

High-Frame-Rate Ultrasound Imaging Innovations for Complex Flow Quantification

by

Hassan Nahas

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

©Hassan Nahas 2021

EXAMINING COMMITTEE MEMBERSHIP

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	QIFA ZHOU Professor, Department of Biomedical Engineering, University of Southern California
Supervisor(s)	ALFRED C. H. YU Professor, Department of Electrical and Computer Engineering, University of Waterloo
Internal Member	DAYAN BAN Professor, Department of Electrical and Computer Engineering, University of Waterloo
Internal Member	GEORGE FREEMAN Associate Professor, Department of Electrical and Computer Engineering, University of Waterloo
Internal-external Member	ZHAO PAN Assistant Professor, Department of Mechanical and Mechatronics Engineering, University of Waterloo

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

STATEMENT OF CONTRIBUTIONS

The author contributions for each Chapter are described below.

Chapters 1, 2 and 7: Hassan Nahas wrote these chapters.

Chapter 3 presents material from an article entitled “A Deep Learning Approach to Resolve Aliasing Artifacts in Ultrasound Color Flow Imaging” which has been published in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. This article was co-authored by Hassan Nahas; Jason S. Au; Takuro Ishii; Billy Y. S. Yiu; Adrian J. Y. Chee; Alfred C. H. Yu. H. Nahas carried out all the algorithmic work, deep learning development and evaluation, analyzed the results, and wrote the manuscript draft. J. S. Au supplied the *in vivo* data for training and validation. T. Ishii, B. Y. S. Yiu and A. J. Y. Chee provided input in editing the manuscript. A. C. H. Yu reviewed the results and edited the manuscript prior to publication.

A version of **Chapter 4**, titled “A GPU-Based, Real-Time Dealiasing Framework for High-Frame-Rate Vector Doppler Imaging” has been submitted to the IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, co-authored by Hassan Nahas, Takuro Ishii, Billy Y. S. Yiu, and Alfred C. H. Yu. Hassan Nahas carried out the algorithmic development, performed the validation, analyzed the results and wrote the manuscript draft. B. Y. S. Yiu supplied the phantom data. T. Ishii provided input in editing the manuscript. A. C. H. Yu reviewed the results and edited the manuscript prior to submission.

A version of **Chapter 5**, titled “Fast Aliasing Resistant Vector Doppler Imaging using Deep Learning and GPU Computing” is being prepared for submission to the IEEE Transactions on Medical Imaging, co-authored by Hassan Nahas, Jason S. Au, Billy Y. S. Yiu, Adrian J. Y. Chee, and Alfred C. H. Yu. H. Nahas carried out the algorithmic and deep learning development, performed the validation, analyzed the results, and wrote the manuscript draft. J. S. Au provided the *in vivo* data for training and evaluation. B. Y. S. Yiu, J. S. Au and A. J. Y. Chee provided input in editing the manuscript. All authors will review the results and manuscript prior to submission.

A version of **Chapter 6**, titled “Aliasing Resistant Vector Doppler at the Bedside using GPU Processing and Deep Learning” is being prepared for submission to the BME Frontiers, co-authored by Hassan Nahas, Billy Y. S. Yiu, Adrian J. Y. Chee, Takuro Ishii, and Alfred C. H. Yu. Hassan Nahas carried out the software development, performed the validation experiments and demos, analyzed the results, and wrote the manuscript draft. The software was developed as branch of the

flow imaging software previously produced by B. Y. S. Yiu and utilizes his GPU kernels for beamforming, Doppler estimation and image processing. B. Y. S. Yiu and A. J. Y. Chee provided input in editing the manuscript. All authors will review the results and manuscript prior to submission.

Abstract

Vascular pathology such as atherosclerosis or arterial stiffening involves significant disruption in the blood flow dynamics found in the vasculature, with complex behavior such as jets and swirling typically seen in diseased vessels. Detecting and mapping complex blood flow dynamics in the human vasculature is therefore important for monitoring disease progression and understanding the pathology. Existing solutions for flow imaging in the clinic cannot reliably capture complex blood flow dynamics. These solutions also suffer from limited frame rates, high operational costs and inaccessibility for point-of-care and routine use.

To meet this need, this dissertation has devised a novel ultrasound-based imaging framework suited for visualizing and quantifying complex blood flow in human vasculature at the bedside. This was achieved using next-generation ultrasound technology and a novel vector flow imaging framework. The framework utilized deep learning and parallel computing principles to achieve real-time vector flow imaging with an expanded range of measurable velocities. The proposed framework was realized on a programmable and mobile ultrasound scanner to support point-of-care use. Performance of the proposed framework was validated in phantom and human experiments, including healthy and diseased arteries. Reliable vector flow imaging was achieved at the bedside in human trials where complex and high flow speed blood flow was imaged at the bedside.

This dissertation presents important progress towards the visualization of complex blood flow dynamics in human vasculature at the bedside and the translation of advanced ultrasound imaging technologies to the clinic. By providing a reliable implementation of vector flow imaging at the bedside, the presented framework should facilitate more comprehensive and large-scale human studies into the hemodynamics of healthy and pathological vasculature. This work also showcases the novel applications of deep learning and parallel computing principles to ultrasound flow imaging, thereby contributing new strategies to the field.

Acknowledgements

I would first like to thank Prof. Alfred Yu for the opportunity to pursue a PhD under his supervision. I am grateful for his support and mentorship throughout this journey and for the high standards he championed.

I would like to thank Prof. George Freeman for his advice and support towards my research and teaching development. Under his stewardship, I have learned important lessons on education and science communication.

I would like to thank Prof. Takuro Ishii for his guidance and mentorship in the formative years of my graduate studies and for inspiring me with his dedication and work ethic. I would also like to thank Dr. Adrian Chee, Dr. Billy Yiu, Dr. Xinxin Duan and Prof. Jason Au for their mentorship, support, and patience. They have played an instrumental role in the development of my research and communication abilities. I am grateful for all my lab colleagues for creating a wonderful environment and inspiring me with their creativity. I cherish the experiences we have had together, the friendships we have built and the banter we have shared.

I would like to thank my parents and my brothers for their unwavering love, support and endless motivation. I am grateful for their guidance, their inspiration and the values they have instilled in me. They have provided me much needed virtual company throughout the otherwise lonely days of a long pandemic. I would also like to thank my family and friends in Canada for providing me a home away from home, and my family in Syria for their continuous emotional support throughout this journey.

Table of Contents

EXAMINING COMMITTEE MEMBERSHIP	ii
AUTHOR'S DECLARATION.....	iii
STATEMENT OF CONTRIBUTIONS.....	iv
Abstract.....	vi
Acknowledgements.....	vii
List of Figures	xiii
List of Tables	xviii
List of Abbreviations	xix
List of Symbols	xxii
Chapter 1 Introduction	26
1.1 Motivation.....	26
1.2 Objectives	26
1.3 Hypothesized Solution	27
1.4 Thesis Outline	28
Chapter 2 Ultrasound Flow Imaging, State-of-the-Art and Limitations	29
2.1 Ultrasound Imaging as a Tool for <i>In Vivo</i> Blood Flow Estimation	29
2.1.1 Motion Estimation in Current Ultrasound Scanners.	30
2.2 Recent Motion Estimation Techniques	34
2.2.1 Vector Flow Estimation	34
2.2.2 High Frame Rate Ultrasound Imaging	37
2.3 Aliasing in Doppler Ultrasound Flow Imaging.....	39
2.3.1 The Principles of Doppler Aliasing.....	39
2.3.2 Aliasing in Color Flow Imaging	40
2.3.3 Aliasing in Vector Doppler Imaging.....	41
2.3.4 Extended Least Squares Vector Doppler	43
2.4 Computing and Hardware Innovations in Ultrasound Imaging	44
2.5 Deep Learning Applications in Ultrasound Imaging	45
2.6 Potential Application for the Framework: Vascular Stenosis	46
2.6.1 Morphological Features	47

2.6.2 Hemodynamic Features	47
2.7 Concluding Remarks	48
Chapter 3 A Deep Learning Approach to Resolve Aliasing Artifacts in Ultrasound Color Flow Imaging.....	49
3.1 Introduction	49
3.2 Theoretical Principles.....	51
3.2.1 Overall Framework for Aliasing Segmentation.....	51
3.2.2 Training Features for Aliasing Identification	52
3.2.3 Candidate Combinations for Training Features.....	54
3.2.4 CNN Architecture.....	55
3.2.5 Dealiasing and Image Regularization.....	56
3.3 Experimental Methods.....	58
3.3.1 Data Collection.....	58
3.3.2 Signal Processing.....	59
3.3.3 Data Organization for CNN Training	60
3.3.4 Training Implementation	62
3.3.5 Testing and Segmentation Evaluation	62
3.3.6 Dealiasing Performance Evaluation	63
3.4 Results	64
3.4.1 CNNs Trained with Frequency, Power, and Bandwidth Features Yielded the Best Aliasing Detection Performance	64
3.4.2 Aliasing-Detection CNN Can Accurately Identify Most of the Aliased Pixels	65
3.4.3 Visualization of Dealiased CFI Frames After Adaptive Phase Unwrapping.....	66
3.5 Discussion	68
3.5.1 Summary of Contributions	68
3.5.2 Insights on CNN Training for Aliasing Detection.....	69
3.5.3 Applications of Interest	70
3.5.4 Limitations of Investigation	71
3.6 Conclusion.....	71
Chapter 4 A GPU-Based, Real-Time Dealiasing Framework for Vector Doppler Imaging	73
4.1 Introduction	73
4.2 Technical Principles	75

