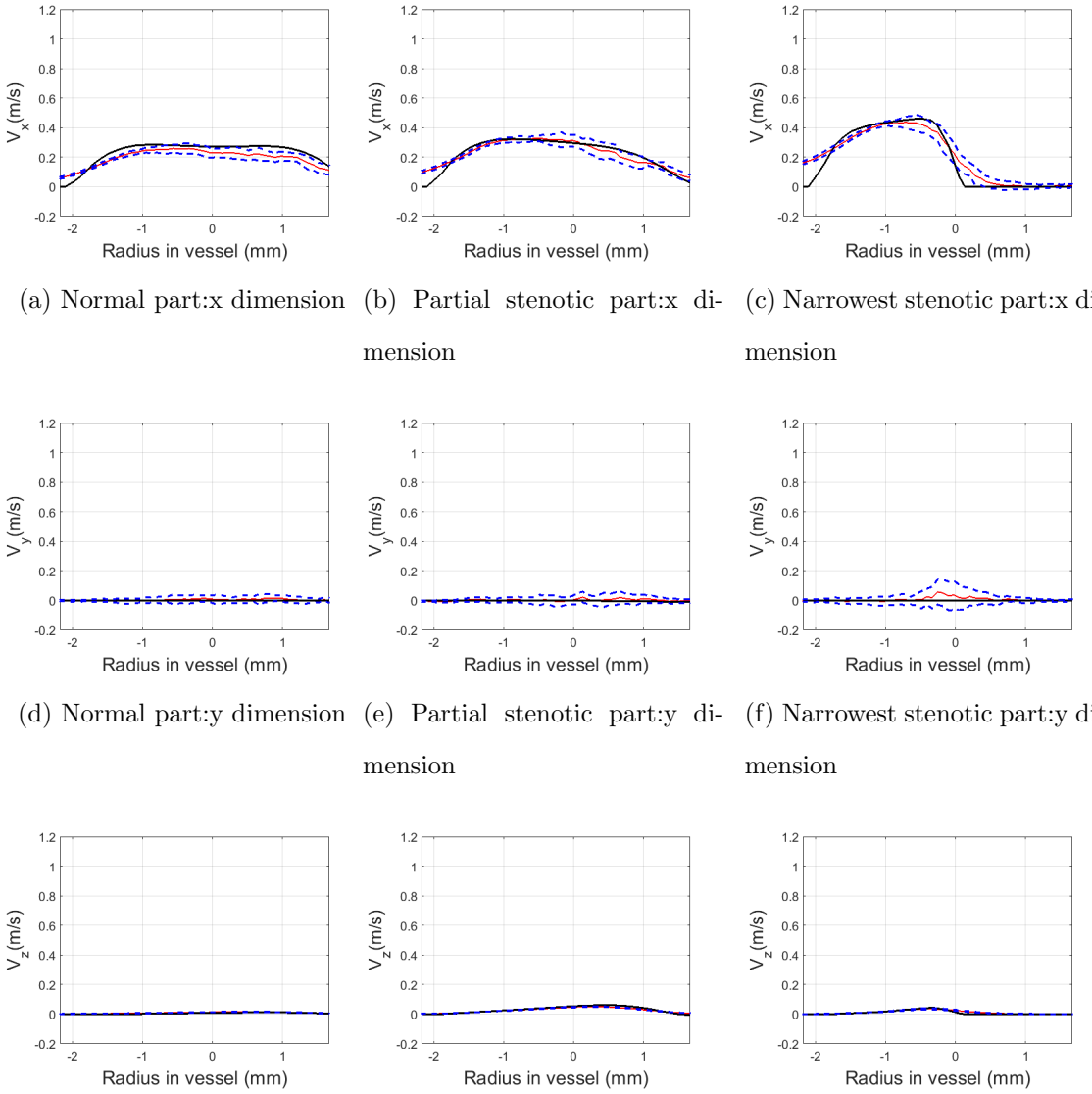
Figure 5.5: The Stenotic Vessel Model with Geometry

density every frame, thus keeping the scatterer distribution constant throughout the simulation. The velocity estimator is the two-tiered method described in Chapter 2.

5.2.1 Single Stenosis

We start with a single, semi-sphere stenosis in a cylindrical vessel. The diameters of the stenosis and the vessel are both 4 mm. Blood is modeled as an incompressible Newtonian fluid with dynamic viscosity and density of the blood are 3.5 mPas and 1050 kg/m^3. The initial setting at the inlet is plug flow with a 0.2 m/s velocity. The outlet boundary has a constant pressure of 100 mmHg. The CFD model geometry is presented in Fig 5.5. Other simulation setting are the same as Table 2.1 (in Chapter 2).

The velocity estimation results using the flow estimator proposed in Chapter 2 are shown in Fig. 5.6. In general, the estimated velocities match the simulated velocities with good accuracy except at the vessel or stenosis edges where the velocity change is sharp. At the narrowest part of the stenosis, the lateral-x velocity has larger bias than other parts. Since our estimation is based on speckle tracking which is a kernel based method, the velocities within a kernel are averaged, resulting in bias. This is more severe when the velocity gradient is large. In this case, reducing kernel size could help



(a) Normal part:x dimension (b) Partial stenotic part:x dimension (c) Narrowest stenotic part:x dimension

(d) Normal part:y dimension (e) Partial stenotic part:y dimension (f) Narrowest stenotic part:y dimension

(g) Normal part:z dimension (h) Partial stenotic part:z dimension (i) Narrowest stenotic part:z dimension

Figure 5.6: Estimation Results for Single Stenosis Model. (a)(d)(g) Normal Part; (b)(e)(h) Partial Stenotic Part; (c)(f)(i) Narrowest Stenotic Part. Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean±1 Standard Deviation.

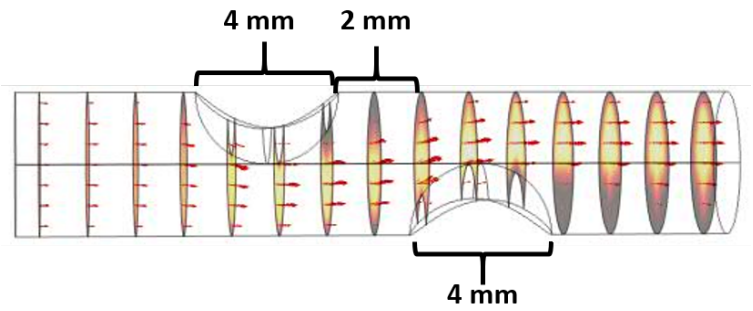


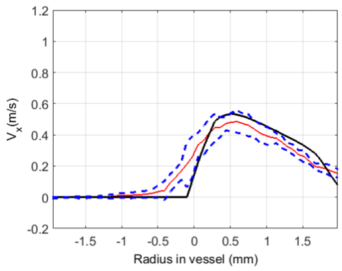
Figure 5.7: The Stenotic Vessel Model with Geometry

reduce the bias, but the standard deviation is expected to be higher. Since the flow for the double stenosis model is more complex, in the next section, we investigate this tradeoff on a vessel with double stenosis.

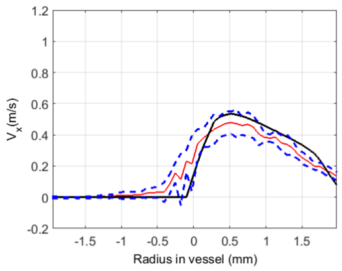
5.2.2 Double Stenosis

We consider a vessel with double stenosis. We model the vessel as a cylindrical tube with 4 mm diameter and two semi-spherical stenoses (4 mm diameter) placed 2 mm apart on opposite walls. Blood modeling parameters, inlet and outlet conditions all have the same settings as that of the single stenosis model described in Section 5.2.1. The CFD simulated output with the model geometry is presented in Fig. 5.7.

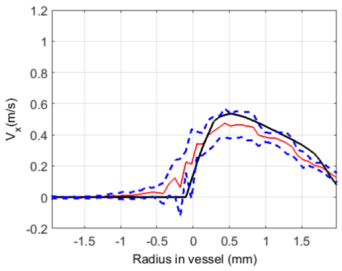
We first investigate flow estimation accuracy based on kernel size. As we can see from Fig. 5.8, reducing the kernel size helps reduce the bias in the primary flow direction (V_x), especially at the edge of the stenosis. This is reasonable as a larger kernel tends to average the velocities within it. However, smaller kernel size increases the standard deviation in both lateral dimensions. For instance, in lateral-x dimension, the average standard deviation is increased from 10% to 15% and 18% for the half-size and minimum-size kernel. Fortunately, increasing the packet size from 32 to 48 can shrink the standard deviation to within 12% for half-size kernel.



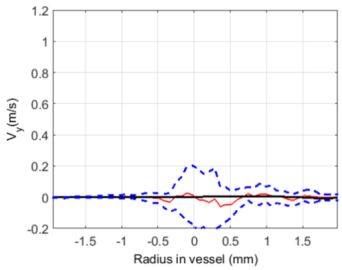
(a) Normal-size kernel V_x



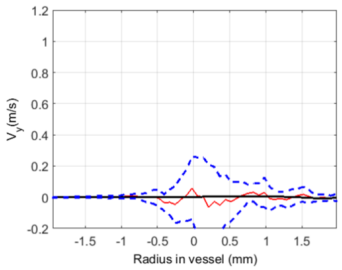
(b) Half-size kernel V_x



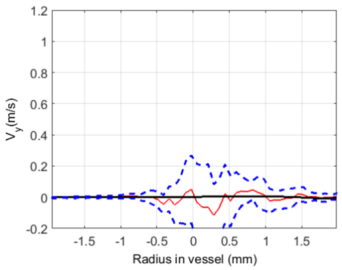
(c) Minimum-size kernel V_x



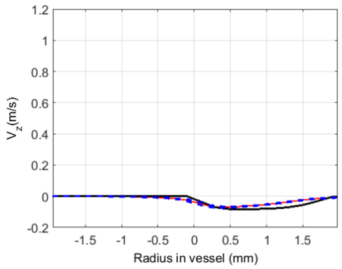
(d) Normal-size kernel V_y



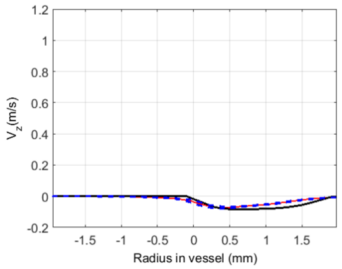
(e) Half-size kernel V_y



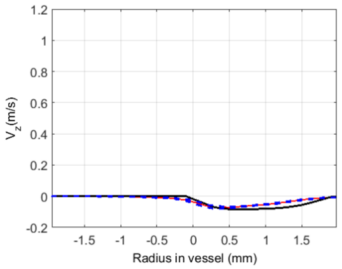
(f) Minimum-size kernel V_y



(g) Normal-size kernel V_z



(h) Half-size kernel V_z



(i) Minimum-size kernel V_y

Figure 5.8: Estimation Results for Double Stenosis Model (Left Capture). (a)(d)(g) Normal-size Kernel: 25×3×3; (b)(e)(h) Half-size Kernel: 13×3×3; (c)(f)(i) Minimum-size Kernel: 3×3×3;. Dark Solid Lines Represent the Actual Velocity, Red Solid Lines Represent the Mean of the Estimated Velocity, Blue Dashed Lines Represent the Mean±1 Standard Deviation.

Table 5.3: Flow Rate Estimation for Double Stenosis Model (Beam-to-flow-angle is 90^o). The Real Flow Rate is 150 mL/min.

Slice	Left	Middle	Right
	Avg.±Std (mL/min)		
1	142.19±4.44	187.67±6.39	
2	136.84±4.08	150.60±4.37	
3	138.95±5.24	141.85±3.39	
4	143.36±5.96	141.12±4.87	
5	149.76±5.97	142.84±3.79	171.84±5.49
6	170.62±6.04	148.69±4.61	145.14±5.19
7		149.50±3.39	146.26±6.68
8		151.47±8.39	145.28±4.65
9		140.95±4.60	140.01±6.19
10		120.01±6.65	135.55±3.86

The increase in packet size means reducing temporal resolution of flow estimation. However, our flow estimation can still be up to 40 estimations per second with a PRF of 2 kHz.

One of the benefits of using plane wave imaging for flow estimation is that the estimation can cover multiple scanlines without moving the scanhead. However, since imaging quality usually gets worse at the edge, we need to investigate the imaging range we can trust for flow estimation. Our 2D transducer array covers about 10 mm in both lateral dimension. Since the simulated vessel is 20 mm long, it is not possible to cover the entire vessel without moving the transducer. So we put the transducer array at three different positions to generate three different sets of flow estimation

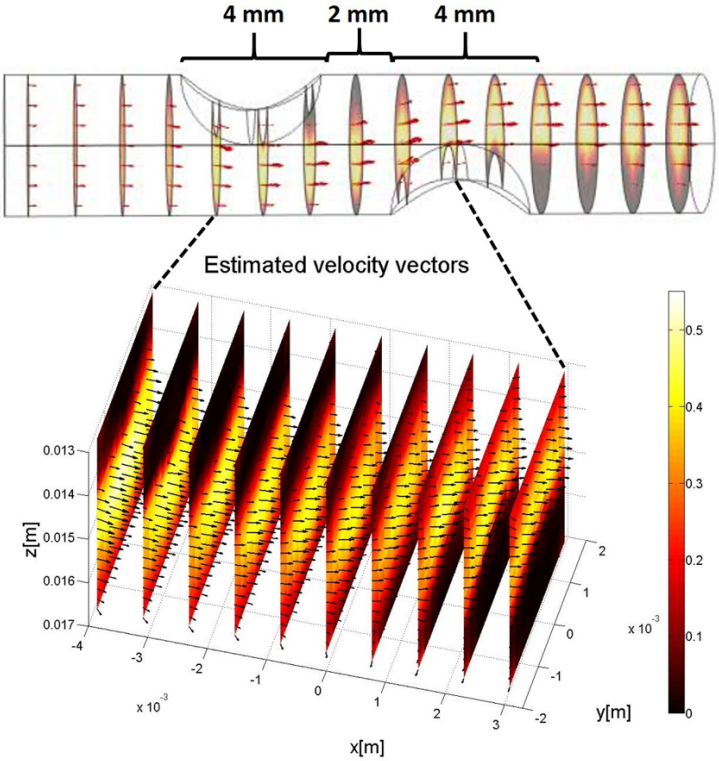
results. We first start with putting the transducer on top of center of the vessel, and then we move the transducer 3 mm left and right, allowing overlap between the three sets (left, middle and right). The flow estimation results are shown in Fig. 5.9, 5.10, 5.11. The flow rate estimation results for the middle capture, along with the overlapped region for the left and right captures are shown in Table 5.3. We can see that only the results in the first and tenth slice are not trustworthy, meaning the flow imaging range can be as large as 5 mm in lateral dimensions.

5.3 Summary

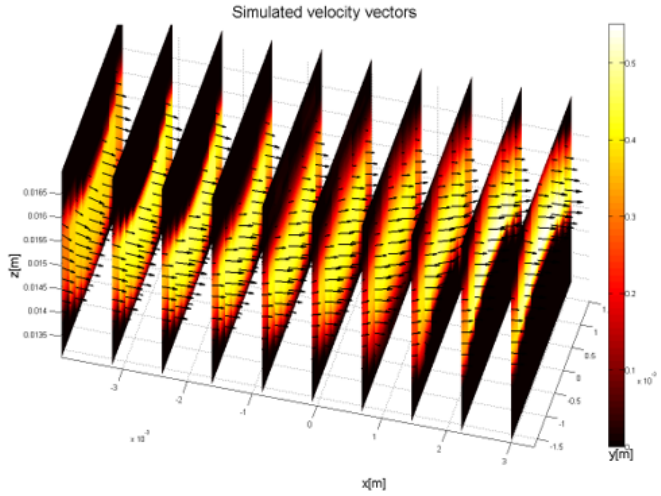
In this chapter, we demonstrated that our velocity estimator is capable of accurately estimating 2D or 3D velocity vectors in challenging models, for both synthetic aperture and plane wave imaging systems.

In the SA-VFI challenge, for the four straight vessel datasets, the estimated velocity profiles closely match the true velocity profiles, with small biases and standard deviations. For the spinning disk dataset, the estimation is accurate in all places except at the vertical center line of the disk. For the testing dataset, we can see that the flow pattern in the carotid bifurcation has been captured well. Our method with 8% magnitude bias and 7^o angle bias ranked 3rd in the challenge.