4.2.1 Key Parameters of Interest.....	75
4.2.2 Overall Parallelization Strategy	76
4.2.3 Stage One: Parallelized Derivation of Pen-Optimal Aliasing Correction Vector.....	77
4.2.4 Stage Two: Parallelized Computation of Optimal Dealiased Velocity Vector	80
4.3 Implementation of Real-Time ELS-VD.....	82
4.3.1 General Considerations for Fast Execution.....	82
4.3.2 Fast Memory Access Based on the Use of One Thread Block Per Pixel.....	83
4.3.3 Optimization of the CalculateResidues Kernel	83
4.3.4 Optimization of the FindBestP Kernel.....	84
4.3.5 Optimization of the BlockMatching Kernel.....	84
4.4 Experimental Methods	85
4.4.1 Experimental Hardware	85
4.4.2 Dealiasing Performance Analysis	86
4.4.3 Throughput Analysis.....	87
4.5 Results.....	87
4.5.1 Real-Time Aliasing-Resistant Flow Vector Mapping Achieved in a Stenosed Carotid Bifurcation	87
4.5.2 ELS-VD Processing Throughput Influenced by Multiple Parameters.....	90
4.5.3 Block Matching is the Primary Computational Bottleneck.	90
4.6 Discussion.....	92
4.6.1 Summary of Contributions.....	92
4.6.2 Insights on GPU Acceleration of ELS-VD	94
4.7 Conclusion	95
Chapter 5 Fast Aliasing Resistant Vector Doppler Imaging Using Deep Learning and GPU Computing	96
5.1 Introduction.....	96
5.2 Theoretical Principles	98
5.2.1 System and Problem Definition.	99
5.2.2 Module 1: Least Squares Vector Doppler Estimator	99
5.2.3 Module 2: Aliasing Segmentation Using Convolutional Neural Networks	100
5.2.4 Module 3: Aliasing Resistant Vector Doppler	102
5.3 Neural Network Development and Training.....	103

5.3.1 Dataset Collection	103
5.3.2 Dataset Organization	105
5.3.3 Training Hardware and Implementation.....	105
5.4 Processing Pipeline for Aliasing Resistant Vector Doppler Imaging.....	106
5.4.1 Processing Hardware and Environment	106
5.4.2 Image Formation and Doppler Processing	106
5.4.3 Aliasing Resistant Vector Flow Estimation.....	107
5.5 Experimental Validation.....	108
5.5.1 Evaluation of Aliasing Segmentation by the CNN.....	108
5.5.2 Aliasing Resistance and Processing Throughput Evaluation	108
5.6 Results	110
5.6.1 Evaluation of Aliasing Segmentation by the CNN.....	110
5.6.2 Aliasing-Free Vector Flow Maps Realized Using the Proposed Framework.....	112
5.6.3 Processing Throughput Increased in the Proposed Framework	114
5.7 Discussion	116
5.7.1 Summary of Contributions	116
5.7.2 A Framework Suited for Clinical Translation	117
5.8 Conclusion.....	117
Chapter 6 Aliasing Resistant Vector Doppler at the Bedside Using GPU Processing and Deep Learning.....	119
6.1 Introduction	119
6.2 Framework for Real-time Aliasing Resistant HiVFI.....	120
6.2.1 Hardware	122
6.2.2 Transmission Protocol.....	122
6.2.3 Software Implementation Overview.....	124
6.2.4 Live Mode Implementation	124
6.2.5 Replay Mode Implementation	127
6.3 Experimental Demonstration.....	131
6.3.1 Demonstration Protocol.....	131
6.3.2 Imaging Scenarios	131
6.3.3 Framework Output and Analysis.....	132
6.4 Results	133

6.4.1 Real-time Performance Achieved at the Bedside.....	133
6.4.2 Aliasing Resistant Vector Flow Improves Visualization of Multidirectional Flow.....	135
6.4.3 Aliasing Resistant Vector Flow is Possible in Excessive Aliasing Scenarios	137
6.5 Discussion.....	139
6.5.1 Summary of Contributions.....	139
6.5.2 Insights on Real-time Performance at the Bedside	140
6.5.3 <i>In Vivo</i> Applicability of the Proposed Framework	141
6.5.4 Perspectives for Further Validation and Development	142
6.6 Conclusion	142
Chapter 7 Discussion and Future Directions.....	144
7.1 Summary of Contributions.....	144
7.1.1 Aliasing Removal in Color Flow Visualization Using Deep Learning.....	146
7.1.2 Real-time Aliasing Resistant Vector Doppler Using GPU Computing	146
7.1.3 Robust Velocity Estimation with High Processing Throughput	147
7.1.4 Aliasing Resistant Vector Doppler Imaging at the Bedside.....	148
7.2 Limitations and Future Directions	149
7.2.1 Deep Learning Generalizability	149
7.2.2 Conditions of Extreme Aliasing.....	149
7.2.3 Expanding the Range of Application	150
Bibliography	151
Appendix.....	173

List of Figures

Figure 1.1. a) A neural network is trained to detect flow errors due to high velocities in ultrasound flow imaging. b) A robust velocity estimator that corrects flow errors is realized in a GPU-accelerated pipeline. c) An integrated estimator is devised to achieve robust velocity estimation with high processing throughput. d) A programmable scanner is configured for reliable error-free vector flow imaging. An output sample of the framework is shown on the carotid bifurcation showcasing B-mode for structure and vector flow map + speckle for flow dynamic visualization.....	28
Figure 2.1. a) Ultrasound imaging is done by placing the ultrasound transducer on the vessel being examined underneath the skin. b) Ultrasound pulses are sent periodically. After each pulse, the transducer receives the ultrasound reflections from any scatterers or boundaries. c) The received signal can be processed to form the B-mode image to indicate structure.....	29
Figure 2.2. As the scatterer moves away from the transducer, subsequent ultrasound pulses arrive more delayed. Cross-correlation of the received signals from two consecutive pulses can be used to estimate motion.....	31
Figure 2.3. a) B-mode of a carotid bifurcation model with a red cross denoting a region for motion estimation. b) The region of interest is insonified with multiple consecutive pulses Δt apart. The region of interest lies on a depth z_0 in this line. Following a series of transmissions, one can construct a ‘slow-time’ signal with a sampling rate of Δt . c) The short-time Fourier transform of the slow-time signal demonstrates the distribution of velocities across time. This spectrum, here windowed to a single cardiac cycle, clearly demonstrates the systolic and diastolic phases and two biomarkers of interest are highlighted: The peak systolic velocity and the spectral bandwidth. Another quantity of interest is the Nyquist limit of the system, which denotes the maximum speed that can be imaged.	32
Figure 2.4. CFI of a carotid bifurcation model, showing the speed of flow relative to the transducer which in this image would be vertically above the carotid. As such, fluid in the upper branch would be moving towards the transducer, hence the color red. In the lower branch, blood would be moving away from the transducer, hence the color blue.	33
Figure 2.5. Block matching can be used to track speckle motion. For each pixel of interest in the speckle image, a neighborhood of size $B \times B$ at frame t is matched in the next frame $t + \Delta t$. The new location of this neighborhood then corresponds to the displacement of the speckle patterns, and consequentially the velocity.....	35
Figure 2.6. a) Transmit-receive steering is used to insonify the location of interest from multiple directions. Doppler estimation can be applied for each steering angle to yield a Doppler frequency measurement per steering angle. These can then be combined to derive the full vector. b) If this is applied on the entire flow region, a vector flow map can be produced to indicate both flow speed and direction throughout the lumen.	36
Figure 2.7. Traditional ultrasound imaging is typically done using a) beamline emissions, where ultrasound is focused along a narrow beam, which raster scans the imaging view to produce high-quality images but at a low frame rate. Alternatively, broad waves can be used such as b) Diverging waves which spread out the further they propagate and c) Planewaves which maintain a constant field of view throughout the depth. Using these transmission types, a much larger section of the imaging view can be insonified at the same time, achieving high frame rates.	38

Figure 2.8. In conditions of vascular stenosis, a jet forms with high velocities that exceed the maximum measurable velocity of the system. This manifests as aliasing artifacts in the CFI where the fast flow regions appear to be flowing in the opposite direction.	41
Figure 2.9. a) In vector Doppler, aliasing is more complex as different pixels may contain aliasing in different combinations of the transmit-receive steering angles, leading to more unique erroneous vectors. b) A color-coded labeling of the aliasing artifact in the carotid phantom, with different colors encoding different aliasing patterns. 3 sample aliasing patterns are exposed while the frame contained 15 unique aliasing patterns.	42
Figure 3.1. Use of two CNNs for aliasing detection in ultrasound CFI. CNN #1 is for general segmentation of aliased pixels, which may include single and double aliasing. CNN #2 is specifically for the segmentation of double aliasing.....	52
Figure 3.2. Data features used for CNN training: 1) Doppler frequency; 2) Doppler power; 3) Doppler bandwidth; 4) speckle flow angle; 5) speckle flow speed. These features are derived from different parts of the CFI signal processing pipeline. Optical flow analysis was performed to derive flow speckle patterns.....	53
Figure 3.3. Design and implementation of CNN for CFI aliasing detection. a) Its architecture is based on a U-net with minor modifications, including leaky rectified linear units (ReLU), two dropout layers (denoted in black), and top-layer convolutional kernels whose number equaled to the number of training features. b) CNN training was done using augmented datasets that comprised M original feature frames (e.g. Doppler frequency maps) and their derived versions based on axial flipping, sign negation, and axial flipping with sign negation.	55
Figure 3.4. Dealiasing based on CNN inference output. (a) The original normalized CFI is first phase unwrapped once using the segmentation from CNN #1. External and internal boundaries of aliased CFI islands were extracted to determine the signage of full-cycle phase subtraction during phase unwrapping. $2v_{alias}$ denotes the original CFI dynamic range. (b) Stepwise illustration of the entire dealiasing algorithm. Phase unwrapping is performed for a second time using CNN #2 to resolve double aliasing cases. Denoising, inpainting, and median filtering were then applied to regularize the dealiased CFI values.	57
Figure 3.5. Illustration of the manual segmentation procedure, as applied to CFI frames with (a) single aliasing only, and (b) single and double aliasing. For the latter, the procedure was separately performed to obtain the aliasing reference map (covering both single and double aliasing) and the double aliasing reference map.	61
Figure 3.6. Radar plots of recall, precision, and intersection over union (IoU) for four feature combinations in training aliasing-detection CNNs. Over 7 CNN training and inference trials, the median values are shown in different test cases: (a) entire dataset for femoral CFI, (b) CFI frames taken during systole, (c) CFI frames taken during post-systolic recirculation. (d), (e), and (f) respectively show each metric's range (i.e. maximum minus minimum) in the 7 trials.	64
Figure 3.7. Segmentation performance of the aliasing-detection CNNs. (a-d) Original CFI at different phases of the cardiac cycle. (e-g) Aliased regions (including both single and double aliasing) identified by CNN #1, with errors color-coded. (h-k) Double-aliased regions identified by CNN #2, with error colors coded. All images are overlaid on top of the original B-mode images.	66
Figure 3.8. CFI visualization of femoral bifurcation before and after dealiasing. In (a), a time plot of spatially-averaged Doppler frequency is shown for the red box shown in (c); two time points of interest are indicated: systole (red line) and post-systolic recirculation (green line). For systole, three CFI frames are shown: (b) original	

without dealiasing, (c) after dealiasing using the proposed framework, (d) after manual dealiasing. Corresponding results for post-systolic recirculation are shown in (e), (f) and (g) respectively. 67