Velocity estimation in vessels with stenosis is quite accurate except when the velocity gradient is large near the stenosis boundary. We showed that reducing the kernel size results in smaller bias but larger standard deviation. We also showed that our flow rate estimation is accurate over a 5 mm range in lateral dimensions.

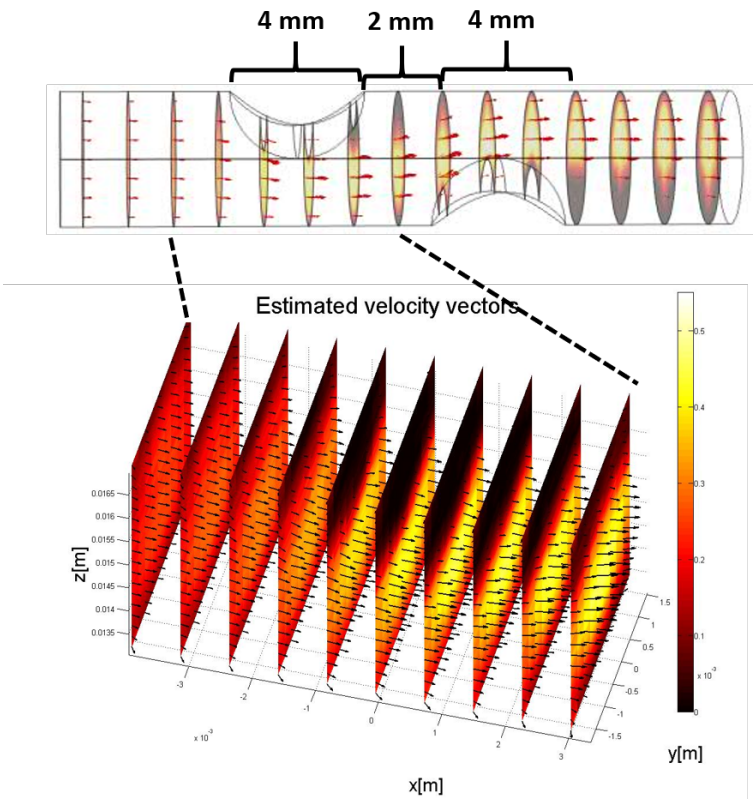


(a) Estimated velocity vectors

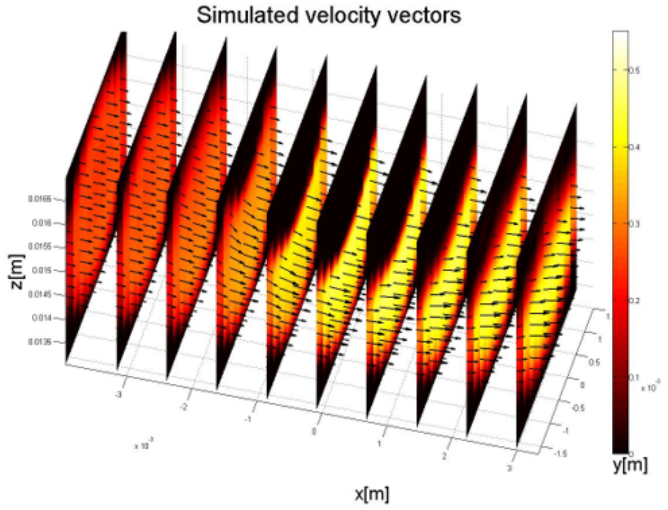


(b) Simulated velocity vectors

Figure 5.9: Middle Set. Black Arrows Represent Velocity Vectors. Color Represents Velocity Magnitudes.

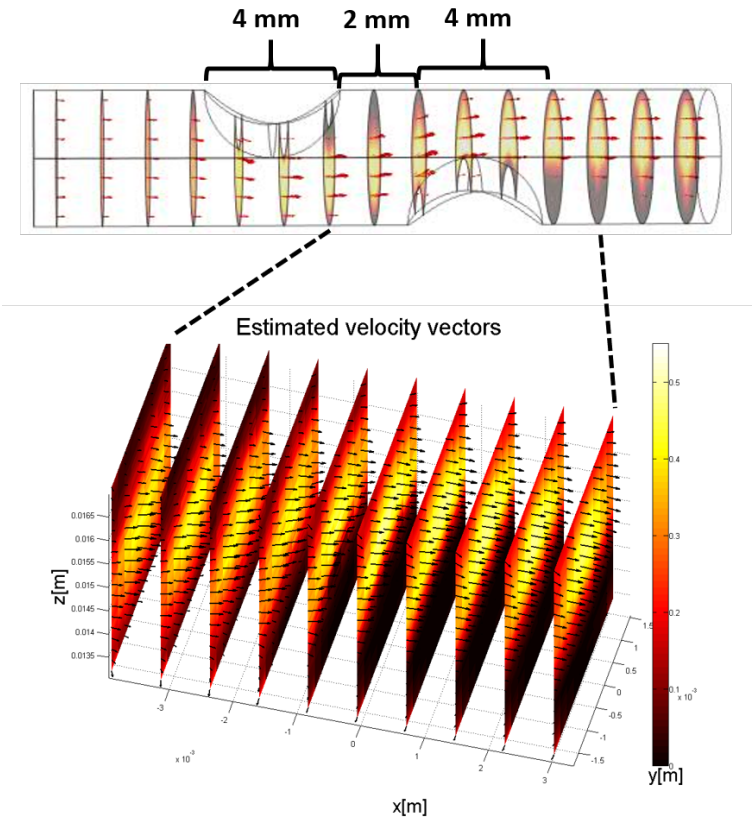


(a) Estimated velocity vectors

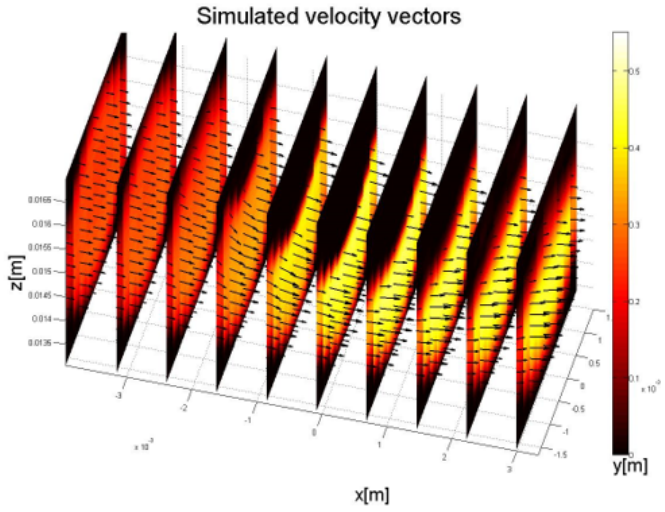


(b) Simulated velocity vectors

Figure 5.10: Left Set. Black Arrows Represent Velocity Vectors. Color Represents velocity Magnitudes.



(a) Estimated velocity vectors



(b) Simulated velocity vectors

Figure 5.11: Right Set. Black Arrows Represent Velocity Vectors. Color Represent Velocity Magnitudes.

Chapter 6

PLANE WAVE IMAGING CHALLENGE IN MEDICAL ULTRASOUND

Plane wave imaging has been popular in medical ultrasound community due to its high frame rate. Compared to linear/phased array beamforming or synthetic aperture ultrasound, which needs multiple transmits to generate a frame, plane wave imaging is capable of generating a frame in every transmit. While the image quality can be sub-optimal due to lack of transmit focusing, it can be compensated by techniques such as compounding [30]. In flow estimation, high pulse repetition frequency is desirable, which makes plane wave imaging a suitable beamforming scheme for this application [18].

Although plane wave imaging has been the topic of numerous papers, different researchers have used different simulation settings, making it hard to compare the performances. In 2016, the International Ultrasonics Symposium (IUS) initiated a open competition (PICMUS) where all participants use the same raw data (channel data before beamforming) to generate B-mode images. Such an effort helped provide a fair platform to evaluate the performance of different beamforming techniques [62].

PICMUS provided both simulated and experimental datasets. The two simulated datasets were generated using Field II [44], the simulation parameters are listed in Table 6.1. The first simulated dataset includes vertically and horizontally distributed scatterers as shown in Fig. 6.1a. The second simulated dataset includes vertically and horizontally distributed cysts as shown in Fig. 6.1b. In addition, two experimental datasets were provided, with one focusing on resolution of the point spread functions and the other focusing on contrast ratio of the cysts. The experimental datasets are also shown in Fig. 6.1.

Table 6.1: Simulation Setup

Parameter	**Slow moving clutter**
Pitch	0.30 mm
Element width	0.27 mm
Element height	5 mm
Elevation focus	20 mm
Number of elements	128
Aperture width	38.4 mm
Transmit frequency	5.208 MHz
Sampling frequency	20.832 MHz
Pulse bandwidth	67%
Excitation	2.5 cycles

The PICMUS challenge consists of two stages. In the first stage, the participants are given the training datasets described earlier, and asked to submit their best results to the competition. In the second stage, blind datasets are used to avoid overfitting, and the organizer is charged with writing a journal paper discussing the results. For both datasets, we were able to enhance the contrast ratio without affecting other metrics.

6.1 Stage I Evalutation

6.1.1 Metrics and Scoring Scheme

B-mode image quality is usually measured using two metrics: resolution and contrast. In the challenge, the resolution is measured by full width half maximum

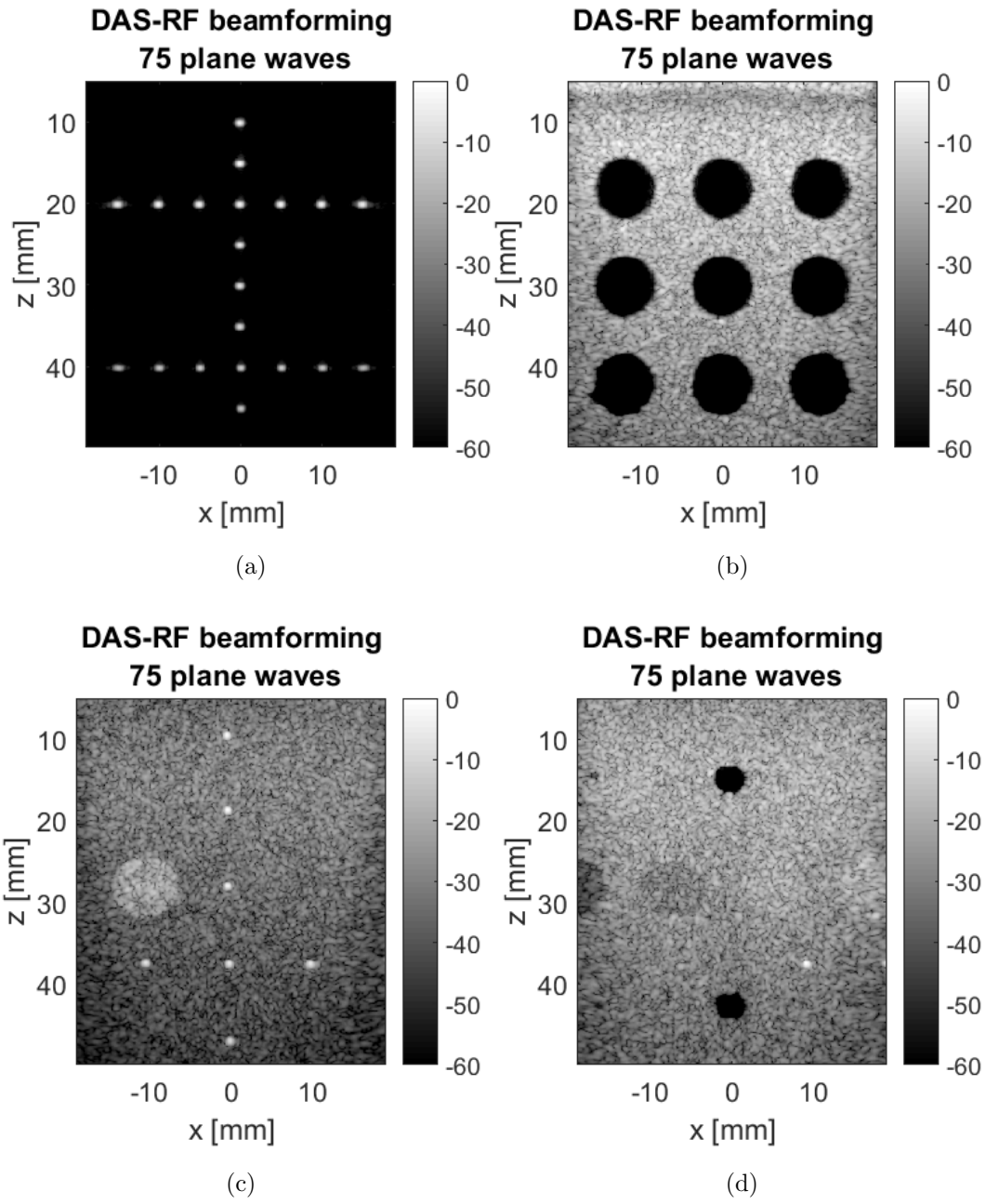


Figure 6.1: Example of the Simulated (a and b) and Experimental (c and d) Datasets.

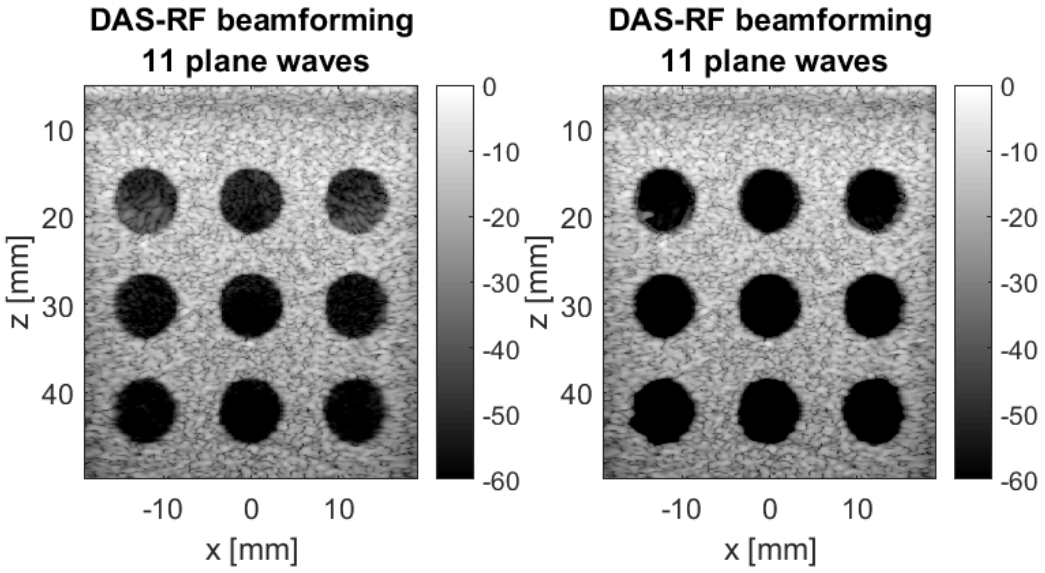
(FWHM) in both axial and lateral dimensions [62]. All the scatterers are measured and the average values are used in the scoring system.

Contrast-to-noise ratio (CNR) is used as the metric for contrast.It is calculated as $CNR = 20log_{10}(\frac{|\mu_{in}-\mu_{out}|}{\sqrt{(\sigma_{in}^2+\sigma_{out}^2)/2}})$, where μ_{in} and μ_{out} are averaged gray level values in and out of the cyst area, and σ_{in} and σ_{out} are the standard deviations of the gray level values in and out of the cyst area. CNR values of all the cysts in an image (Fig. 6.1b and 6.1d) are averaged to provide a single CNR value for each image.

The performance on four datasets is measured using either FWHM or CNR. A score is given based on the ranking among all participants. The best team gets 100 points and the worst team gets 0 point for a certain dataset, and the scores for all datasets are summed up to provide the final score. There are four categories in the challenge. The first three categories require the participants to use 1, 11, 75 firings for compounding (named C1, C11 and C75), while the fourth category allows the participants to choose the number of firings and normalized the score with the number of firings (named Cfree). Both geometrical distortion and speckle quality violation are penalized by deducting 40 points from the total score. If the scatter positions are off by more than a wavelength, the geometrical distortion test is said to fail. If the speckle quality in several predefined regions fails to follow Raleigh distribution, the speckle quality test is said to fail.

6.1.2 Choice of beamforming parameters

We choose to use delay-and-sum (DAS) beamforming since it is easy to implement and has low complexity [16]. In the context of this challenge, parameters such as apodization window, f-number are important for good image quality. We tested different apodization windows and found that Hamming and Tukey50 windows provide best CNR results for cysts. Rectangular window gave the best resolution performance but the worst CNR. In general, a smaller f-number provided better FWHM but worse CNR. We use a f-number of 1 to maintain a good balance between the two metrics.



(a) Proposed algorithm without Step 4 (b) Proposed algorithm

Figure 6.2: Speckle Contrast Images for Simulation Data with or without Step 4 of Our Proposed Algorithm. Images are Shown with 60 dB Dynamic Range.

For Cfree and C11 categories, we are allowed to choose arbitrary firings from the 75 firings provided. It is reasonable to always include the central firing (38th) as it is with 0 firing angle and provides best image quality. We choose to use 3 firings in the Cfree categories. Table 6.2 shows the comparison between Hamming and Tukey50 windows with different firing indices combinations. We actually tested all 37 possible combinations for C3 and found that firing sequence (35,38,41) is the best for CNR and firing sequence (1,38,75) is the best for resolution. Our goal in the challenge was to maximize the CNR, and so we chose (35,38,41) for Cfree. For similar reasons, we chose (23,26,29,32,35,38,41,44,47,50,53) for C11.

Table 6.2: Choice of Apodization Window and Firing Indices. Scores were Obtained from the Ranking System on the Platform. Hamming Window with Firing Indices of 35, 38 and 41 is the Best Choice.

Window	Firings	CNR (dB)		Axial, Lateral FWHM (mm)		Score
		Simulation	Experiment	Simulation	Experiment	
Hamming	1,38,75	8.45	9.07	0.57, 0.62	0.40, 0.50	98.8
	35,38,41	**10.20**	**13.50**	**0.57, 1.18**	**0.40, 1.06**	**136.3**
	36,38,40	9.95	12.68	0.58, 1.19	0.40, 1.07	128.1
	37,38,39	9.70	13.47	0.57, 1.20	0.40, 1.08	136.3
Tukey 50	1,38,75	7.70	8.46	0.57, 0.49	0.41, 0.49	77.5
	35,38,41	9.45	11.61	0.57, 1.00	0.40, 0.91	123.8
	36,38,40	9.20	11.02	0.57, 1.00	0.40, 0.92	115.6
	37,38,39	9.10	11.29	0.57, 1.00	0.40, 0.93	120.0

6.1.3 Post Processing

In cyst imaging, the cyst area is sometimes contaminated with fill-in originating from surrounding tissue due to insufficient aperture size (side-lobes) or aperture sampling (grading-lobes). To obtain a better contrast ratio for the cyst images, we used a post-processing technique based on edge detection. While in practice, the sonographers can manually select the area for post processing, here we propose an automated procedure, which detects the edges of a cyst and then smoothes the cyst areas. This is necessary in the context of this challenge, as we are required to use the same method for both the cyst images and point target images.

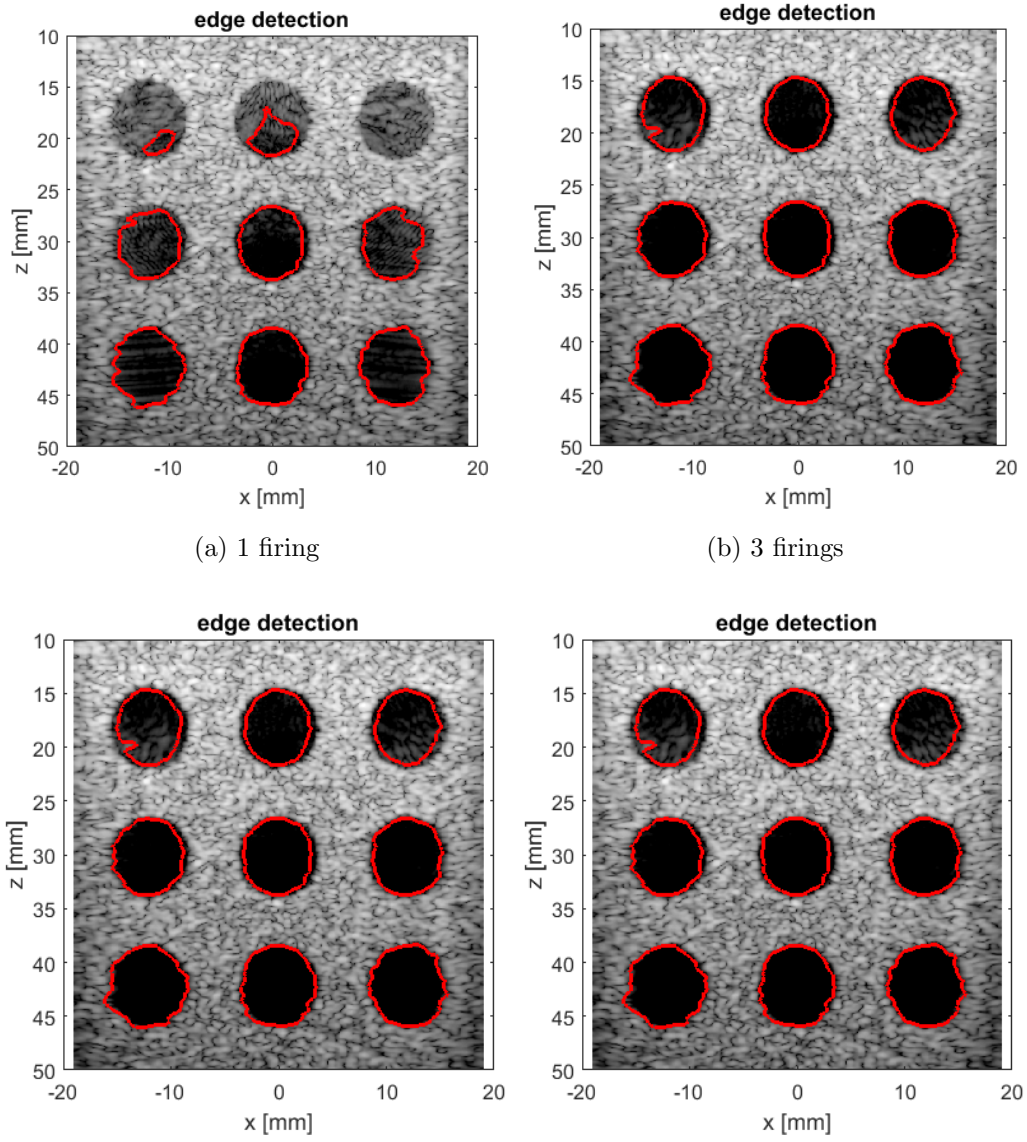
Our post-processing scheme is summarized in Algorithm 4. The original image is blurred with a 20×20 Gaussian window to reduce speckle in the background, thus

Algorithm 4: Proposed post-processing algorithm steps

1. Blur the image with a large kernel size (20×20 Gaussian window) to reduce speckle noise.
2. Use Canny edge detector for edge detection.
3. Identify cyst area using contour tracing.
4. Reduce the intensity of the pixels in the cyst area by 1/10.
5. Use 3×3 Gaussian window for smoothing in the cyst areas.

preventing the selection of low amplitude speckle spots as small cysts (Step 1). Canny edge detector is used for edge detection (Step 2) since it is known to be effective in detecting weak edges [63]. After the edges are detected, we exclude those edges that do not form a closed curve. We use contour tracing to determine the cyst areas (Step 3). Step 4 reduces the intensity of the pixels in the cyst area by 1/10, which helps further increase the contrast between the cyst and the background. Finally, a 3×3 Gaussian window is used for smoothing the cyst area (Step 5).

Note that adjusting the intensity of the pixels in the cyst makes this procedure sensitive to the edge detection quality. Including Step 4 helps increase the CNR by about 1 dB for the given data sets; the visual improvement can also be seen in Fig. 6.2. The proposed algorithm would have to be adjusted if there are any hyperechoic lesions. Adjustment in edge detection needs to be made to make the algorithm work for hyperechoic cysts. However, it would be possible to set both positive and negative thresholds and apply these to hyperechoic and hypoechoic lesions, respectively, in the same image.



(a) 1 firing

(b) 3 firings

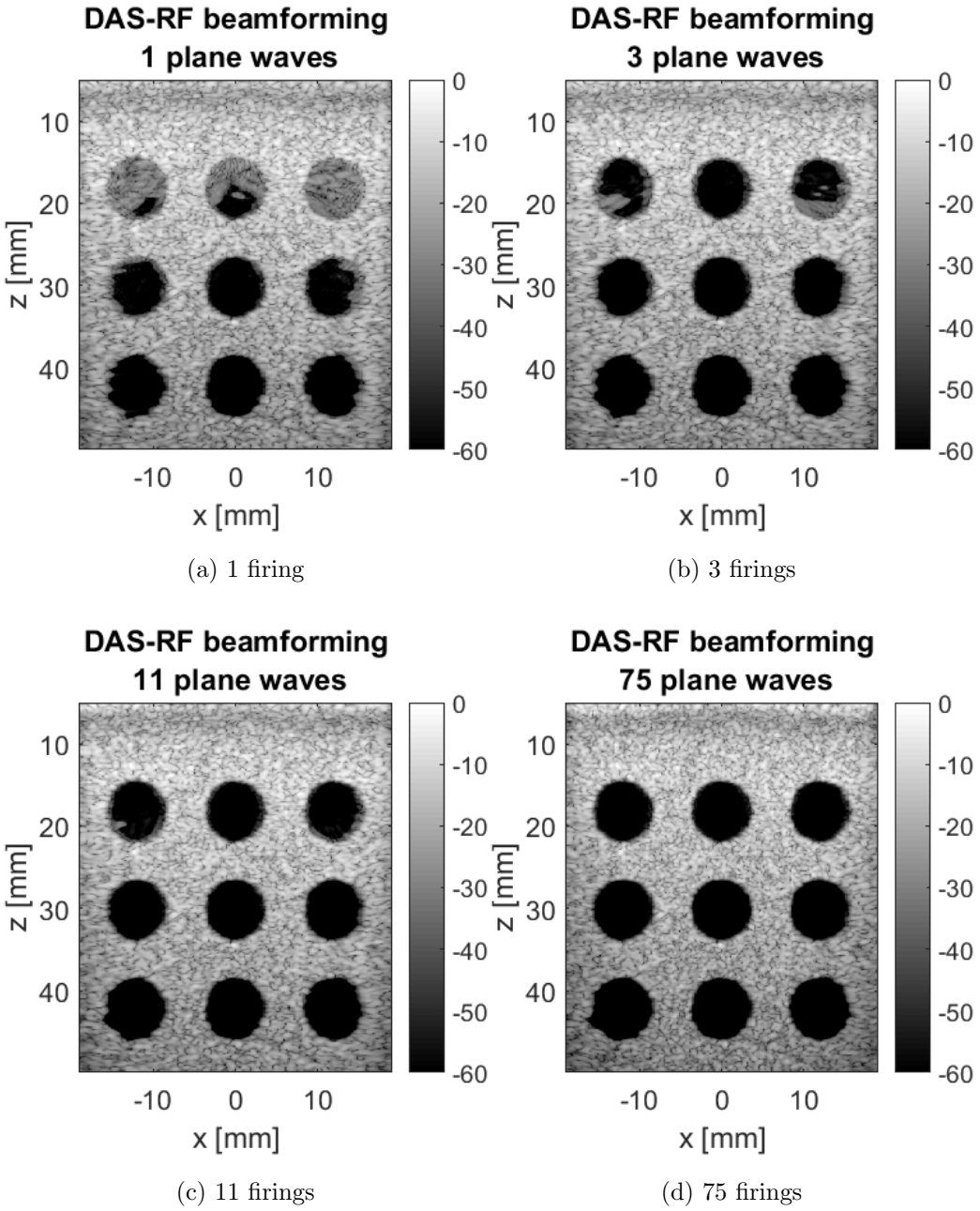
(c) 11 firings

(d) 75 firings

Figure 6.3: Edge Detection Results for the Simulation Data. Red Circles Indicate the Cyst Areas Detected by the Algorithm. Cyst Images are Shown with 50 dB Dynamic Range.

6.1.4 Results

First, the edge detection results for the simulation data are shown in Fig. 6.3. The edge detection performs well, except for the first row of the 1 firing case where



(a) 1 firing

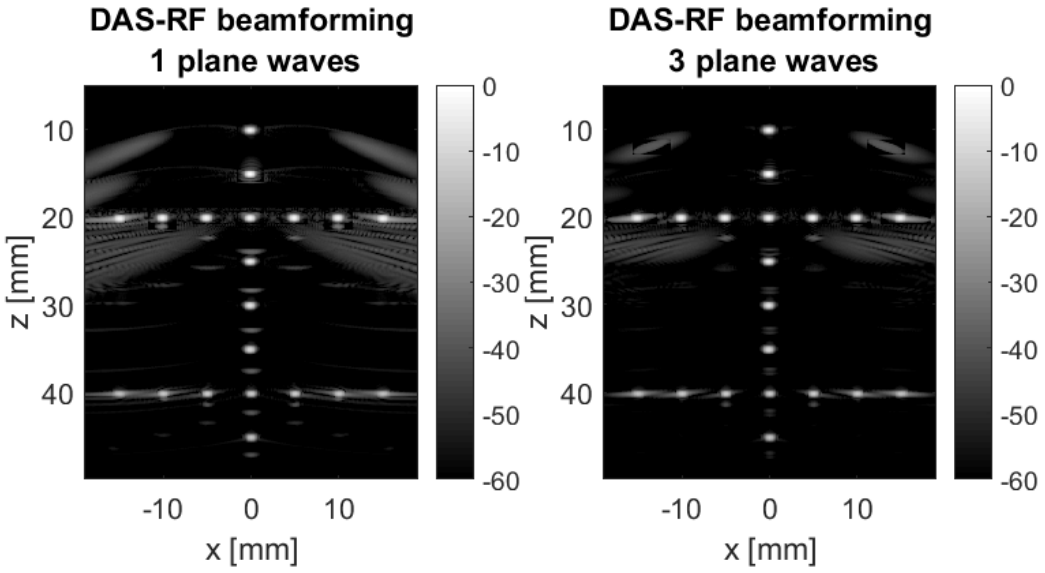
(b) 3 firings

(c) 11 firings

(d) 75 firings

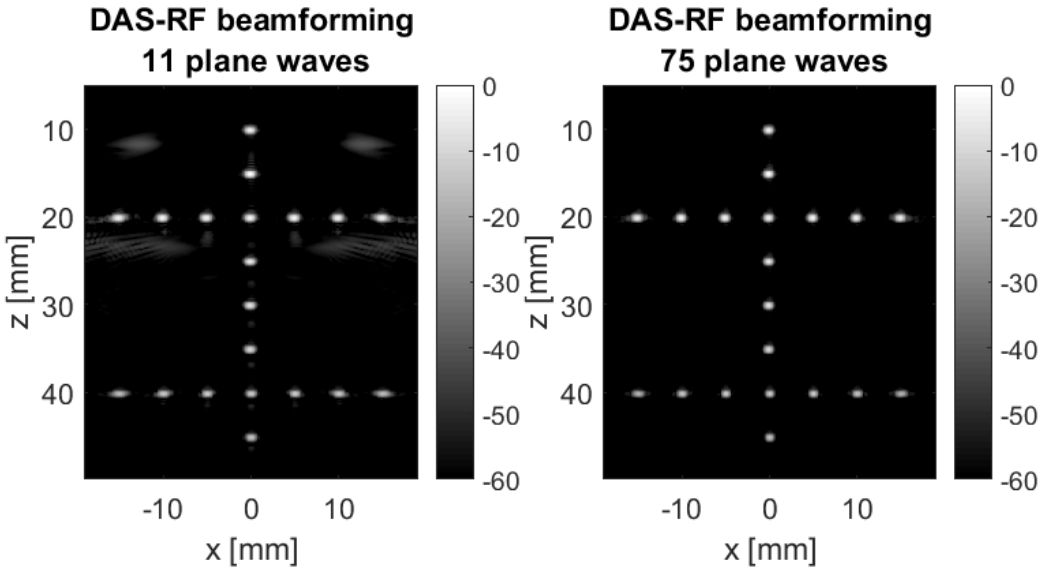
Figure 6.4: Speckle Contrast Images for Simulation Data with Different Number of Firings. Images are Shown with 60 dB Dynamic Range.

it almost fails. This is expected as there are severe sidelobes in those cysts. The final cyst images with simulation data are shown in Fig. 6.4. The CNR measurement



(a) 1 firing

(b) 3 firings



(c) 11 firings

(d) 75 firings

Figure 6.5: Point Target Images for Simulation Data with Different Number of Firings. Images are Shown with 60 dB Dynamic Range.

results are also shown in Table 6.3. Compared to the images before edge detection and filtering, the CNR improves by 0.97 dB, 1.43 dB, 2.12 dB and 2.29 dB, for 1,

Table 6.3: Final Measurement Results Obtained from the Platform.