Figure 4.1. Multi-level parallelization of our real-time ELS-VD framework. At the frame level, each pixel can be processed independently. Within each pixel, execution of the ELS-VD algorithm involves sequential execution of 6 compute kernels (see Table 4.1 for each kernel's function). Within each kernel, various matrix arithmetic and computational operations can also be parallelized. 76

Figure 4.2. Peak-systolic flow vector profiles obtained from a carotid bifurcation phantom with 50% stenosis. Results are derived using: (a) least squares vector Doppler estimation, with aliasing artifacts highlighted in white; (b) the real-time ELS-VD framework; (c) manually dealiasing (shown for reference). 88

Figure 4.3. Real-time feasibility of the proposed framework. (a) Frame processing time and the number of flow pixels, shown over all 2,775 frames in a single cardiac cycle. (b) Reference flow speed profile in the white square denoted in Figure 4.2c. 89

Figure 4.4. Computing performance of the real-time ELS-VD framework as a function of different parameters, including: numbers of flow pixels C , maximum aliasing order L , block matching window dimension B , and slow-time block matching window size M . (a) Frame processing time as a function of C for $L = 1, 2, 3$ (with $B = 20, M = 30$); (b) Number of flow pixels that can be processed in real-time (25 fps) for different $B \times B$ block matching windows ($L = 1, 2, 3; M = 30$). (c) Number of pixels that can be processed in real-time (25 fps) as a function of M ($L = 1, 2, 3; B = 20$). 91

Figure 4.5. Processing time for each kernel in the real-time ELS-VD framework. Results are shown (in percentages) as a function of: (a) slow-time block matching window size M ($L = 1; B = 20$); (b) block matching spatial window size $B \times B$ ($L = 1; M = 20$); (c) Maximum aliasing order L ($B = 20; M = 20$). Number of flow pixels C was kept constant at 65,536 (256×256) in all cases. 92

Figure 5.1. The proposed aliasing resistant framework: a) Mean Doppler frequencies are processed through a least squares vector Doppler estimator to provide an initial estimate of the flow map. b) A CNN is then used to identify aliased pixels which are operated by the c) regularized ELS-VD estimator to achieve an aliasing-free vector flow map. 98

Figure 5.2. Architecture of the CNN used in this study. The CNN is an encoder-decoder with 4 resolution levels connected via skip connection. The input is the normalized flow speed and angle components derived using LS-VD estimation and the output is the aliasing probability map for the imaging view. 101

Figure 5.3. Aliasing segmentation performance on the test set: (a-c): Femoral bifurcation with 1 cycle aliasing; (d-f): Carotid artery with 1 cycle aliasing due to plaque; (g-i): Carotid artery with 1 cycle aliasing due to low PRF; (j-l): Femoral bifurcation with 2 cycle aliasing due to low PRF. (m-o): Carotid artery with two cycle aliasing due to low PRF and Probe tilt. The first column is a visualization of the vector flow map calculated using conventional LS-VD estimator. The second column showcases segmentation performance relative to the reference with errors color-coded. The right column is a visualization of the mean Doppler frequency maps from the one the transmit-receive steering angle combinations to emphasize aliasing cycle order and contours. 111

Figure 5.4. Vector flow maps obtained on the carotid bifurcation phantom with 50% stenosis at systole using (a) LS-VD estimation, (b) ELS-VD, (c) proposed framework and (d) LS-VD estimation at high PRF = 3000 Hz/angle. e) shows the aliasing segmentation produced by the second network. f) shows a visualization of the

mean Doppler frequency map obtained from the $(-10^\circ, 10^\circ)$ transmit-receive steering angle combination. g) The mean flow speed in the white square in (d) is shown to identify the phase of the frame in the cardiac cycle. 113

Figure 5.5. Aliasing resistant vector flow estimation at systole in *in vivo* acquisitions of the femoral artery (a-c), carotid artery with plaque (d-f) and tilted carotid artery at PRF = 2000 Hz/angle. Vector flow maps derived using LS-VD, accelerated ELS-VD and the proposed framework are shown in columns 1, 2 and 3 respectively.....114

Figure 5.6. Processing throughput analysis performed on the Carotid phantom (a, e), Femoral artery (b, f), Carotid artery with plaque (c, g) and tilted carotid artery at PRF = 2000 Hz/angle). In the top row, the number of flow pixels and aliased pixels are plotted over the flow speed profile throughout the cardiac cycle. In the bottom row, the processing time per frame of the accelerated ELS-VD framework is compared with the proposed framework. 115

Figure 6.1. The proposed system was realized on a programmable scanner with an ultrasound module for transmission and reception control, a processing module with a GPI and a PC-backed with a console application that controls the imaging processes. The rendered frames consist of a B-mode + flow speckle + vector flow map. The scanner was configured using the ultrasound module for plane wave transmission using two interleaved sequences: one for B-mode consisting of 30 steering angles; and one for vector Doppler consisting of two steering angles repeated 30 times each. The PC backend was responsible for data and operational management. Data was copied to the GPU in the processing module where GPU kernels were used to derive the B-mode image, the speckle image and the vector flow map. In live mode, each batch of RF data was processed to a framework output using least squares vector Doppler estimation. In replay mode, one batch of RF data was processed to 30 vector flow maps using aliasing resistant velocity estimation.....121

Figure 6.2. a) In live mode, one batch every 4 are processed to form one framework output using a series of GPU kernels that perform image formation, Doppler processing, least squares vector Doppler estimation. b) In replay mode, all batches are processed with a sliding window strategy to form 30 framework output frames per batch. Two CPU threads are developed to operate on a two-buffer system. The first thread utilizes GPU kernels for image formation and Doppler processing of a single batch of frames, the results of which are stored in the buffer. The second thread operates on a frame-by-frame basis to perform aliasing resistant vector flow estimation and image preprocessing. 123

Figure 6.3. a) The proposed system being used by the healthy volunteer to scan their carotid bifurcation. The output of the framework is displayed in a window on the screen. A screen capture of the framework output on the carotid bifurcation is shown in (b) live mode and (c) replay mode at systole. f) A rendering of the same frame shown in c) but without aliasing resistance. 133

Figure 6.4. A distribution of the framework output frame period is shown for (a) live mode and (b) replay mode. c) A plot of the flow speed in the red square in Figure 6.3c throughout the whole cineloop, along with the processing time of the velocity estimator per velocity frame estimated.....134

Figure 6.5. Two snapshots of the femoral bifurcation imaging demo are taken: peak systole (a-c) and recirculation (d-f). a, d) Framework output with aliasing resistance showing B-mode, speckle pattern and vector projectile visualization. b, e) Vector projectile visualization with least squares vector Doppler without aliasing resistance. c, f) Aliasing segmentation produced by the neural network. g) Plot of the flow speed in the entry to the femoral bifurcation (see red square) and processing time of the velocity estimator per frame. 136

Figure 6.6. a) Snapshot of framework output at systole in the carotid artery imaged at an angle. b) Vector flow visualization using conventional least squares vector Doppler without aliasing resistance. c) Doppler frequency at Tx-Rx steering angle $(-10^{\circ}, 10^{\circ})$ showcasing two-cycle aliasing that is labeled. d) Aliasing segmentation produced by the first component of the aliasing resistant vector flow estimator. e) Plot of the flow speed in the stenosed region (see green square) and processing time of the velocity estimator per velocity frame estimated.....137

Figure 6.7. a) Snapshot of framework output at systole in the carotid bifurcation phantom with 50% stenosis. b) Vector flow visualization using conventional least squares vector Doppler without aliasing resistance. c) Doppler frequency at Tx-Rx steering angle $(-10^{\circ}, 10^{\circ})$ showcasing two-cycle aliasing in the body of the jet. d) Aliasing segmentation produced by the first component of the aliasing resistant vector flow estimator. e) Plot of the flow speed in the stenosed region (see red square) and processing time of the velocity estimator per frame.....138

List of Tables

Table 3.1 Imaging Parameters Used in Data Collection.....	58
Table 3.2 RMSD Between the CNN-based Dealiased CFI and the Manually Dealiased CFI.	68
Table 4.1 Kernels Used in the Framework.	77
Table 4.2 Imaging Parameters Used in the Experiments.....	86
Table 5.1 Imaging Parameters Used in the Experiments.....	104
Table 5.2 Segmentation Performance with Threshold = 0.1.	110
Table 6.1 Array and Transmission Parameters.....	122
Table 6.2 Transmit-Receive Steering Angles Used.....	125
Table 6.3 Imaging Parameters for the Four Scenarios.	132
Table 6.4 Processing Times of Key Stages in the Framework for Live and Replay Modes.	135

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
API	Application programming interface
B-mode	Brightness mode
BMF	Blood mimicking fluid
CFI	Color flow imaging
CNN	Convolutional neural network
CPU	Central processing unit
CUDA	Compute unified device architecture
CVD	Cardiovascular disease
DAS	Delay-and-sum
DoS	Degree of stenosis
ECST	European carotid surgery trial
EDV	End-diastolic velocity
ELS-VD	Extended least squares vector Doppler
FIR	Finite impulse response
FPS	Frames per second
GB	Gigabyte
GHz	Gigahertz
GPGPU	General-purpose graphical processing unit
GPU	Graphical processing unit
HiFRUS	High frame rate ultrasound
HiVFI	High frame vector flow imaging

HRI	High resolution image
Hz	Hertz
IEEE	Institute of electrical and electronics engineers
IIR	Infinite impulse response
IoU	Intersection-over-Union
IUS	International ultrasonics symposium
JSON	JavaScript object notation
LRI	Low resolution image
LS-VD	Least squares vector Doppler
MHz	Megahertz
MRI	Magnetic resonance imaging
NASCET	North american symptomatic carotid endarterectomy trial
ORE	Office of research ethics
PCIe	Peripheral component interconnect express
PI	Pulsatility index
PRF	Pulse repetition frequency
PSV	Peak systolic velocity
PVA	Polyvinyl alcohol
RAM	Random access memory
ReLU	Rectified linear unit
RF	Radio frequency
RMSD	Root-mean-squared difference
Rx	Receive
SAD	Sum of absolute differences