		1 firing	3 firings	11 firings	75 firings
Contrast (dB)	Simulation	12.79	15.10	17.30	19.09
	Experiment	11.45	13.80	14.95	15.45
Axial resolution (mm)	Simulation	0.40	0.40	0.40	0.41
	Experiment	0.57	0.57	0.56	0.56
Lateral resolution (mm)	Simulation	0.71	0.70	0.66	0.54
	Experiment	0.90	0.90	0.77	0.55

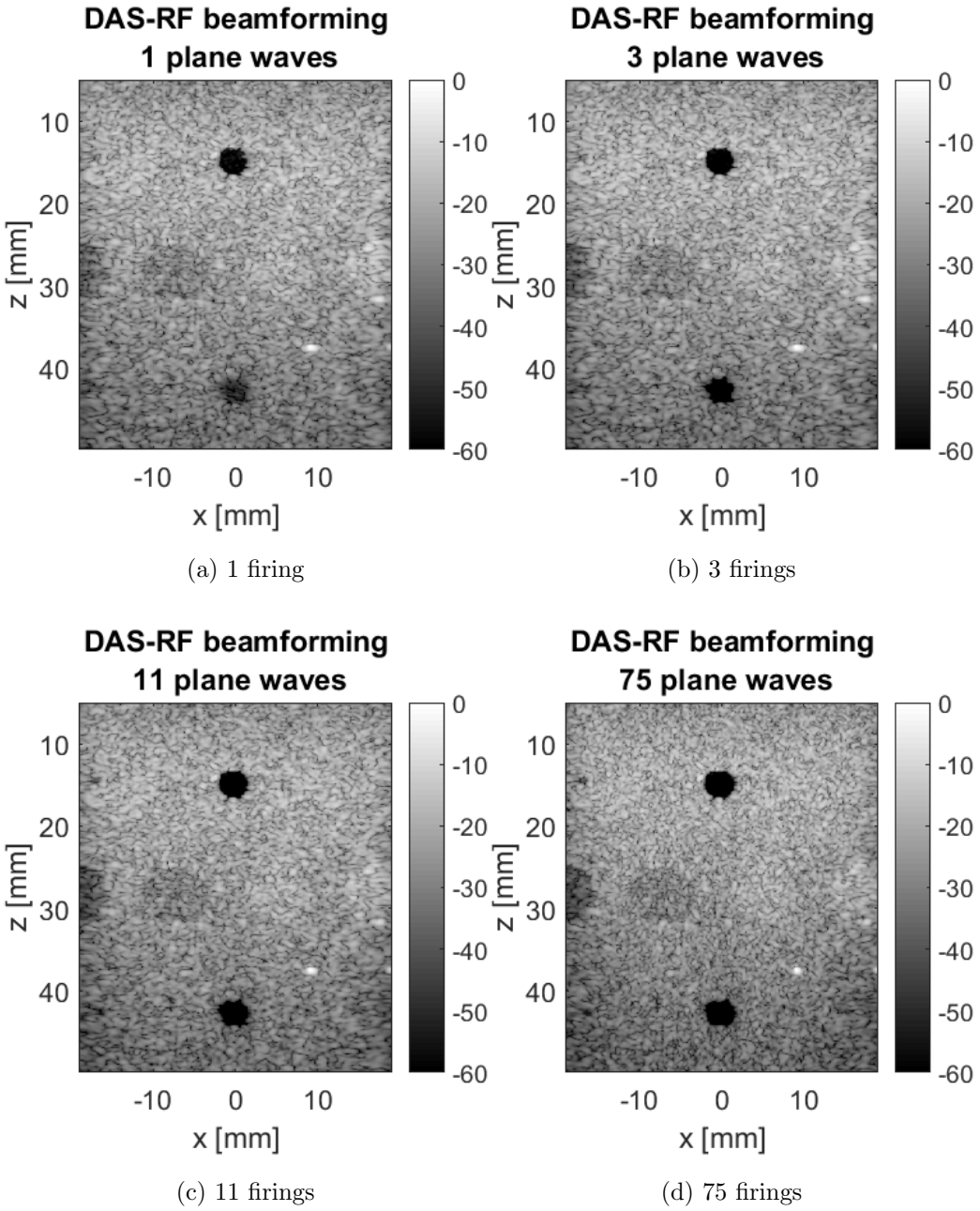
3, 11, 75 firing(s) respectively. As the number of firings increases, the image quality improves, the edge detection is more effective, and so the improvement in CNR is more significant.

The same conclusion can be drawn for the CNR results with the experimental data. In fact, for the 1 firing case, the image quality was too poor for the edge detector to detect the deeper cyst, while both cysts were detected for the other 3 cases.

As for the point target images, edge detector does not help improve the resolution as no edges are detected. As our parameters were chosen towards having a better CNR in the CNR-resolution tradeoffs, the resolution results are not as good.

6.2 Stage II

After the conference, the organizers of the challenge improved their datasets and provided a new set of metrics. Participants were required to submit their code and the organizers then ran the code with blind datasets. The results were provided as feedback to the participants. Similar to the first stage, both simulated and experi-



(a) 1 firing

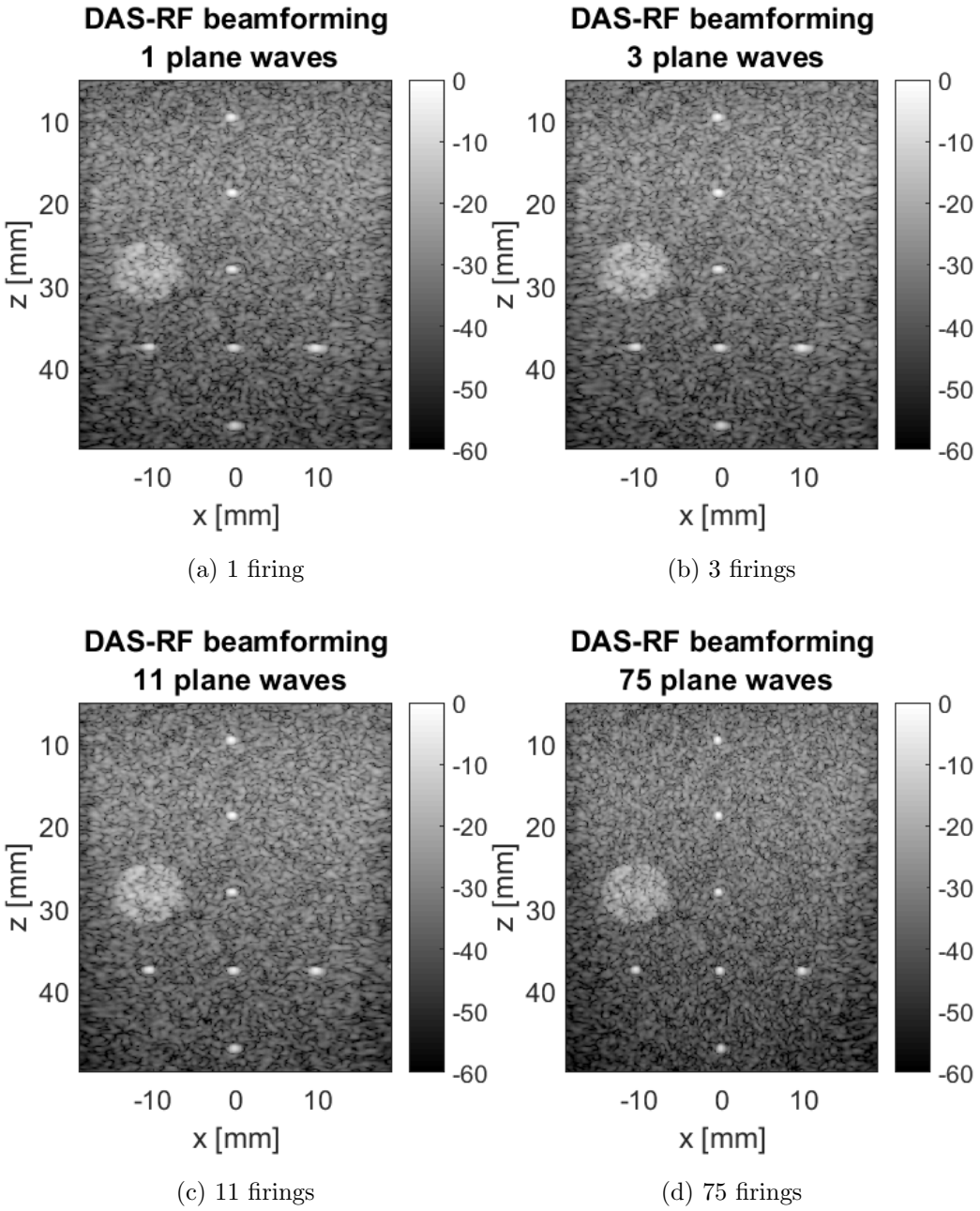
(b) 3 firings

(c) 11 firings

(d) 75 firings

Figure 6.6: Speckle Contrast Images for Experimental Data with Different Number of Firings. Images are Shown with 60 dB Dynamic Range.

mental datasets were included in the second stage. In fact, one simulated and three experimental datasets were designed for the challenge. As shown in Fig. 6.8, they



(a) 1 firing

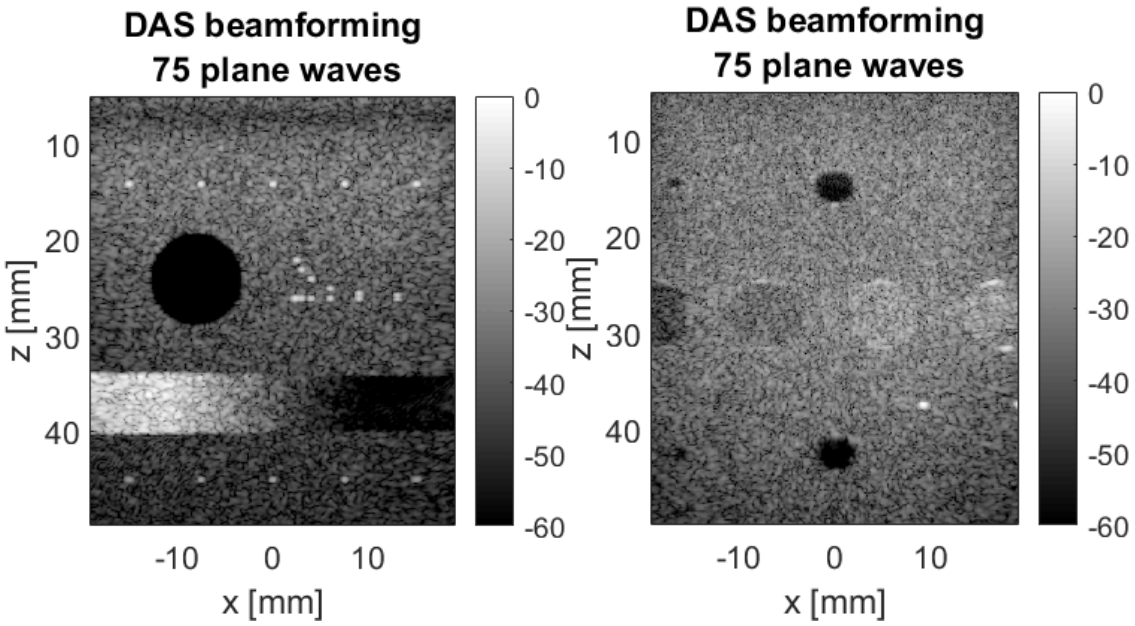
(b) 3 firings

(c) 11 firings

(d) 75 firings

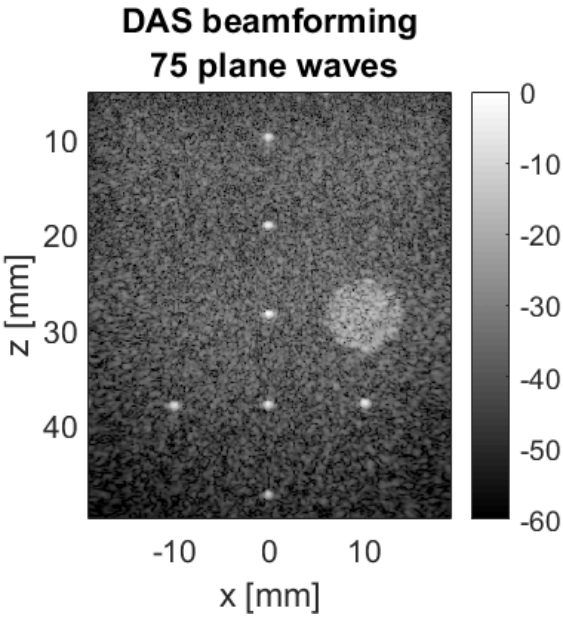
Figure 6.7: Point Target Images for Experimental Data with Different Number of Firings. Images are Shown with 60 dB Dynamic Range.

were named as numerical, and in-vitro type 1-3. In this section, we describe the new aspects of the challenge and discuss the results.

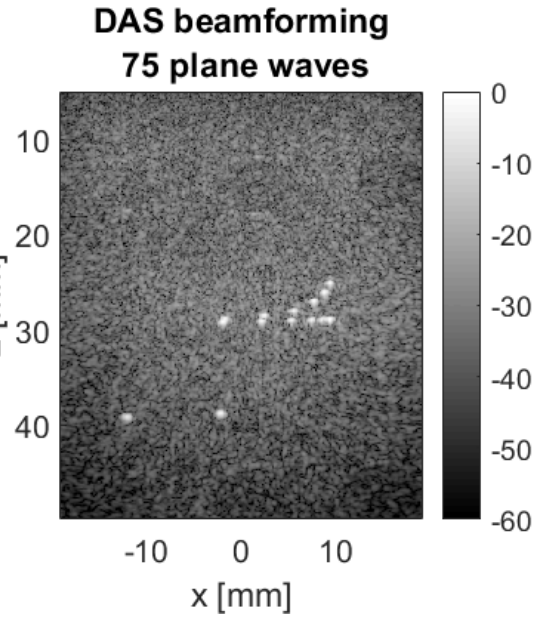


(a) Numerical

(b) In-vitro type 1



(c) In-vitro type 2



(d) In-vitro type 3

Figure 6.8: Datasets in PICMUS Stage II. Images are Generated with Compounding of 75 Firings and Shown with 60 dB Dynamic Range.

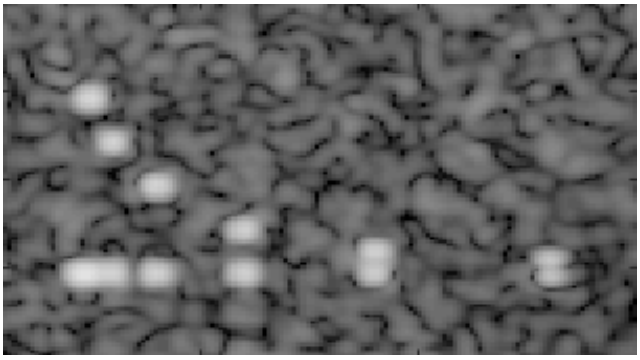


Figure 6.9: The Scatterer Distribution for Measuring Image Resolution.

6.2.1 New Metrics

Four metrics were used in the first stage of the challenge, including FWHM, CNR, geometrical distortion and speckle quality. In addition to these four metrics, the second stage of the challenge introduced two more metrics, resolution and intensity linearity [64].

The resolution was measured by varying the distance between two scatterers and checking if two local maxima can be found. As shown in Fig. 6.9, the distance varies in both axial and lateral dimensions. The separations between scatterers are 4, 3, 2, 1, 0.5, 0.25 mm. If two local maxima is not found for a certain separation distance, the resolution is determined as the next larger separation distance [64].

In order to prohibit intensity transformation on the ultrasound image which would harm the fairness of the CNR based comparison, a speckle region with linear intensity variation along lateral dimension was added in the numerical dataset. As shown in Fig. 6.10, if the intensity linearity in the predefined region does not hold, the test fails [64].

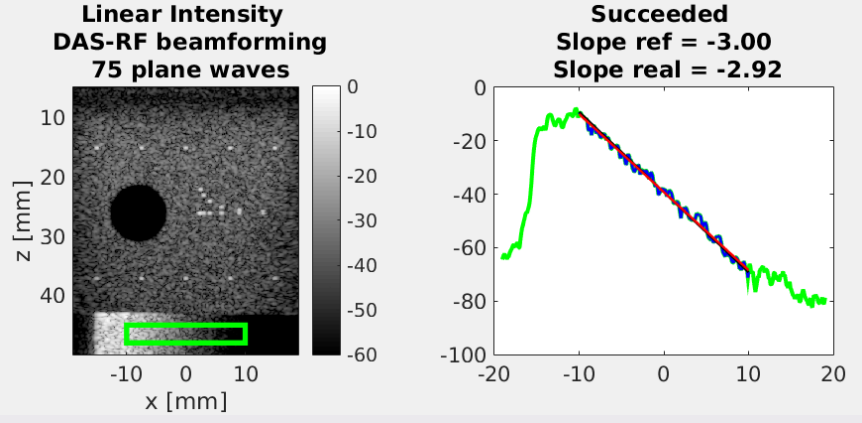
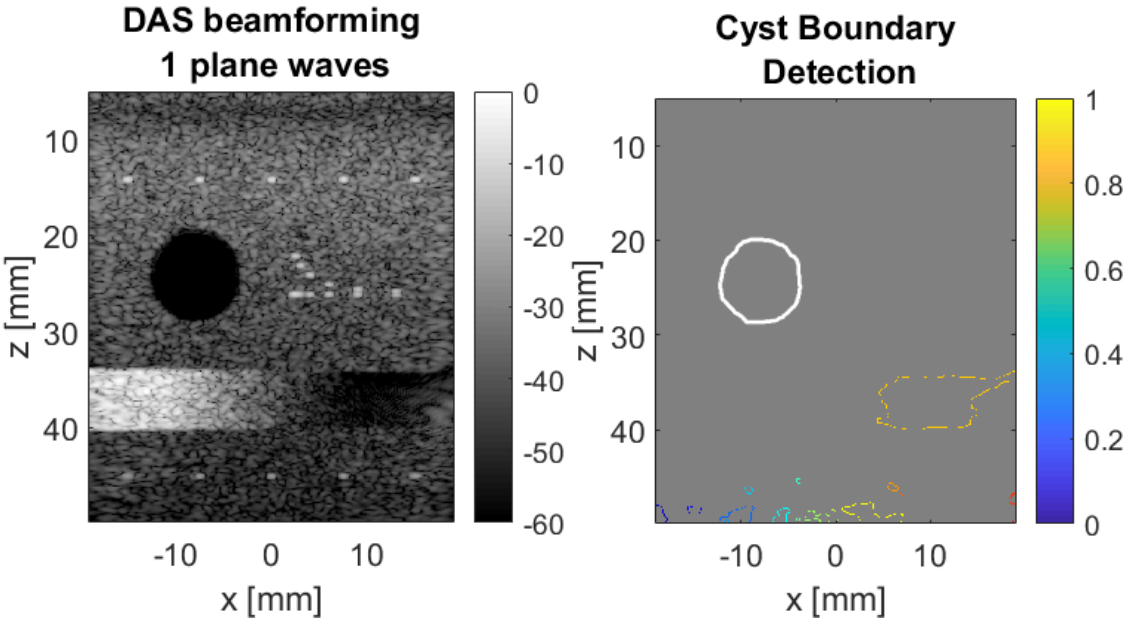


Figure 6.10: Illustration of the Intensity Linearity Test.

6.2.2 Improved Method

In order to cope with the new datasets and metrics, we modified our previous method. Basically, we improved our cyst boundary detection accuracy, so that only the real cyst area is processed. The key to making the method more robust is to replace hard thresholds with adaptive ones. In our new method, the dynamic range of the image before using Canny detector is adaptively chosen based on the histogram of the image. After the edge detection, only closed loop boundaries that are significantly larger than other small holes are preserved as cyst areas. The effectiveness of the improved method can be seen from Fig. 6.11, where only the real cyst area is identified (white circle). Small holes in the deeper part of the image, which exist because of the change of intensity level over depth, are not identified as cysts. A drawback of this method is the possibility of missing small cysts. However, we would rather miss some small cysts than distort images if cysts are wrongly detected.



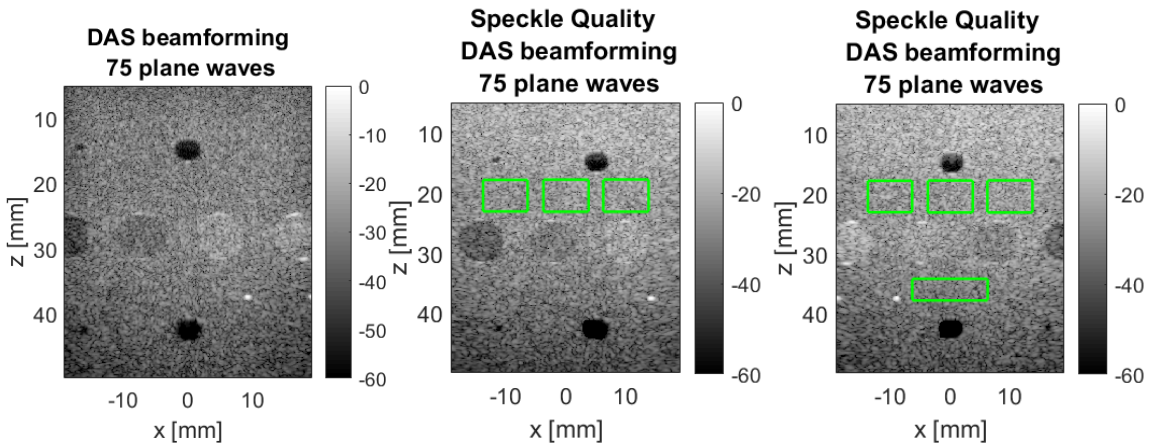
(a) Final image

(b) Cyst Detection

Figure 6.11: Our Improved Method is Able to Accurately Identify Cysts. Images are Shown with 60 dB Dynamic Range.

6.2.3 Results

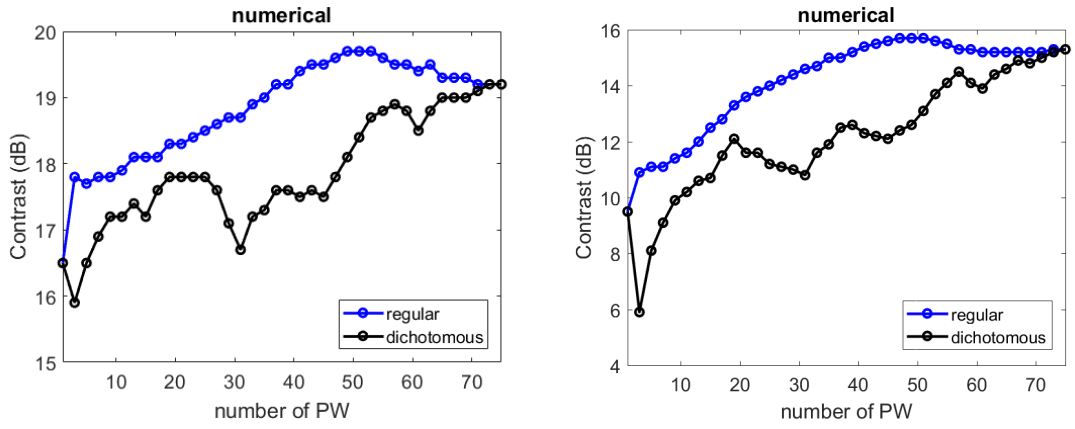
As mentioned before, the participants submitted codes and the organizer generated the results which can be accessed at [65]. Except for the numerical dataset, the other datasets (in-vitro type 1-3) have 6 realizations. One of the realization is the same as the training set provided to the participants, and the other five have varied cysts or point scatteres locations. Fig. 6.12 compares two varied realizations of in-vitro type 1 datasets with the original one, where the entire image is shifted to the right in Fig. 6.12b and the point scatterer location is shifted to the left in Fig. 6.12c. These realizations are named in-vitro type 1-1 through 1-6 in this section. The use of multiple realizations avoids overfitting to the training sets in algorithm development.



(a) In-vitro type1-1 (b) In-vitro type1-2 (c) In-vitro type1-4

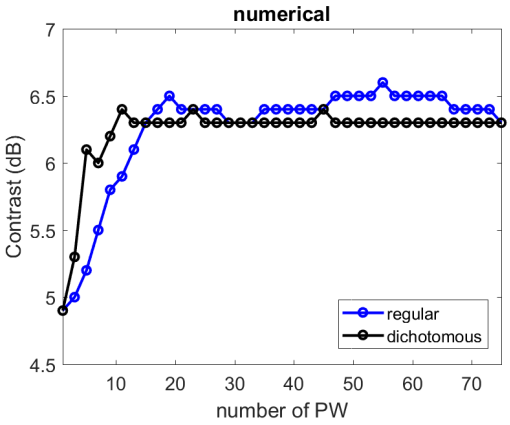
Figure 6.12: Illustration of the Various Realizations of In Vitro Type 1 Datasets.

There were a total of 14 participants in Stage 2 of the challenge. We compare with two of the top performers in the competition, namely Szasz and Deylami who were the top 3 in at least one of the four categories in the competition. In [66], Szasz et. al. formulated the beamforming as an inverse problem and solved it using Laplacian priors through basic pursuit. They achieved excellent performance in both resolution and contrast. However, the images generated by their method do not hold the speckle characteristics usually seen in ultrasound images. In [67], Deylami et. al. used a modified version of minimum variance (MV) beamforming method. They were also able to achieve excellent performance in both resolution and contrast. However, similar to Szasz's method, the generated images did not look like a normal ultrasound image. Moreover, minimum variance beamforming is known to be highly computationally complex. Possibly for this reason, only the single firing plane wave imaging results were available online for Deylami's method. In addition to these two other participants' results, we also include the classic delay-and-sum (DAS) results for comparison.

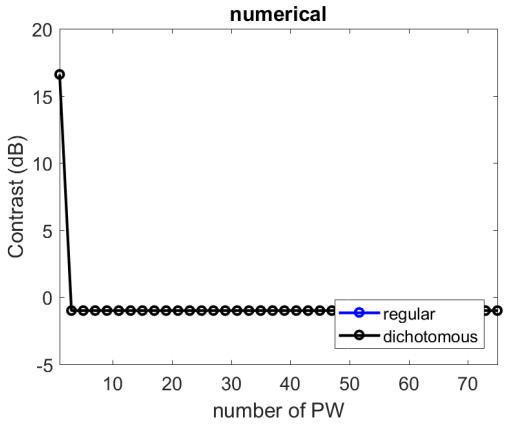


(a) Our method

(b) DAS



(c) Szasz's method



(d) Deylami's method

Figure 6.13: CNR results for Numerical Dataset. (a) Our method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. Our Method Has Best CNR for All Cases. Improvement Over DAS is More Than 4 dB.

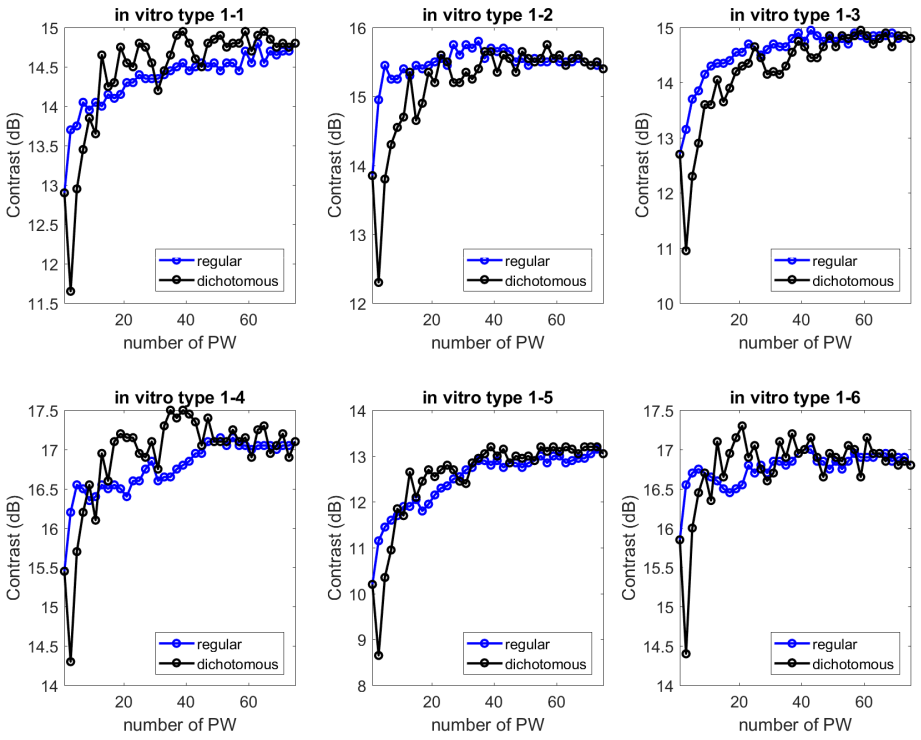


Figure 6.14: CNR Results for In-vitro Type 1 Dataset Using Our Method. Significant Improvement Over DAS in All Cases except for In-vitro Type 1-5.

Fig. 6.13 shows the results of CNRs for numerical dataset. We can see that our method has the best CNR for all numbers of firings. The improvement over DAS is more than 4 dB. Fig. 6.14, 6.15, 6.16 and 6.17 show the results of CNR for in-vitro type 1 dataset. We can see that our method has the best CNRs except for in-vitro type 1-5. For the other five cases, significant improvement over DAS exists, but not for type 1-5. As mentioned in Section 6.2.2, this is because our method missed cysts in type 1-5. Szasz's method performed much better for in-vitro type 1 than for numerical dataset. However, as shown in Fig. 6.18 and 6.20, Szasz's method always fails the speckle quality test for numerical dataset and occasionally fails the test for

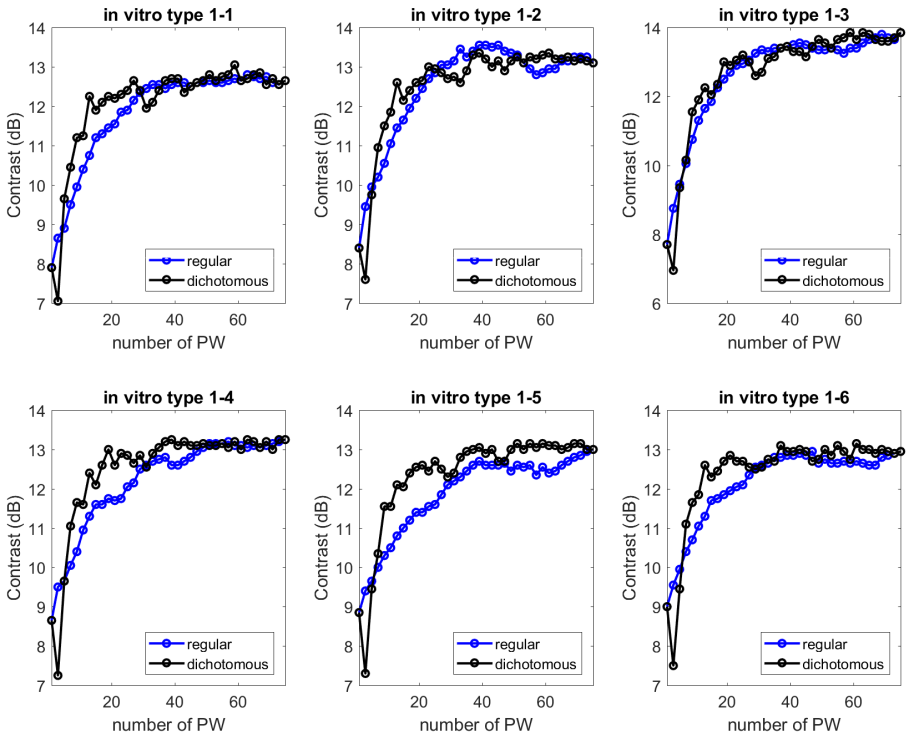


Figure 6.15: CNR Results for In-vitro Type 1 Dataset Using DAS. CNR is Lower than Our Method in All Cases except for In-vitro Type 1-5.

in-vitro type 1 dataset. In contrast, our method does not fail the speckle quality test for either datasets.