SIMT	Single-instruction multiple thread
SNR	Signal-to-noise ratio
Tx	Transmit
TO	Transverse-oscillations
UFFC	Ultrasonics, ferroelectrics, and frequency control
US	Ultrasound
US4US	Programmable ultrasound scanner developed by US4US LTD.
VFI	Vector flow imaging
VPI	Vector projectile imaging

List of Symbols

α	Leak RELU parameter
Δt	Slow-time inter-pulse period
D_l	Sum of absolute differences between speckle frames for the l^{th} velocity candidate
ϵ	Aliasing error in normalized frequency units
ϵ_n	Aliasing error for the Doppler frequency measurements from the n^{th} steering angle
$\epsilon_n(x, y, t)$	Aliasing error for the Doppler frequency measurements from the n^{th} steering angle, evaluated at lateral position x , axial position y and slow-time frame t
ϵ	Vector of aliasing errors
θ	Angle of beam relative to transducer
θ_n	n^{th} transmit steering angle relative to transducer
θ_{tx}	Transmission steering angle
λ	Regularization factor
μ	Factor relating Doppler frequency to flow speed
$\hat{\sigma}(x, y, t)$	Doppler variance at lateral position x , axial position y and slow-time frame t
ϕ_n	n^{th} receive steering angle relative to transducer
A_{sten}	Cross-sectional of stenosed vessel
A_{open}	Cross-sectional of open vessel
$\mathbf{A}$	Matrix of cosines and signs of the transmit and receive steering angles.
$\mathbf{A}^\dagger$	Pseudo-inverse of $\mathbf{A}$
$\mathbf{A}^p$	Projection matrix of $\mathbf{A}$
$\hat{b}_d$	Doppler bandwidth
B	Block matching window length
c_o	Speed of sound in tissue

C	Number of pixels being analyzed
$\mathbf{d}$	Displacement vector
f_d	true Doppler frequency
$\hat{f}_d$	Measured mean Doppler frequency
f_o	Central frequency of transmitted pulse
f_{PRF}	Pulse repetition frequency
f_{max}	Maximum unaliased Doppler frequency that can be measured
f_{min}	Lowest unaliased Doppler frequency that can be measured
$\hat{\mathbf{f}}$	Vector of normalized Doppler frequency measurements
$\mathbf{f}$	Vector of true normalized Doppler frequency values
G	Slow-time ensemble length for Doppler processing
$\mathbf{I}$	Identity matrix
K	The number of aliasing vectors being considered
L	Maximum number of cycles of aliasing being investigated
M	Block matching correlation ensemble length
N	Number of transmit-receive steering angles in vector Doppler estimation
N_{tx}	Number of unique transmit steering angles
$\hat{P}_d$	Estimated Doppler power
$\mathbf{p}$	Pen-optimal estimate of the aliasing correction vector
$\mathbf{p}_k$	k^{th} Aliasing correction vector candidate
P	Number of frames received in one batch
$\mathbf{q}_i$	i^{th} candidate of ambiguous velocity correction vectors
Q	Number of batches received before a batch is processed for an output frame
r_k	Residual of the k^{th} aliasing correction candidate

$R(x)$	Cross-correlation function
$R_s(l)$	Auto-correlation function of s evaluated at lag l
$\mathbb{R}^n$	n dimensional set of real numbers
$\hat{s}(x, y, t)$	Flow speckle image evaluated at the lateral position x , axial position y and slow-time frame t
$s_m(i, j)$	$(i, j)^{\text{th}}$ block pixel in the m^{th} flow speckle frame
$s_{D,m}^l(i, j)$	$(i, j)^{\text{th}}$ block pixel in the m^{th} flow speckle frame after displacement using the l^{th} velocity candidate
t_0	Initial time instance
u_d	Normalized Doppler frequency
u_c	Clutter filter cutoff in normalized frequency
$u_n(x, y, t)$	Normalized Doppler frequency from the n^{th} steering angle at lateral position x , axial position y and slow-time frame t
$\hat{u}_d$	Measured normalized Doppler frequency
$\hat{u}_n(x, y, t)$	Measured normalized Doppler frequency from the n^{th} steering angle at lateral position x , axial position y and slow-time frame t
v_{max}, v_{alias}	Maximum velocity that can be measured.
v	Speed of scatterer relative to ultrasound beam direction.
$\mathbf{v}$	Velocity vector of scatterer
$\mathbf{v}(x, y, t)$	Velocity vector at lateral position x , axial position y and slow-time frame t
$\hat{\mathbf{v}}$	Estimate of velocity vector
$\mathbf{v}_D$	Dealiased velocity vector
$\mathbf{v}_l$	l^{th} velocity candidate
$\mathbf{v}_m(t-1)$	Median velocity vector in a small neighborhood in frame $t-1$
$\mathbf{v}_{pre}$	Preliminary estimate of the velocity vector using least squares vector Doppler estimation
v_y	Axial component of velocity vector

v_x	Lateral component of velocity vector
$x_{i(CNN)}$	i^{th} CFI pixel derived using CNN-based dealiasing
$x_{i(manual)}$	i^{th} CFI pixel derived using manual dealiasing
$w[g]$	Clutter filtered and beamformed analytic signal at slow-time frame g
$w_n[g]$	Clutter filtered and beamformed analytic signal at slow-time frame g from the n^{th} steering angle
$w_{cc}[g]$	Clutter filtered and beamformed analytic signal after coherent compounding of all steering angles signals
W	Ultrasound beam width
z_0	Depth of interest being interrogated along ultrasound beam
z_{max}	Depth of imaging
$\mathbb{Z}$	Set of all integers

Chapter 1

Introduction

1.1 Motivation

Mapping the complex blood dynamics found in human arteries is crucial for assessing cardiovascular health (Steinman & Taylor, 2005) and understanding the physiology of the circulatory system (Younis et al., 2004). For example, blood flow dynamics are significantly altered in conditions of vascular stenosis (Thiriet, Delfour, & Garon, 2015): where the lumen of the vessel is narrowed leading to high flow speeds (1-3 m/s or more) and turbulence. Detecting and quantifying disturbed flow dynamics and high velocities is important for assessing the severity of disease and predicting the risk for morbid events such as stroke (Grant et al., 2003; von Reutern et al., 2012).

In the clinic, the modality of choice for assessing flow *in vivo* is ultrasound imaging (Oglat et al., 2018) due to the accessibility and safety of the modality but the existing commercial tools fail to capture complex and transient blood flow dynamics (Gill, 1985). That is because the tools found in the clinic today cannot simultaneously capture flow speed and direction for the entire imaging view and suffer from low temporal resolution. Moreover, measurements made using conventional scanners are subject to significant inter-operator variability (Lui, Steinman, Cobbold, & Johnston, 2005), with errors in flow speed estimation reaching 30% in some scenarios (Hoskins, 2011). There is a number of recent innovations in ultrasound flow imaging that aim to address some of these shortcomings individually (Jensen, J. A., Nikolov, Yu, & Garcia, 2016a; Jensen, J. A., Nikolov, Yu, & Garcia, 2016b), but a reliable implementation tailored for complex blood mapping has yet to be realized at the bedside. One recurring downside of implementations found in the literature is a limited velocity range that can be imaged (Gododi et al., 2017; Hansen et al., 2019), which makes these techniques unsuited for pathological scenarios such as vascular stenosis. It is also important that ultrasound flow imaging can be achieved at the bedside for point-of-care and routine applications (Moore & Copel, 2011).

1.2 Objectives

The objective of this thesis is to devise an ultrasound-based framework for the *in vivo* vector flow mapping of complex and transient blood flow dynamics in human arteries at the bedside. This framework needs to meet the following criteria:

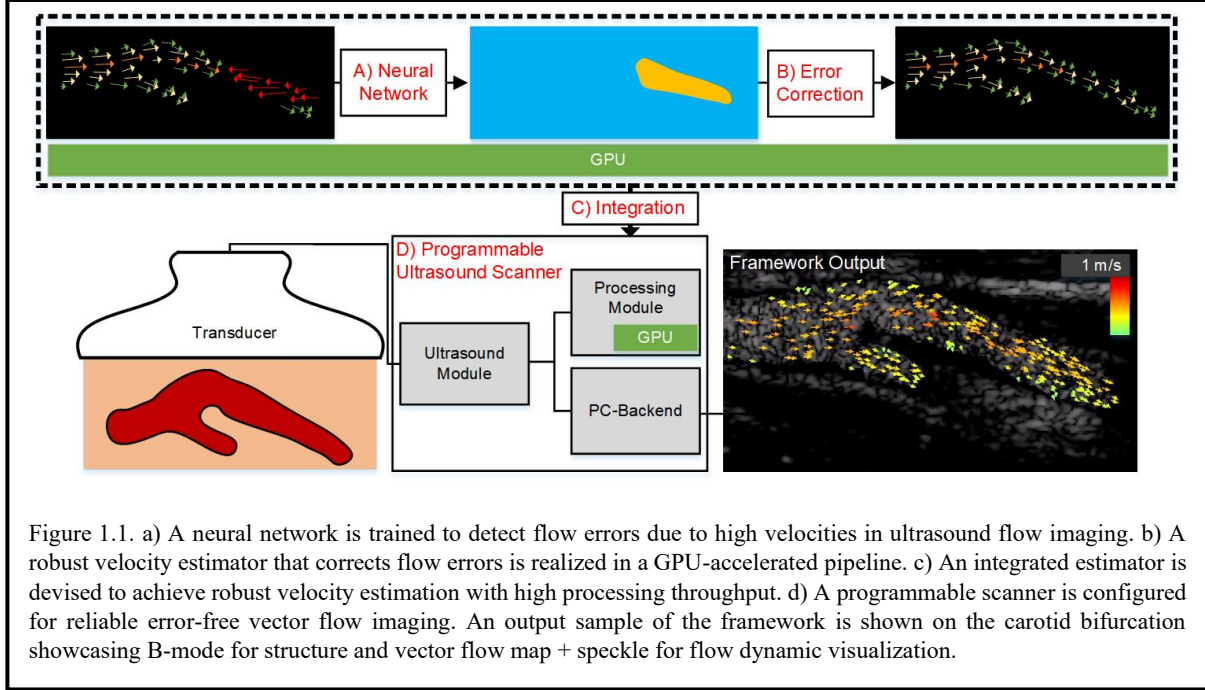
1. **Wide range of measurable velocities suited for physiological flow.** Velocities in pathological situations can range from <5 cm/s to 1 m/s or even more in high stenosis conditions (Thiriet et al., 2015). Capturing a wide range of velocities is important to ensure the utility of the framework across a range of applications.
2. **Mapped at high frame rates to capture transient dynamics (> 1000 fps).** Flow phenomena that are interesting for cardiovascular health, such as jets and vortices, may only exist for a fraction of the cardiac cycle (Ha, Choi, Park, & Lee, 2015; Hansen, Udesen, Gran, Jensen, & Nielsen, 2009; Steinman & Taylor, 2005; Younis et al., 2004), therefore high frame rates are necessary.
3. **Real-time playback of the vector flow information (>20 fps playback) for point-of-care application.** It is important that hemodynamic assessment can be done at the bedside and provide feedback immediately for practical and efficient screening applications (Moore & Copel, 2011).