Fig. 6.21 and 6.23 compares the results of FWHMs and resolutions in axial dimension for numerical dataset. Four methods have comparable results for these two metrics. This is also true for all the in-vitro datasets and so we do not present them graphically. In the lateral dimension, the results of resolutions for four methods are identical, as shown in Fig. 6.24. However, Deylami's method provides much better FWHM result for the single firing case than the other three methods in lateral dimension, as shown in Fig. 6.22. For all these four metrics, our proposed method has

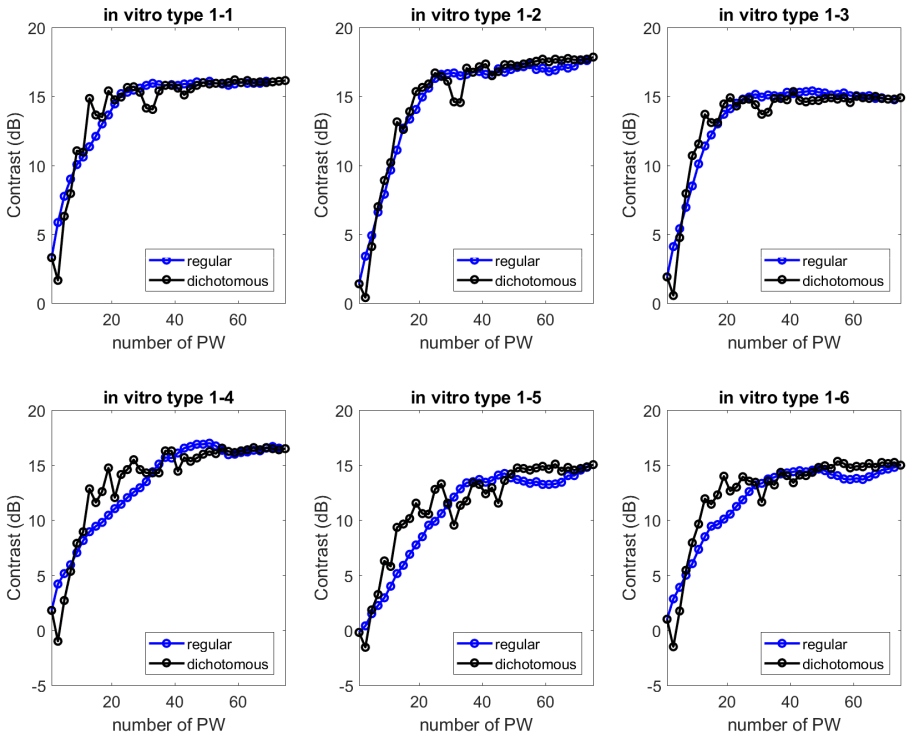


Figure 6.16: Results of CNRs for In-vitro Type 1 Dataset Using Szasz's Method[66]. Szasz's Method is Comparable to Our Method for In-vitro types 1-1, 1-2, 1-3 and 1-5 and Worse than Our Method for Types 1-4 and 1-6.

comparable performances to the DAS method, which confirms that our method does not affect other image regions with falsely identified cysts.

6.3 Summary

In this chapter, we presented our work in the plane wave imaging challenge (PICMUS). Our method is based on DAS with judicious choice of parameters and inclusion of post processing. The purpose of post processing is to improve the contrast performance of the cysts. We proposed to use automatic cyst detection after beamforming, and then manipulate the intensities within the cyst region. Results have proven our

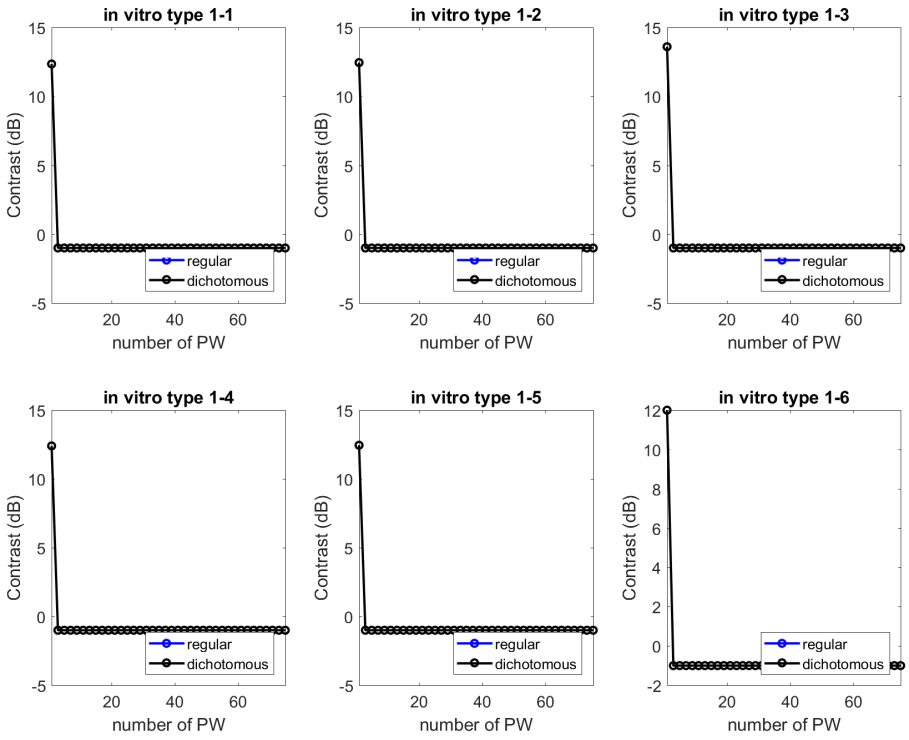
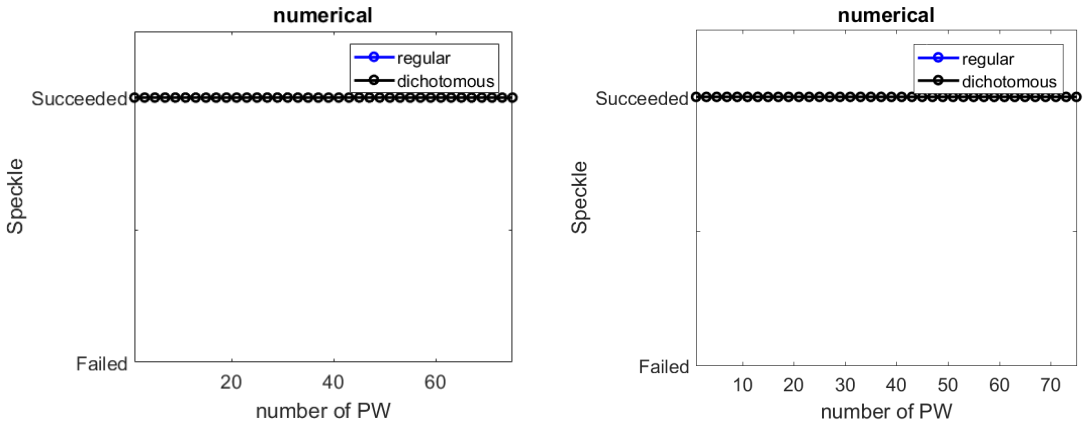


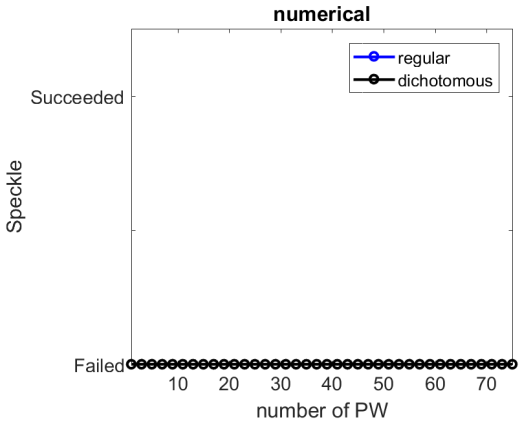
Figure 6.17: Results of CNRs for In-vitro Type 1 Dataset Using Deylami's Method[67]. Deylami's Method Only Has Results for Single Firing, and They are not as good as Our Method.

method to be both effective and robust. Overall, DAS is still the most robust and cost effective method for beamforming. Other methods can provide better performance in resolution and contrast, but sometimes introduce distortion or harm the speckle quality. More insights on how these new methods compare with DAS will be available once the results from stage 2 are compiled by the organizers in a future journal paper. My committee chair and I have been invited to be co-authors.

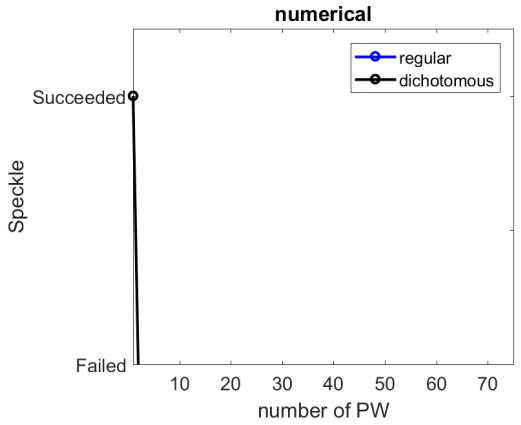


(a) Our method

(b) DAS



(c) Szasz's method



(d) Deylami's method

Figure 6.18: Results of Speckle Quality Test for Numerical Dataset. (a) Our Method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. All Methods Passed the Speckle Quality Check except for Szasz's Method.

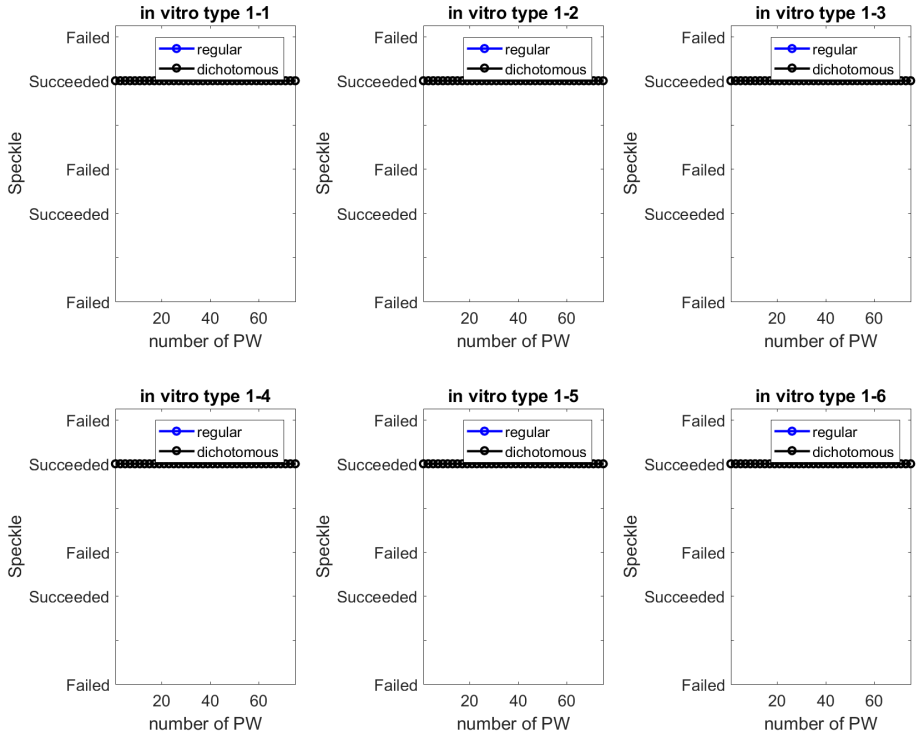


Figure 6.19: Results of Speckle Quality Test for In-vitro Type 1 Dataset Using Our Method. Our Method Passed the Speckle Quality Check All the Time, while Szasz's Method Failed Occasionally.

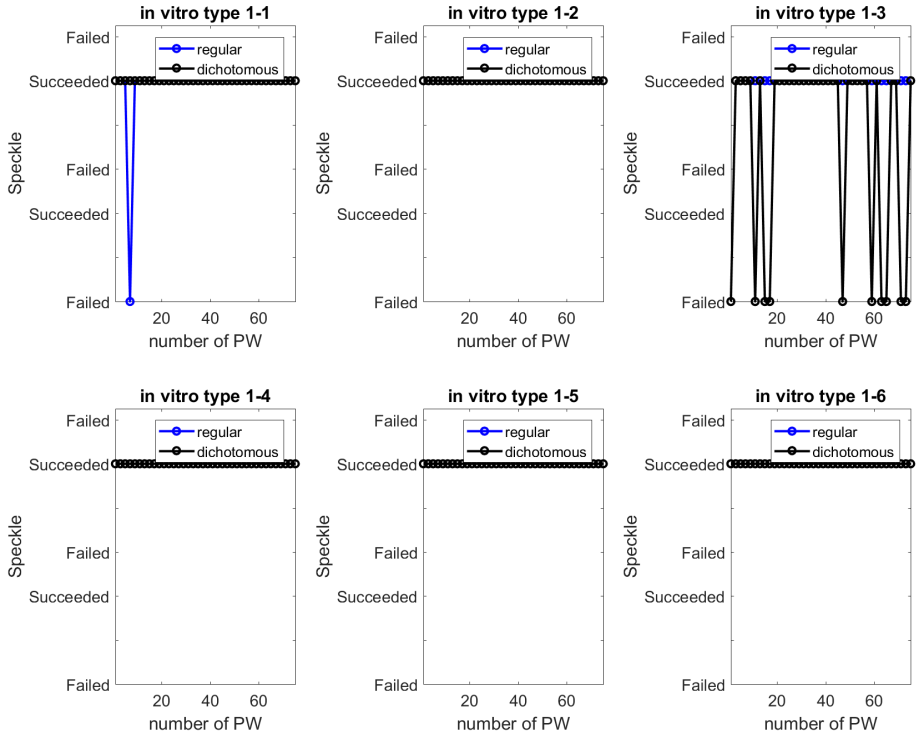
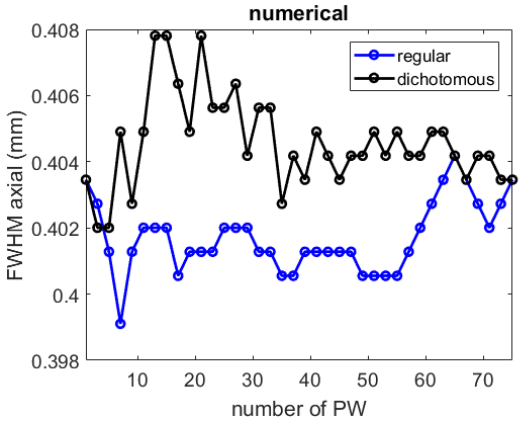
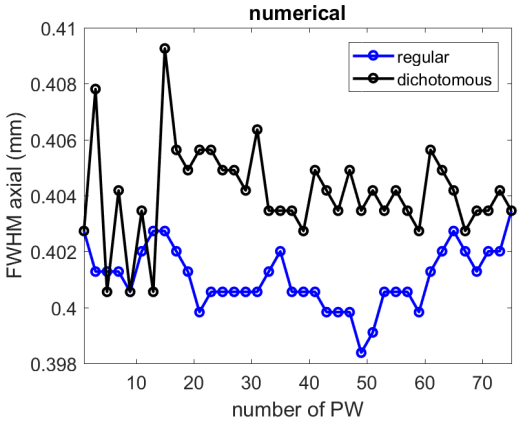


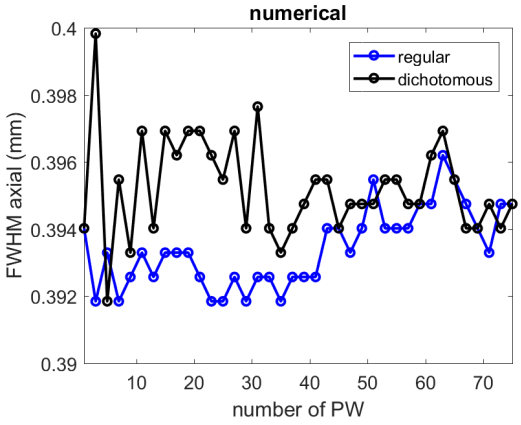
Figure 6.20: Results of Speckle Quality Test for In-vitro Type 1 Dataset Using Szasz's Method[66]. Our Method Passed the Speckle Quality Check All the Time, while Szasz's Method Failed Occasionally.



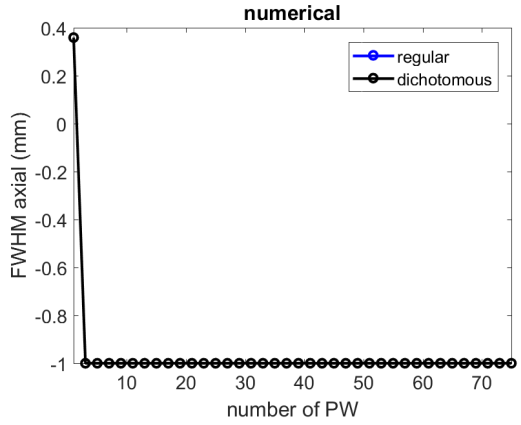
(a) Our method



(b) DAS

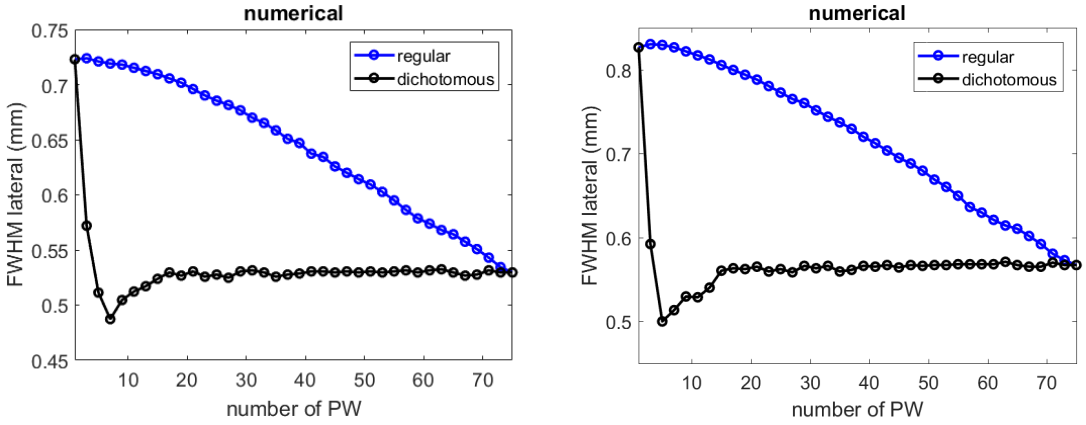


(c) Szasz's method



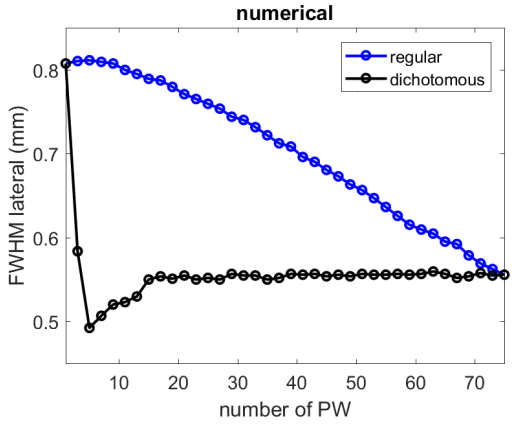
(d) Deylami's method

Figure 6.21: FWHM Results in Axial Dimension for Numerical Dataset. (a) Our Method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. Our Method is Comparable to DAS. Both Szasz's and Deylami's Methods Provide Slightly Better Results in Axial FWHMs.

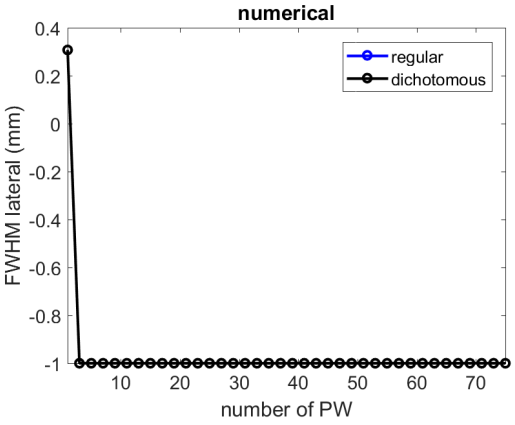


(a) Our method

(b) DAS

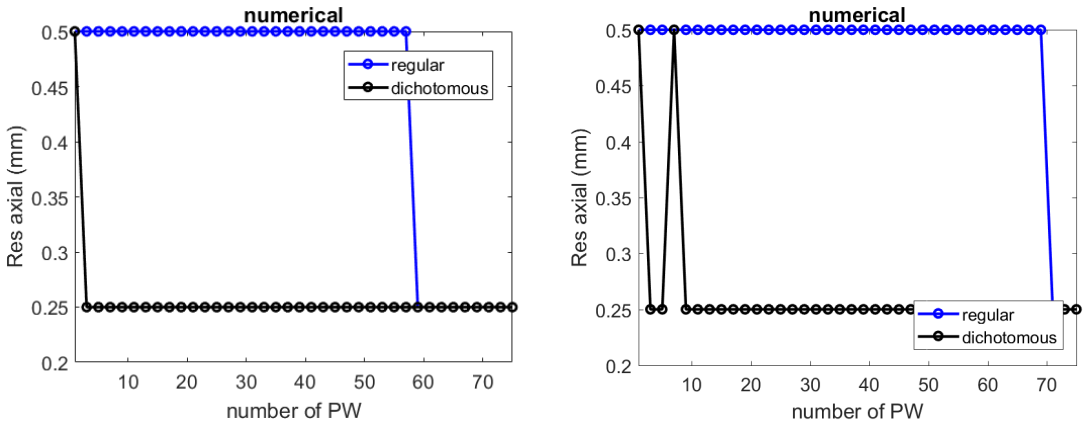


(c) Szasz's method



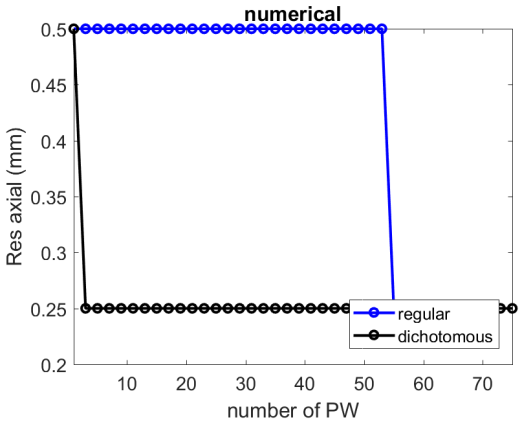
(d) Deylami's method

Figure 6.22: FWHM Results in Lateral Dimension for Numerical Dataset. (a) Our Method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. Our Method is Better Than Both DAS and Szasz's Method for Lateral FWHM, Deylami's Method is the Best for the Single Firing Case.

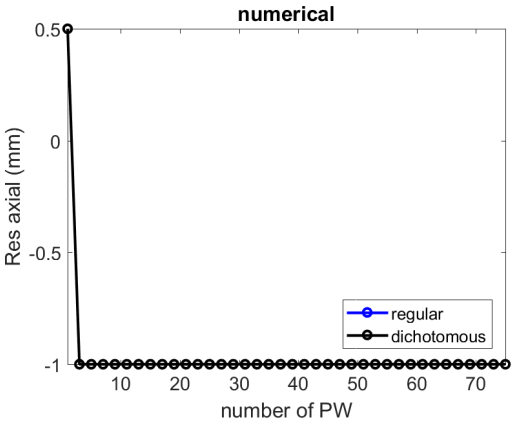


(a) Our method

(b) DAS

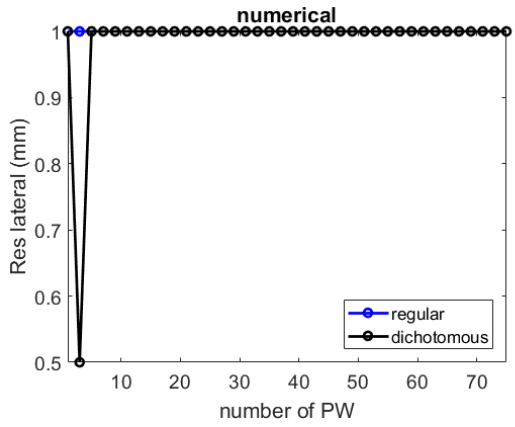


(c) Szasz's method

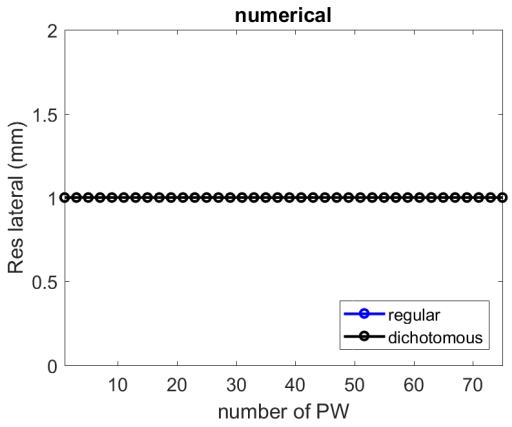


(d) Deylami's method

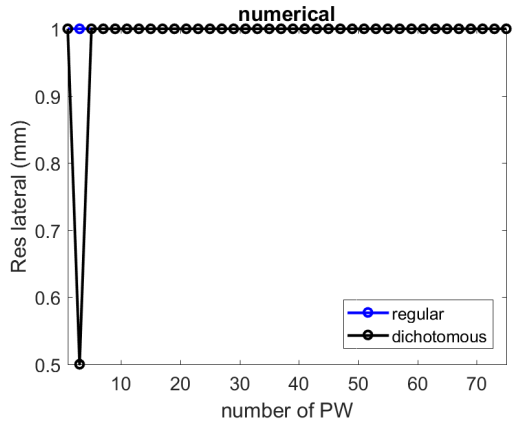
Figure 6.23: Resolution Results in Axial Dimension for Numerical Dataset. (a) Our Method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. All Methods Have Comparable Results for Axial Resolution.



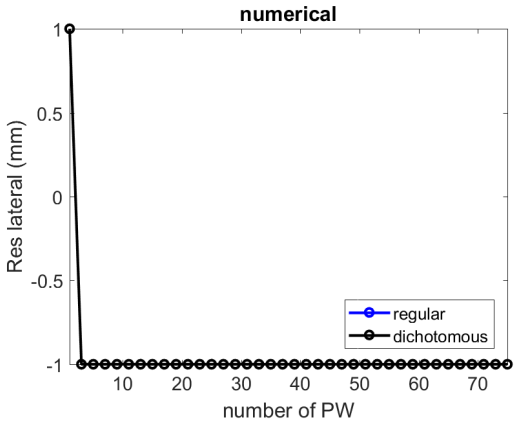
(a) Our method



(b) DAS



(c) Szasz's method



(d) Deylami's method

Figure 6.24: Resolution Results in Lateral Dimension for Numerical Dataset. (a) Our Method, (b) DAS, (c) Szasz's Method[66] and (d) Deylami's Method[67]. All Methods Have Comparable Results for Lateral Resolution.

Chapter 7

CONCLUSION

In this work, we developed a low-complexity flow rate estimation system for plane-wave imaging systems. It is based on a low-cost two-tiered 3D velocity estimation scheme with sum-of-absolute difference (SAD) for coarse-grained search and synthetic lateral phase for fine tuning the estimation results. Additional features include kernel power weighting for accurate flow estimation and low complexity power iteration based approach for clutter filtering.

1. Low cost blood velocity estimator with sub-pixel accuracy

We proposed a two-tiered scheme to combine the low complexity SAD estimator with a 3D version of the synthetic lateral phase method (SLP3D) [41] to provide accurate blood velocity estimation with sub-pixel accuracy. We further improved the sub-pixel accuracy by correcting phase of cross correlation based on autocorrelation. The proposed method achieved a 9× reduction in terms of normalized computation, compared to a direct implementation of SLP3D [42]. For the case when the blood did not have clutter, we achieved excellent velocity estimation accuracy for both plug flow and parabolic flow models. For plug flow, the average estimation bias was within 2% and the average standard deviation is within 10%. For parabolic flow, when the beam-to-angle was 90^o, the average estimation bias was within 7% and average standard deviation was within 8%.

2. Low cost eigen-based clutter filter

Clutter signal has to be removed before any velocity estimation can be done due to its magnitude being much higher than that of the blood signal. We proposed a low-cost eigen-based clutter filter method [45], which can remove one or more subspaces

and thus can handle both slow moving clutter where the clutter velocity is 0.5% of the peak blood velocity and fast moving clutter where the clutter velocity is 10% of the peak blood velocity. We validated our clutter filter performance with clutter-to-blood ratio (CBR) simulations. The results showed that the power/subspace iteration based clutter filter has comparable performance with the full SVD based clutter filter with significantly reduced complexity.

3. Flow rate estimation for parabolic flow

For accurate flow rate estimation of blood with clutter, we proposed a weighting of kernel power that assigns fractional weights to kernels that are not fully inside the vessel, thereby improving overall flow rate estimates. We validated our techniques through Field-II [44] simulations for parabolic flow with both slow and fast moving clutter and for 60^o and 90^o beam-to-flow angles. We show that our blood velocity estimation scheme is quite accurate, with less than 8% average bias for both slow and fast moving clutter. The average standard deviation of the eatimation is smaller for 90^o scenario ($< 12\%$) than that of the 60^o scenario ($< 16\%$). Volumetric flow rate estimation is also quite accurate. For a beam-to-flow angle of 90^o, the bias is 8.2%, and the standard deviation is 5.6% for slow moving clutter; the bias is 8.8% and the standard deviation is 3.1% for fast moving clutter. The complexity of our flow rate estimation system is only about 2.2 times higher than the baseline plane-wave beamforming system.

4. Flow estimation using challenging models

We tested our flow estimation method on challenging models such as spinning disk and carotid bifurcation. The spinning disk and carotid bifurcation were part of the synthetic aperture vector flow imaging (SA-VFI) challenge in IUS18. Since these two models were in 2D, we used a 2D version of our two-tiered method with techniques to account for aliasing. Estimation results for spinning disk showed that

our method is capable of accurately estimating velocity vectors over all angles, with average magnitude bias less than 10% and average angle bias less than 10^o. Estimation results for carotid bifurcation demonstrated that our method is able to accurately estimate velocity vectors with average magnitude bias of 9% and average angle bias of 8^o. In addition, we also tested our method on vessels with stenosis by using velocity profiles generated from COMSOL. The velocity estimation was quite accurate with low bias and deviation except at the stenosis boundaries. The flow rate estimation was still accurate with lower than 6% bias in the central part of the imaging aperture.

5. Post processing for plane-wave imaging

Plane-wave imaging provides fast aquisition rate which is beneficial for 3D velocity estimation. We participated in the PICMUS challenge in IUS16 which aimed at evaluating different beamforming methods for plane wave imaging. We used delay-and-sum (DAS) beamforming and edge detection based techniques for post processing. Our method helped improve the contrast ratio of cysts without affecting other parts of the image. Compared to other methods in the challenge that were not based on DAS, our method was superior in terms of contrast ratio of cysts but had worse resolution on point spread functions. In addition, we passed all distortion checks, such as speckle quality check, while others failed occasionally. Our results will be included in the paper that is being put together by the organizers.

Future work

While the proposed velocity estimator has superior performance for plug/parabolic flow, its performance in "real" flows has yet to be studied. We assumed straight vessels with a perfect cylinder shape, for plug/parabolic flow. In practice, the vessel shape is never regular–the radius changes and bifurcation exists. We also assumed constant flow rate while the blood flow is pulsatile in nature. Theoretically, the pulsatility is not an issue as long as the pulse repetition frequency (PRF) is high and the packet

size is reasonable so that the velocity changes within a packet is bounded. However, more validation is needed and an optimal packet size needs to be chosen.