1.3 Hypothesized Solution

An ultrasound framework for high frame rate (>1000 fps) vector flow mapping, capable of measuring a suitable range of velocities for physiological flow and a playback frame rate of at least 20 fps, can be realized on a programmable ultrasound scanner for use at the bedside with the following strategy:

1. Train a deep learning model to detect flow estimation errors stemming from high velocities (Figure 1.1a).
2. Utilize parallel computing for high throughput correction of high-velocity errors detected by the deep learning model (Figure 1.1b).
3. Devise a robust velocity estimator by integrating the deep learning model and the correction algorithm to achieve a high throughput velocity estimator with an expanded velocity range (Figure 1.1c).

4. Realize a real-time ultrasound flow imaging framework utilizing the robust velocity estimator on the programmable scanner for bedside use (Figure 1.1d).



1.4 Thesis Outline

- **Chapter 2** details the underlying principles of this thesis for ultrasound flow imaging, current state-of-the-art techniques and their limitations.
- **Chapter 3** describes the experimental work done towards training and evaluating a deep learning model for flow error detection in the existing and popular ultrasound modality, color flow imaging.
- **Chapter 4** presents a GPU-accelerated pipeline for robust velocity estimation.
- **Chapter 5** describes the integration of the deep learning principles for error detection with the GPU accelerated pipeline to achieve a robust high throughput velocity estimator.
- **Chapter 6** describes how the flow estimator is realized in a vector flow imaging framework on a programmable ultrasound scanner at the bedside.
- **Chapter 7** integrates the contributions of this thesis and discusses future directions.

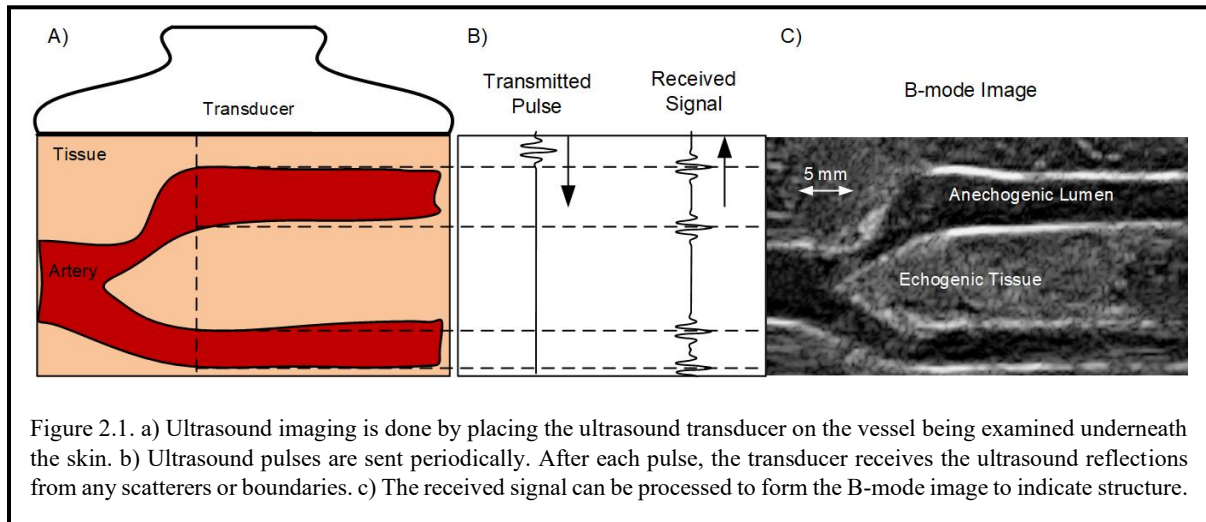
Chapter 2

Ultrasound Flow Imaging, State-of-the-Art and Limitations

This chapter provides a brief overview of ultrasound flow imaging, the operating principles behind basic and relevant state-of-the-art techniques, and the limitations of those techniques. The chapter finally identifies a potential application for the proposed framework.

2.1 Ultrasound Imaging as a Tool for *In Vivo* Blood Flow Estimation

Ultrasound is currently the go-to tool for blood flow imaging (Bierig & Jones, 2009). Alternative techniques for flow imaging exist but are often inferior in a number of aspects. Magnetic resonance imaging (MRI) can be used to estimate flow speed with high spatial resolution, but the modality is expensive, inaccessible and has a low temporal resolution (Krishnamurthy, Cheong, & Muthupillai, 2014). Computed tomography angiography may be used to detect flow and map vessel geometry but cannot map flow velocities throughout the vasculature (Kumamaru, Hoppel, Mather, & Rybicki, 2010) and also suffers requires ionizing radiation. Optical methods have seen some successes (Chávez-Cartaya, Ramirez-Romero, Calne, & Jamieson, 1994), but light cannot penetrate the human body very well and this limits the modality to very superficial applications. In all, ultrasound has a clear advantage for non-invasive, accessible imaging of blood flow at the bedside.



Ultrasound imaging is done using a pulsed-echo sensing mechanism (Cobbold, 2007) where a series of ultrasound pulses are emitted using an ultrasound transducer into the medium (Figure 2.1a). These are subsequently reflected by tissue boundaries or scatterers (ex: blood cells) and then

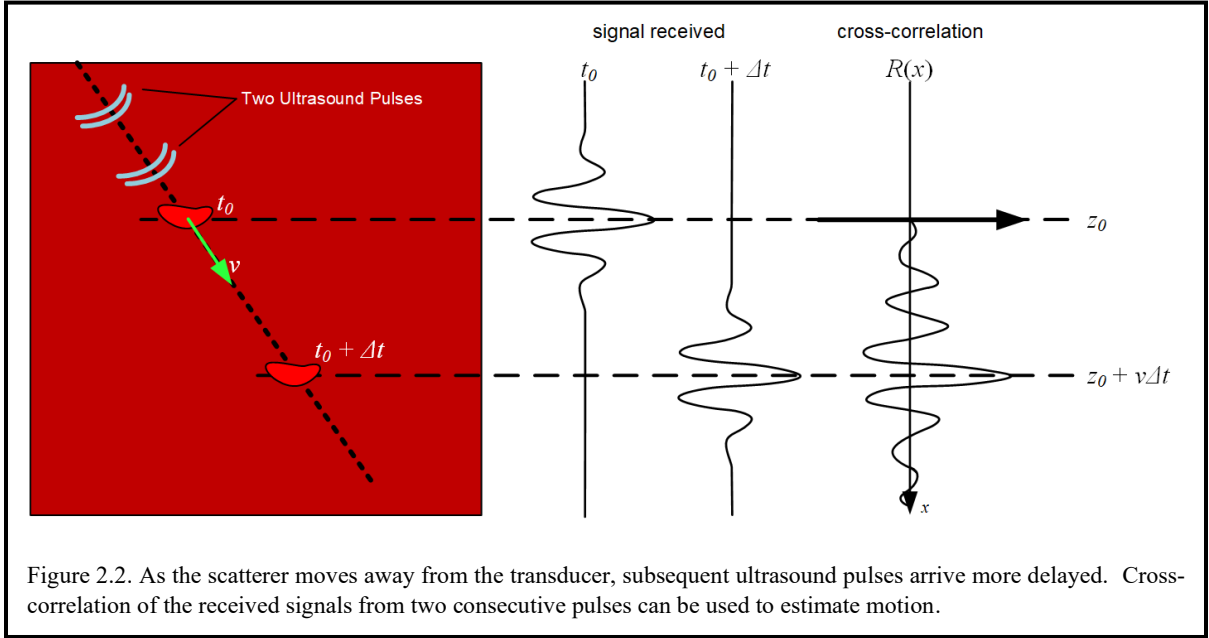
received by the transducer at the surface (Figure 2.1b). They arrive delayed according to the depth of the scatterer from which they originate. An ultrasound scanner typically insonifies a region of interest and receives the reflected pulses from that whole region. The received data is filtered and beamformed to form an image that depicts the received reflections at each location in the region after interference. Following envelope detection and log compression, an image of the tissue structure known as **Brightness-mode Image** (B-mode) can be formed (Lu, J., Zou, & Greenleaf, 1994). An example B-mode image is shown in Figure 2.1c. Blood has lower echogenicity than tissue and thus the lumens of vessels typically appear as dark regions relative to the brighter tissue. B-mode is used to examine the structures of the body noninvasively, including blood vessels (von Reutern et al., 2012) and organs (Sorensen et al., 2013).

2.1.1 Motion Estimation in Current Ultrasound Scanners.

Current ultrasound scanners in the clinic can only indicate motion in the direction of the ultrasound beam. This is done by tracking ultrasound reflections that originate from scatterers in the region of interest across different pulses. To estimate axial motion from the received signals, two approaches are available: time-domain methods (Hein & O'Brien, 1993) or phase-domain methods (Merritt, 1987). In time-domain methods, the received signals in a neighborhood around the location of interest are cross-correlated across two consecutive pulses. Using the pulse repetition frequency (PRF), f_{PRF} , the speed of the scatterers at that location can be calculated. To illustrate this principle in its simplest form, consider a moving scatterer, for example a blood cell, originally at depth z_0 , moving with speed v relative to the transducer, depicted in Figure 2.2. The beamformed signals on the transducer along the line of interest at time instances t_0 and $t_0 + \Delta t$ where $\Delta t = \frac{1}{f_{PRF}}$ are shown along with their cross-correlation function $R(x)$ where x is the sample number in each received signal. It is evident that the cross-correlation will peak at the displacement of the scatterer, which can then be used to calculate the speed at that region relative to the transducer (axial). Correlation-based methods ideally have no limit on maximum measurable speed. In practice, however, high-speed scatterers tend to move out of the imaging plane and the formed speckle patterns decorrelate (Jensen, J. A., 1993). As such, there is a limit on the maximum measurable axial speed using time-domain methods (Cobbold, 2007):

$$v_{max} = W \frac{f_{PRF}}{2} \quad (2.1)$$

where W is the beamwidth.



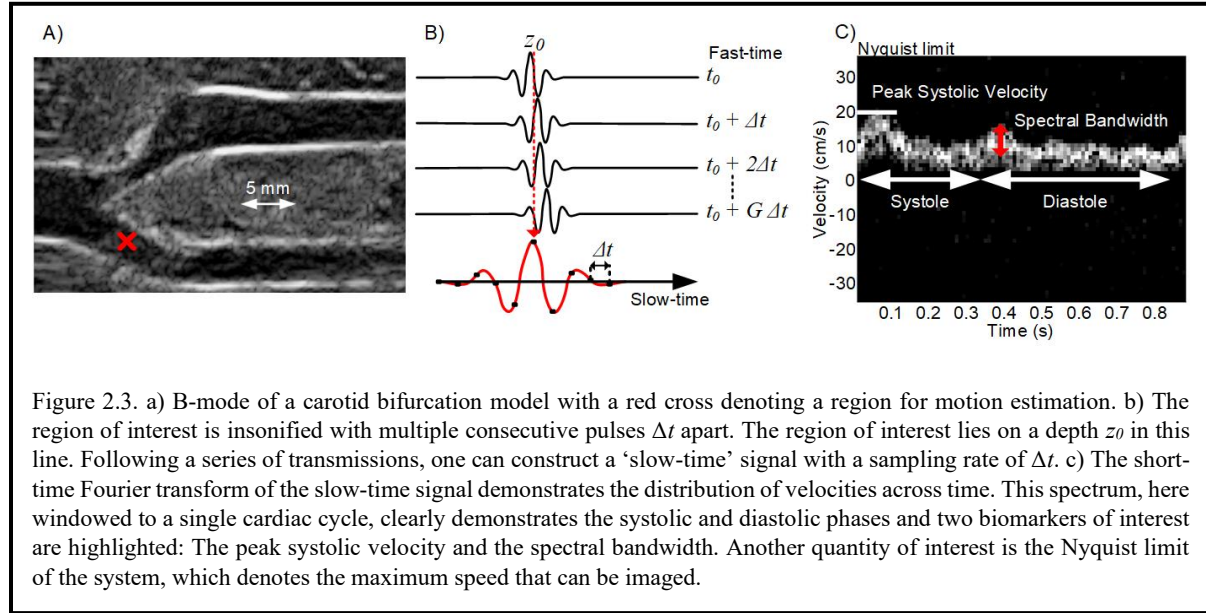
Axial motion can be alternatively estimated by calculating the change in phase between consecutive pulses at the location of interest, and this is the basis of phase-domain methods. Specifically, the change in phase between consecutive pulses at depth of interest z_0 is linearly proportional to the velocity v of the scatterer at an angle of θ relative to the transducer. The phase change per unit time is essentially a change in the frequency of the slow-time waveform, an effect sometimes referred to as the Doppler effect (Cobbold, 2007). The frequency of the slow-time waveform f_d is therefore related to the velocity of the scatters at z_0 :

$$f_d = \frac{2f_o v \cos \theta}{c_o} \quad (2.2)$$

where f_o is the central frequency of the transmitted pulses and c_o speed of sound. Hereinafter, I refer to f_d as the Doppler frequency.

In practice, there is typically a collection of scatterers at the location of interest whose reflections will superimpose. Consequentially, the received signals will contain a distribution of phase changes per frame, or in other words, a spectrum of frequencies. This spectrum at a location of interest (Figure 2.3a) is typically obtained by sampling the ensemble of G received signals at a particular depth (Figure 2.3b) to construct a slow-time signal where the sampling rate is equal to the PRF and then applying a short-time Fourier transform. The resulting spectrogram (Figure 2.3c) depicts the distribution of scatterer velocities in the location of interest and is the source of all current

quantitative measurements of blood flow in clinical practice. This spectrographic modality is typically referred to as **pulsed Doppler**.



While routinely used, the pulsed Doppler modality has significant sources of error. Firstly, it is evident from Equation (2.2) that actual flow speed can only be measured if the angle between the beam and the flow is known. Measuring the actual maximal velocity therefore requires the sonographer to manually estimate that angle of flow, leading to operator-dependency in the measurements (Lui et al., 2005). Secondly, the Doppler spectrum is often contaminated by spectral broadening that is intrinsic to the scanner and the imaging setup. As such, even a single scatterer moving at a fixed speed may show up on the spectrum as a band with some thickness. This spectral broadening stems from the transient time (Newhouse, Bendick, & Varner, 1976) that the scatterer is in the beam as well as nonuniformity in the received wave at the point of interest. As the transient time depends on the speed of the scatterer perpendicular to the transducer, spectral broadening is most pronounced when flow is at 90° relative to the beam. Since vessels usually run parallel to the skin, and by extension the transducer surface, spectral broadening is a significant source of error when calculating any Doppler related indices. As a result of these issues, measurements of the indices from pulsed Doppler can have errors reaching 30% (Hoskins, 2011), and some studies have shown significant variance in these indices when used in assessing stenosis severity (Evans, Barrie, Asher, Bentley, & Bell, 1980).

While pulsed Doppler is popular in the clinic, it is limited in revealing the spatial characteristics of flow as the sonographer can only view one region at a time. To reveal this information spatially and to aid in the identification of the flow regions of interest, a second ultrasound mode may be used: **color flow imaging** (CFI). CFI reveals the spatial flow profile every frame by presenting the central velocity from the spectrum at each pixel encoded in blue and red colors to denote direction relative to the transducer. The mean Doppler frequency $\hat{f}_d$ at the region of interest can be estimated using the Kasai estimator (Kasai, Namekawa, Koyano, & Omoto, 1985):

$$\hat{f}_d = \frac{f_{PRF}}{2\pi} \arg \left\{ \sum_{g=0}^{G-1} w^*[g]w[g+1] \right\} \quad (2.3)$$

where $w[g]$ is the beamformed signal the location of interest after analytic conversion, g is the slow-time sample index and $w^*[g]$ denotes the complex conjugate of $w[g]$. This estimator has been shown to be unbiased (Sirmans & Bumgarner, 1975) and can be computed rapidly. This has made it a cornerstone of CFI and a key factor in the real-time performance of the modality. A sample frame of CFI superimposed on a B-mode image of the carotid is shown in Figure 2.4 on a carotid bifurcation model. Blood mimicking fluid is moving towards the transducer in the upper branch, hence the color red, and away from the transducer in the lower branch, hence the color blue. The faster the flow speed relative to the transducer, the brighter the colors.

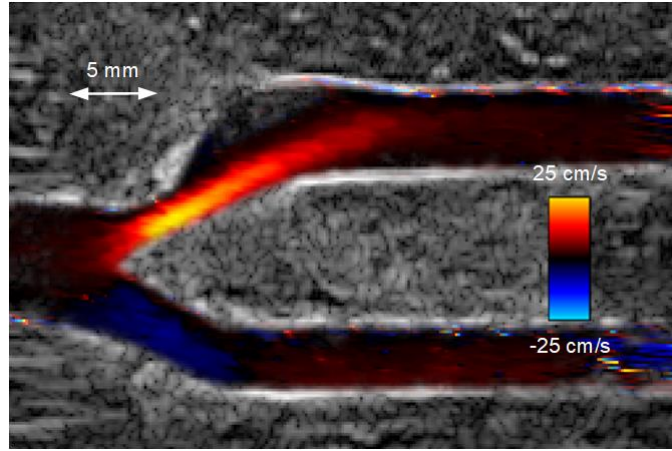


Figure 2.4. CFI of a carotid bifurcation model, showing the speed of flow relative to the transducer which in this image would be vertically above the carotid. As such, fluid in the upper branch would be moving towards the transducer, hence the color red. In the lower branch, blood would be moving away from the transducer, hence the color blue.

CFI is typically not used for quantitative analysis. The measurements shown per pixel are simply projections of the flow vectors in the direction of the ultrasound beam and thus operator-dependency is a significant problem. This is why clinicians often return to the Doppler spectrogram for quantification after they identify the location of interest via CFI.

2.2 Recent Motion Estimation Techniques

It is evident that current ultrasound scanners have serious limitations when imaging blood flow, particularly in pathological conditions such as stenosis or in tortuous vasculature where flow direction throughout the lumen can be ambiguous. The following sections will talk about state-of-the-art techniques in ultrasound imaging that have been developed to address some of the limitations in the current scanners and will, consequentially, will form the backbone of this thesis. As will be presented, however, there are still serious obstacles to these techniques when adopting them for robust vector flow imaging on the bedside.