During the course of this work, we did not have access to in-vitro and in-vivo data and so could not evaluate our method on real data. However, through the SA-VFI challenge which was initiated this year by the IUS18 conference organizers, we had access to both in-vitro data and CFD simulated data that are closer to practical scenarios. Our method provided superior results on all datasets.

Finally, although only performance is compared in the challenge, computational complexity matters in practice. So further reduction in the complexity of our method without compromising on the performance would help pave the way to the proposed flow estimation system being used in real ultrasound systems.

REFERENCES

[1] World Health Organization and World Health Organization. Management of Substance Abuse Unit, *Global status report on alcohol and health, 2014*, World Health Organization, 2014.

[2] S. Holbek, C. Ewertsen, H. Bouzari, M. J. Pihl, K. L. Hansen, M. B. Stuart, C. Thomsen, M. B. Nielsen, and J. A. Jensen, "Ultrasonic 3-D vector flow method for quantitative in vivo peak velocity and flow rate estimation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, no. 3, pp. 544–554, March 2017.

[3] K. L. Hansen, J. Udesen, C. Thomsen, J. A. Jensen, and M. B. Nielsen, "In vivo validation of a blood vector velocity estimator with MR angiography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 1, pp. 91–100, January 2009.

[4] A Harloff, T Zech, F Wegent, C Strecker, C Weiller, and M Markl, "Comparison of blood flow velocity quantification by 4D flow MR imaging with ultrasound at the carotid bifurcation," *American Journal of Neuroradiology*, vol. 34, no. 7, pp. 1407–1413, 2013.

[5] J.A. Jensen, S.I. Nikolov, J. Udesen, P. Munk, K.L. Hansen, M.M. Pedersen, P.M. Hansen, M.B. Nielsen, N. Oddershede, J. Kortbek, M.J. Pihl, and Ye Li, "Recent advances in blood flow vector velocity imaging," in *Proc. of IEEE Int. Ultrasonics Symposium (IUS), 2011*, Oct 2011, pp. 262–271.

[6] B. Dunmire, K.W. Beach, K-H Labs, M. Plett, and D.E. Strandness Jr., "Cross-beam vector Doppler ultrasound for angle-independent velocity measurements," *Ultrasound in Medicine and Biology*, vol. 26, no. 8, pp. 1213 – 1235, 2000.

[7] J. A. Jensen, "A new estimator for vector velocity estimation [medical ultrasonics]," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 4, pp. 886–894, July 2001.

[8] M. J. Pihl, J. Marcher, and J. A. Jensen, "Phased-array vector velocity estimation using transverse oscillations," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 12, Dec 2012.

[9] J. Udesen, F. Gran, K. L. Hansen, J. A. Jensen, C. Thomsen, and M. B. Nielsen, "High frame-rate blood vector velocity imaging using plane waves: Simulations and preliminary experiments," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 8, pp. 1729–1743, August 2008.

[10] O. D. Kripfgans, J. M. Rubin, A. L. Hall, M. B. Gordon, and J. B. Fowlkes, "Measurement of volumetric flow," *Journal of Ultrasound in Medicine*, vol. 25, no. 10, pp. 1305–1311, 2006.

[11] M.J. Pihl and J.A. Jensen, "3D vector velocity estimation using a 2d phased array," in *Proc. of IEEE Int. Ultrasonics Symposium (IUS), 2011*, Oct 2011, pp. 430–433.

[12] M.J. Pihl and J.A. Jensen, "Measuring 3D velocity vectors using the transverse oscillation method," in *Proc. of IEEE Int. Ultrasonics Symposium (IUS), 2012*, Oct 2012, pp. 1881–1885.

[13] J. A. Jensen, H. Holten-Lund, R. T. Nielson, B. G. Tomov, M. B. Stuart, S. I. Nikolov, M. Hansen, and U. D. Larsen, "Performance of SARUS: A synthetic aperture real-time ultrasound system," in *Proceedings of IEEE Ultrasonics Symposium*, 2010, pp. 305–309.

[14] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wenisch, "Sonic Millip3De: Massively parallel 3D stacked accelerator for 3D ultrasound," in *19th IEEE International Symposium on High Performance Computer Architecture*, Feb. 2013, pp. 318–329.

[15] M. Yang, R. Sampson, T. F. Wenisch, and C. Chakrabarti, "Separable beamforming for 3-D synthetic aperture ultrasound imaging," in *Proceedings of IEEE Workshop on Signal Processing Systems*, Taipei, Taiwan, Oct. 2013, pp. 207–212.

[16] M. Yang, R. Sampson, S. Wei, T. F. Wenisch, and C. Chakrabarti, "High frame rate 3-D ultrasound imaging using separable beamforming," *Journal of Signal Processing Systems*, vol. 78, no. 1, pp. 73–84, 2015.

[17] J. A. Jensen, S. Nikolov, A. C. H. Yu, and D. Garcia, "Ultrasound vector flow imaging part i: Sequential systems," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1704–1721, Nov 2016.

[18] J. A. Jensen, S. Nikolov, A. C. H. Yu, and D. Garcia, "Ultrasound vector flow imaging part ii: Parallel systems," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1722–1732, Nov 2016.

[19] J. Norris, C. Zhu, N. Bornstein, and B. Chambers, "Vascular risks of asymptomatic carotid stenosis.," *Stroke*, vol. 22, no. 12, pp. 1485–1490, 1991.

[20] R.S.C. Cobbold, *Foundations of Biomedical Ultrasound*, Oxford University Press, USA, 2007.

[21] B. Hussain, B. Y. S. Yiu, A. C. H. Yu, J. C. Lacefield, and T. L. Poepping, "Investigation of crossbeam multi-receiver configurations for accurate 3-D vector Doppler velocity estimation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1786–1798, Nov 2016.

[22] M. J. Pihl and J. A. Jensen, "A transverse oscillation approach for estimation of three-dimensional velocity vectors, part i: concept and simulation study," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 10, pp. 1599–1607, Oct 2014.

[23] M. Lenge, A. Ramalli, P. Tortoli, C. Cachard, and H. Liebgott, "Plane-wave transverse oscillation for high-frame-rate 2-D vector flow imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, no. 12, pp. 2126–2137, December 2015.

[24] J. A. Jensen, H. Holten-Lund, R. T. Nilsson, M. Hansen, U. D. Larsen, R. P. Domsten, B. G. Tomov, M. B. Stuart, S. I. Nikolov, M. J. Pihl, Y. Du, J. H. Rasmussen, and M. F. Rasmussen, "SARUS: A synthetic aperture real-time ultrasound system," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 60, no. 9, pp. 1838–1852, 2013.

[25] J. A. Jensen, "Directional velocity estimation using focusing along the flow direction. i: theory and simulation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 50, no. 7, pp. 857–872, July 2003.

[26] F. Viola and W.F. Walker, "A comparison of the performance of time-delay estimators in medical ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 50, no. 4, pp. 392–401, April 2003.

[27] S. Fadnes, I. K. Ekroll, S. A. Nyrnes, H. Torp, and L. Lovstakken, "Robust angle-independent blood velocity estimation based on dual-angle plane wave imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, no. 10, pp. 1757–1767, October 2015.

[28] J. Van Cauwenberge, L. Lovstakken, S. Fadnes, A. Rodriguez-Morales, J. Vierendeels, P. Segers, and A. Swillens, "Assessing the performance of ultrafast vector flow imaging in the neonatal heart via multiphysics modeling and in vitro experiments," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1772–1785, Nov 2016.

[29] A. E. C. M. Saris, H. H. G. Hansen, S. Fekkes, M. M. Nillesen, M. C. M. Rutten, and C. L. de Korte, "A comparison between compounding techniques using large beam-steered plane wave imaging for blood vector velocity imaging in a carotid artery model," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1758–1771, Nov 2016.

[30] M. Tanter and M. Fink, "Ultrafast imaging in biomedical ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 1, pp. 102–119, January 2014.

[31] J. Provost, C. Papadacci, J. Arango, M. Imbault, M. Fink, J. Gennisson, M. Tanter, and M. Pernot, "3D ultrafast ultrasound imaging in vivo," *Physics in medicine and biology*, vol. 59, no. 19, pp. L1, 2014.

[32] C. Poelma, J. Mari, N. Foin, M. Tang, R. Krams, C. Caro, P. Weinberg, and J. Westerweel, "3D flow reconstruction using ultrasound PIV," *Experiments in fluids*, vol. 50, no. 4, pp. 777–785, 2011.

[33] C. Poelma, R. Van der Mijle, J. Mari, M. Tang, P. Weinberg, and J. Westerweel, "Ultrasound imaging velocimetry: Toward reliable wall shear stress measurements," *European Journal of Mechanics-B/Fluids*, vol. 35, pp. 70–75, 2012.

[34] C. Tremblay-Darveau, R. Williams, P. S. Sheeran, L. Milot, M. Bruce, and P. N. Burns, "Concepts and tradeoffs in velocity estimation with plane-wave contrast-enhanced Doppler," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1890–1905, Nov 2016.

[35] J. M. Mari, "Study of Fourier-based velocimetry," *Open J. Acoust.*, vol. 3, no. 3A, pp. 21–32, 2013.

[36] L. Thomas and A. Hall, "An improved wall filter for flow imaging of low velocity flow," in *Ultrasonics Symposium, 1994. Proceedings., 1994 IEEE*, Oct 1994, vol. 3, pp. 1701–1704 vol.3.

[37] A.C.H. Yu and L. Lovstakken, "Eigen-based clutter filter design for ultrasound color flow imaging: a review," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 5, pp. 1096–1111, May 2010.

[38] S. Bjaerum, H. Torp, and K. Kristoffersen, "Clutter filters adapted to tissue motion in ultrasound color flow imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 49, no. 6, pp. 693–704, June 2002.

[39] A.C.H. Yu and R.S.C. Cobbold, "Single-ensemble-based eigen-processing methods for color flow imaging - part i. the Hankel-svd filter," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 3, pp. 559–572, March 2008.

[40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, USA, 1989.

[41] X. Chen, M.J. Zohdy, S.Y. Emelianov, and M. O'Donnell, "Lateral speckle tracking using synthetic lateral phase," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 51, no. 5, pp. 540–550, May 2004.

[42] S. Wei, M. Yang, C. Chakrabarti, R. Sampson, T.F. Wenisch, O. Kripfgans, and J.B. Fowlkes, "A low complexity scheme for accurate 3D velocity estimation in ultrasound systems," *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 85–90, Oct 2014.

[43] S. Wei, M. Yang, J. Zhou, R. Sampson, O. D. Kripfgans, J. B. Fowlkes, T. F. Wenisch, and C. Chakrabarti, "Low-cost 3-D flow estimation of blood with clutter," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, no. 5, pp. 772–784, May 2017.

[44] J. A. Jensen, "FIELD: A program for simulating ultrasound systems," *10th Nordicbaltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351–353*, pp. 351–353, 1996.

[45] S. Wei, M. Yang, R. Sampson, O.D. Kripfgans, J.B. Fowlkes, T.F. Wenisch, and C. Chakrabarti, "Low cost clutter filter for 3D ultrasonic flow estimation," in *Signal Processing Systems (SiPS), 2015 IEEE Workshop on*, Oct 2015, pp. 1–6.

[46] M. J. Pihl, M. B. Stuart, B. G. Tomov, M. F. Rasmussen, and J. A. Jensen, "A transverse oscillation approach for estimation of three-dimensional velocity vectors, part ii: experimental validation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 10, pp. 1608–1618, Oct 2014.

[47] L.N. Bohs, B.J. Geiman, M.E. Anderson, S.C. Gebhart, and G.E. Trahey, "Speckle tracking for multi-dimensional flow estimation," *Ultrasonics*, vol. 38, no. 18, pp. 369 – 375, 2000.

[48] X. Chen, H. Xie, R. Erkamp, K. Kim, C. Jia, J. M. Rubin, and M. O'Donnell, "3-D correlation-based speckle tracking," *Ultrasonic Imaging*, vol. 27, no. 1, pp. 21–36, 2005.

[49] A. Swillens, P. Segers, H. Torp, and L. Lovstakken, "Two-dimensional blood velocity estimation with ultrasound: speckle tracking versus crossed-beam vector Doppler based on flow simulations in a carotid bifurcation model," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 57, no. 2, pp. 327–339, February 2010.

[50] R.Z. Azar, O. Goksel, and S.E. Salcudean, "Sub-sample displacement estimation from digitized ultrasound rf signals using multi-dimensional polynomial fitting of the cross-correlation function," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 57, no. 11, pp. 2403–2420, November 2010.

[51] J. A. Jensen and N. B. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 39, no. 2, pp. 262 –267, Mar. 1992.

[52] K. Chang, C. Zhan, and A. Wu, "Joint-decision adaptive clutter filter and motion-tracking adaptive persistence for color Doppler processing in ultrasonic systems," in *2010 IEEE Workshop on Signal Processing Systems (SIPS)*, Oct 2010, pp. 249–253.

[53] S. Fadnes, S. Bjrum, H. Torp, and L. Lovstakken, "Clutter filtering influence on blood velocity estimation using speckle tracking," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 62, no. 12, pp. 2079–2091, December 2015.

[54] J. Lee, S. Ryu, Y. Chang, K. Hsu, Y. Chen, Y. Huang, M. Lee, M. Hsiao, and T. Lee, "Carotid ultrasound criteria for detecting intracranial carotid stenosis," *European neurology*, vol. 57, no. 3, pp. 156–160, 2007.

[55] N. Ackroyd, R. Gill, K. Griffiths, G. Kossoff, and M. Appleberg, "Quantitative common carotid artery blood flow: prediction of internal carotid artery stenosis," *Journal of vascular surgery*, vol. 3, no. 6, pp. 846–853, 1986.

[56] M. Yang, R. Sampson, S. Wei, T. F. Wenisch, and C. Chakrabarti, "Separable beamforming for 3-D medical ultrasound imaging," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 279–290, Jan 2015.

[57] "International Ultrasonics Symposia Synthetic Aperture Flow Imaging Challenge," `http://sites.ieee.org/ius-2018/files/2018/01/2018_IEEE_IUS_SA-VFI_Challenge.pdf`.

[58] A. Swillens, P. Segers, H. Torp, and L. Lovstakken, "Two-dimensional blood velocity estimation with ultrasound: speckle tracking versus crossed-beam vector Doppler based on flow simulations in a carotid bifurcation model," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 2, pp. 327–339, February 2010.

[59] A. King, M. Shipley, and H. Markus, "The effect of medical treatments on stroke risk in asymptomatic carotid stenosis," *Stroke*, vol. 44, no. 2, pp. 542–546, 2013.

[60] COMSOL Inc., "Comsol multiphysics, version 5.2," `http://www.comsol.com`.

[61] A. Swillens, L. Lvstakken, J. Kips, H. Torp, and P. Segers, "Ultrasound simulation of complex flow velocity fields based on computational fluid dynamics," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 3, pp. 546–556, March 2009.

[62] H. Liebgott, A. Rodriguez-Molares, F. Cervenansky, J. A. Jensen, and O. Bernard, "Plane-wave imaging challenge in medical ultrasound," in *2016 IEEE International Ultrasonics Symposium (IUS)*, Sept 2016, pp. 1–4.

[63] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: Algorithm and fpga implementation," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, July 2014.

[64] "Plane-wave Imaging evaluation framework for Medical Ultrasound," `https://www.creatis.insa-lyon.fr/EvaluationPlatform/picmus/index.html`.

[65] "Participants Results of the Plane-wave Imaging evaluation framework for Medical Ultrasound," `https://www.creatis.insa-lyon.fr/~bernard/picmus_results/`.

[66] T. Szasz, A. Basarab, and D. Kouam, "Beamforming of ultrasound images modelled as stable random variables," in *2016 IEEE International Ultrasonics Symposium (IUS)*, Sept 2016, pp. 1–4.

[67] A. M. Deylami, J. A. Jensen, and B. M. Asl, "An improved minimum variance beamforming applied to plane-wave imaging in medical ultrasound," in *2016 IEEE International Ultrasonics Symposium (IUS)*, Sept 2016, pp. 1–4.