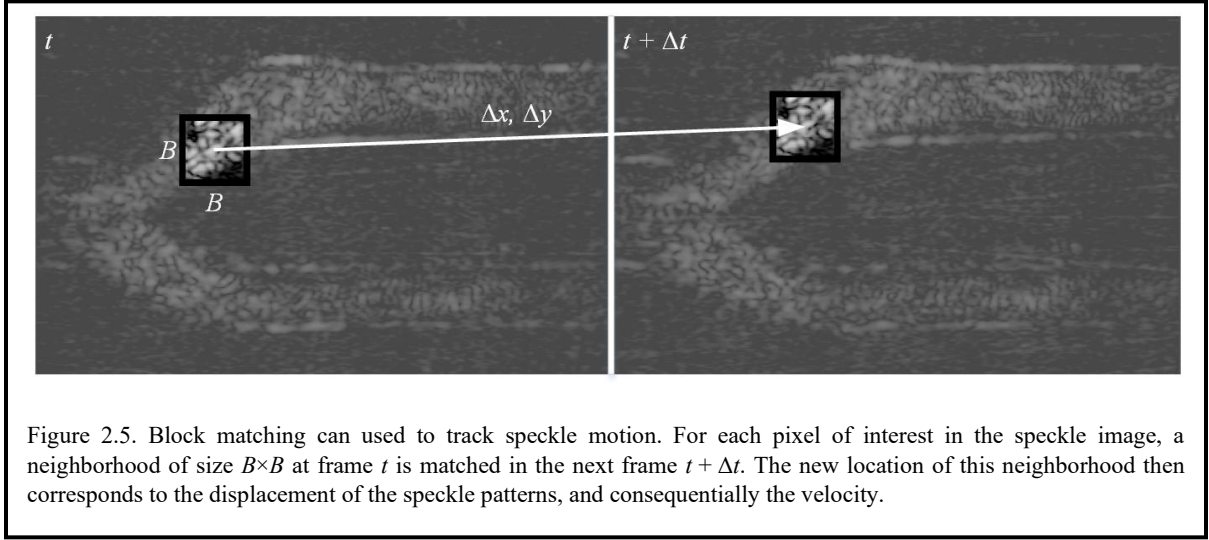
2.2.1 Vector Flow Estimation

As described previously, current scanners only measure flow speed relative to the transducer, with actual flow speed only attainable by assuming a certain beam-to-flow angle: a significant source of error. To overcome this issue, the aforementioned motion estimation techniques have been extended to derive flow in both lateral and axial directions (Jensen, J. A. et al., 2016a) in what is referred to as vector flow imaging (VFI). Following the same classification in the single-dimensional case, vector flow estimation techniques can also be classified as phase-based or time-based.

1) Time-based Techniques

Vector flow estimation can be derived by time-domain methods (Hein & O'Brien, 1993) that exploit the inter-frame correlation of the reflected ultrasound signals between consecutive frames to estimate motion, borrowing from the field of optical flow estimation in image processing. Due to the interference of waves reflected from multiple scatterers, speckle patterns emerge. As the scatterers move along the lumen, the speckle pattern is conserved provided imaging is done at a sufficient frame rate and with a broad beam. Extending the time-domain techniques used in the axial case, motion estimation can therefore be done by correlating the received speckle signals between frames in multiple directions (Jensen, J. A. et al., 2016a; Villagómez Hoyos, Stuart, Hansen, Nielsen, & Jensen, 2016; Yeung, Levinson, & Parker, 1998), often by applying optical flow algorithms from the particle image velocimetry domain (Poelma, 2016). In its simplest form, **speckle tracking** can be done using

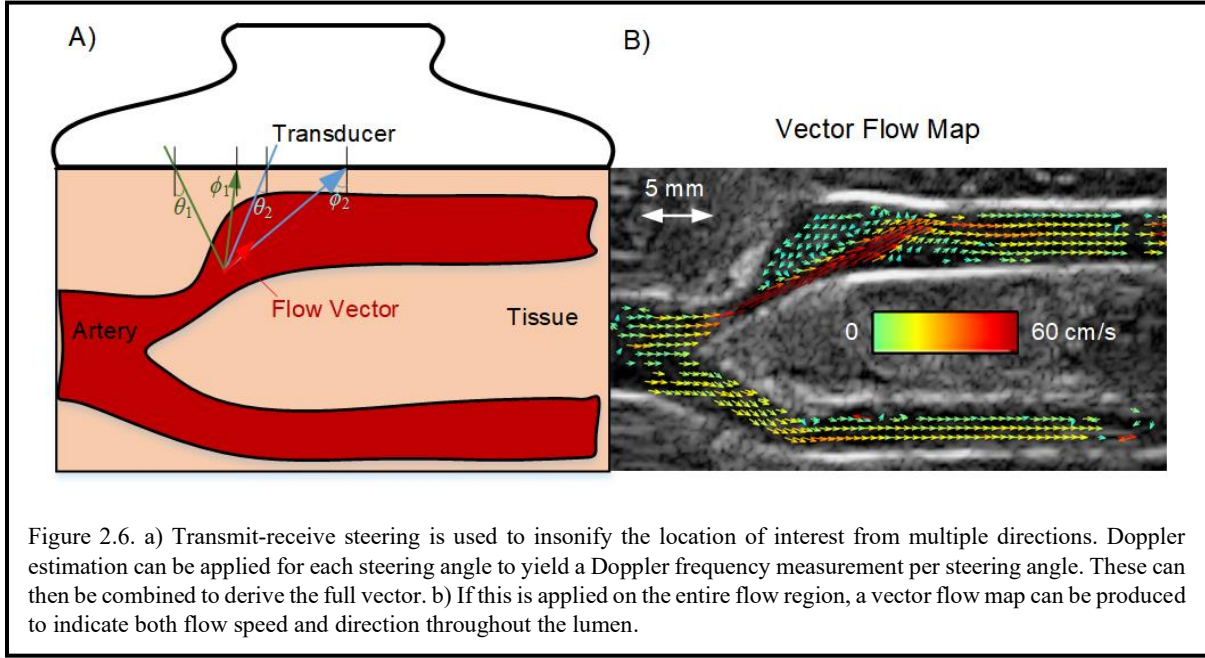
block matching (Loizou, Pattichis, & D'hooge, 2017). After filtering the beamformed signal to exclude tissue, the flow speckle patterns resulting from blood flow scatterers can be imaged (Yiu & Yu, 2013). Figure 2.5 shows two consecutive frames of the speckle images obtained on the carotid bifurcation model. For a particular region, velocity $\mathbf{v}$ can be derived by finding the displacement vector $\mathbf{d} = [\Delta x, \Delta y]^T$ of a $B \times B$ window between frame t and $t + \Delta t$. The velocity is then given by $\mathbf{v} = \frac{\mathbf{d}}{\Delta t}$.



Similar to the time-domain methods in Section 2.1.1, VFI using techniques such as speckle tracking is not ideally limited in the maximum velocity that can be measured. In practice, however, the quality of speckle patterns *in vivo* makes speckle tracking unreliable without contrast agents. Block matching for speckle tracking has indeed been extensively studied and a variety of implementations exist. Innovations exist in the form of adaptive kernel size and shape selection, regularization (Jiang & Hall, 2011) and sub-pixel estimation (Céspedes, Huang, Ophir, & Spratt, 1995).

2) Phase-based Techniques

There are several phased-based techniques (Jensen, J. A., 2017; Jiang & Hall, 2011; Udesen, Nikolov, & Jensen, 2005), but the most relevant to this thesis is the multi-angle **vector Doppler** estimation. The main principle behind this is the estimation of the phase difference between consecutive ultrasound pulses sampled at the region of interest using two or more different steering angles (Dunmire, Beach, Labs, Plett, & Strandness, 2000; Ricci, Bassi, & Tortoli, 2014) (Figure 2.6a).



Multi-angle Doppler estimation is achieved through a repeating and alternating transmission of ultrasound pulses with N_{tx} different unique steering angles. The data is beamformed on receive to form the transmit-receive steering angle combinations $\{(\theta_1, \phi_1), (\theta_2, \phi_2), \dots, (\theta_N, \phi_N)\}$, with N denoting the number of transmit-receive steering angle combinations. For each angle combination n , a Doppler equation as in Equation (2.2) can be constructed with the measured normalized Doppler frequency $\hat{u}_n = \hat{f}_n / f_{PRF}$ related to the flow velocity $\mathbf{v}$ by:

$$v_y(\cos \theta_n + \cos \phi_n) + v_x(\sin \theta_n + \sin \phi_n) = \frac{c_o f_{PRF}}{f_o} \hat{u}_n \quad (2.4)$$

where v_y is the axial flow speed, v_x is the lateral flow speed, c_o is the speed of sound in the medium and f_o is the central frequency of the emitted pulse. At least two steering angle combinations are necessary, but more are typically used to increase the robustness of the estimator through the use of least squares optimization (Yiu & Yu, 2016). With all N equations, one can construct the following system:

$$\mathbf{A}\mathbf{v} = \mu\hat{\mathbf{t}}, \quad (2.5)$$

$$\mathbf{A} = \begin{bmatrix} \cos \theta_1 + \cos \phi_1 & \sin \theta_1 + \sin \phi_1 \\ \vdots & \vdots \\ \cos \theta_N + \cos \phi_N & \sin \theta_N + \sin \phi_N \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} v_y \\ v_x \end{bmatrix} \quad \mu = \frac{c_0 f_{PRF}}{f_0} \quad \hat{\mathbf{f}} = \begin{bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_N \end{bmatrix}$$

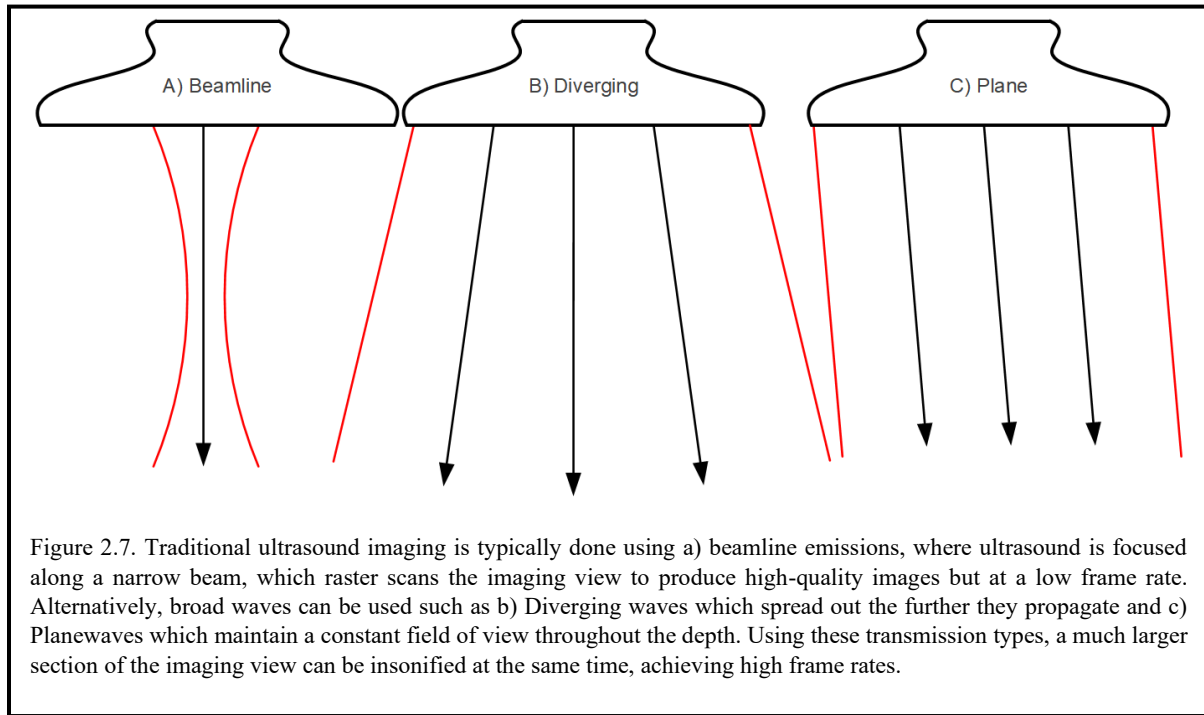
When $N > 2$, Equation (2.5) is overdetermined and the optimal velocity candidate $\hat{\mathbf{v}}$ can be derived using least-squared error optimization with the pseudo-inverse of $\mathbf{A}$:

$$\hat{\mathbf{v}} = \mu \mathbf{A}^\dagger \hat{\mathbf{f}}, \quad \mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (2.6)$$

This can be repeated for the entire imaging view to generate a vector flow map, shown in Figure 2.6b. Relative to speckle tracking, phase-based techniques such as multi-angle Doppler are less sensitive to noise and can be applied without contrast. Cross-beam (2 angles) vector Doppler has been shown to estimate the axial component of flow speed more robustly than speckle tracking via block matching (Swillens, A., Segers, Torp, & Lovstakken, 2010). A multi-angle vector Doppler estimator with four transmit-receive steering angle combinations was shown to achieve superior performance relative to speckle tracking, both in accuracy and standard deviations (Ekroll, Wigen, Fadnes, & Avdal, 2018).

2.2.2 High Frame Rate Ultrasound Imaging

Imaging frame rates of conventional ultrasound systems are insufficient to capture the complex transient flow dynamics and how they evolve throughout the cardiac cycle (Cikes, Tong, Sutherland, & D'hooge, 2014). The frame rate is primarily limited by the transmission strategies employed in these scanners, usually consisting of a beamline that raster scans the imaging view to construct the image line by line (see Figure 2.7a) (Cobbold, 2007). This is typically done to maximize the signal-to-noise ratio, and when combined with duplex imaging modes, can yield frame rates as low as 10 per second (Merritt, 1987). To overcome this limitation, new transmission strategies were devised that moved away from focused emissions, where theinsonified region is limited, to unfocused waves, where the entire imaging view is insonified with one pulse. These transmission schemes support high frame rate imaging ultrasound imaging (HiFRUS) on the order of 1000 fps (Tanter & Fink, 2014).



Specifically, two types of unfocused emissions are used: diverging (Jensen, J. A., Nikolov, Gammelmark, & Pedersen, 2006; Nikolov & Jensen, 2002) and planar (Montaldo, Tanter, Bercoff, Benech, & Fink, 2009). In diverging waves (Figure 2.7b), the focal point of the emission is placed behind the transducer, such that the wave spreads around as it progresses through the tissue, insonifying a large width with depth. Alternatively, plane waves (Figure 2.7c) place the focal point at an infinite distance away and thus the width of the wavefront remains consistent. Compared to diverging waves, plane waves might insonify a smaller region but have a more consistent phase profile that makes processing easier, which in turn makes vector flow estimation easier. In all these techniques, SNR and resolution can be significantly improved by averaging the images from multiple emissions, which in turn reduces the effective frame rate (Denarie et al., 2013). Ultimately, it is a tradeoff that the sonographer or the engineer has to tune for their specific application.

HiFRUS has been crucial in making vector flow imaging practical (Jensen, J. A. et al., 2016b). For example, speckle tracking greatly benefited of HiFRUS as insonifying the entire imaging view creates a coherent speckle pattern that can be more reliably tracked (Joos et al., 2018). Furthermore, the increased frame rate allows motion to be captured before significant out-of-plane motion takes place. HiFRUS has also been integrated with vector Doppler imaging and used to visualize complex blood

flow in bifurcations (Au et al., 2020). In all, HiFRUS is necessary to image the transient flow dynamics present in the cardiovascular system (Hansen et al., 2009).

2.3 Aliasing in Doppler Ultrasound Flow Imaging

2.3.1 The Principles of Doppler Aliasing

The phase-domain techniques used in Doppler flow imaging are limited in the maximum velocity that can be measured. This limitation stems from the slow-time sampling principles inherent in phase-domain techniques and which are discussed in 2.1.1. Flow that is too fast induces a phase shift in slow-time that exceeds the Nyquist limit determined by the PRF, resulting in aliasing errors that cause the incorrect flow speed to be reported (Pellett, Tolar, Merwin, & Kerut, 2005). Quantitatively, the maximum measurable flow speed can be derived by applying the maximum Nyquist limit to Equation (2.2) to yield the following relation:

$$v_{max} = \frac{f_{PRF} \times c_o}{4f_o} \quad (2.7)$$

There are three scenarios where Doppler aliasing is more likely to occur in three scenarios:

1) Limited PRF due to System Constraints and Requirements

The available data transfer bandwidth and power consumption in the system dictate the maximum frequency of pulses that can be received and transferred for processing, leading to hard limits on the available PRF in the system (Boni, Yu, Freear, Jensen, & Tortoli, 2018; Hu, Wu, Chang, & Li, 2021; Yiu, Walczak, Lewandowski, & Yu, 2019). Next-generation flow imaging techniques such as vector Doppler may also require the PRF to be shared among different sequences to achieve reliable performance (Yiu & Yu, 2016), further limiting the available PRF.

2) Reduced PRF to Capture Slow Flow Dynamics or Achieve Deeper Imaging

The PRF may be artificially lowered by the sonographer when examining slow-flow or looking for deeper vasculature (Buklioska-Ilievska et al., 2019). As the PRF increases, the minimum detectable velocity increases and therefore slow flow near the vessel walls or peripheral arteries may be lost in high PRF settings. Additionally, a high PRF reduces the maximum depth that can be imaged, and therefore deeper vasculature (ex: aorta) would require the PRF to be reduced during operation (Cikes et al., 2014).

3) Flow that is Fast or has a Low Beam-to-flow Angle

Blood flow speed in pathological conditions (ex: vascular stenosis in the carotid) can reach or even exceed 1m/s in pathological conditions and would typically exceed the available v_{max} (Nishimura et al., 2011). Alternatively, flow in tortuous vasculature or in the ventricles may have a low beam-to-flow angle and this leads to phase shifts in the slow-time signal exceeding the Nyquist limit (Au, Yiu, & Yu, 2019).

2.3.2 Aliasing in Color Flow Imaging

Aliasing is a serious problem in color flow imaging when applied to tortuous geometry or pathological vasculature. Under conditions of aliasing, the mean normalized Doppler frequency measured $\hat{u}_d$ for CFI is phase wrapped an integer number:

$$\hat{u}_d = u_d + \epsilon, \quad \epsilon = \pm 1, \pm 2 \dots \quad (2.8)$$

where ϵ denotes the discrete aliasing error and u_d is the true normalized mean Doppler frequency. The impact of aliasing is that regions of fast flow are rendered with an integer error in the color brightness, with the polarity likely flipped. An example of aliasing in CFI is shown in Figure 2.8 for the carotid bifurcation model. Here, the systolic phase is corrupted by aliasing near the stenosed region due to the formation of a jet with velocities that exceed v_{max} . In tortuous geometry, aliasing can make it difficult to interpret the flow dynamics using CFI as the combination of aliasing and changing flow direction manifests a mosaic of colors (Goddi et al., 2017).

One of the earliest methods for removing aliasing artifacts was the extended Kasai estimator (Lai, Torp, & Kristoffersen, 1997), where time-domain cross-correlation was used to determine the aliasing free estimate of u_d . Alternatively, staggered pulse emissions (Posada et al., 2016) have been successfully used for aliasing-free cardiac CFI. This expanded the measurable flow speed 6 times beyond the conventional limit but required significant modification to the transmission scheme. More recently, a dual-wavelength strategy has been used to expand the maximum imageable velocity four times (Porée et al., 2021). Image processing techniques were also introduced, including a supervised fuzzy logic approach (Shahin, Ménard, & Eboueya, 2000) and an unsupervised segmentation + phase unwrapping approach (Muth, Dort, Sebag, Blais, & Garcia, 2011). Image processing techniques have the advantage of avoiding any system modification and can thus be applied more easily relative to systemic techniques, but a reliable two-cycle dealiasing has yet to be demonstrated *in vivo* using these techniques.

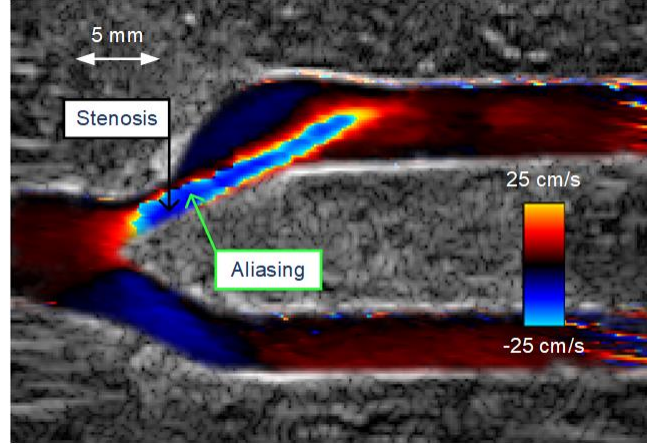


Figure 2.8. In conditions of vascular stenosis, a jet forms with high velocities that exceed the maximum measurable velocity of the system. This manifests as aliasing artifacts in the CFI where the fast flow regions appear to be flowing in the opposite direction.

2.3.3 Aliasing in Vector Doppler Imaging

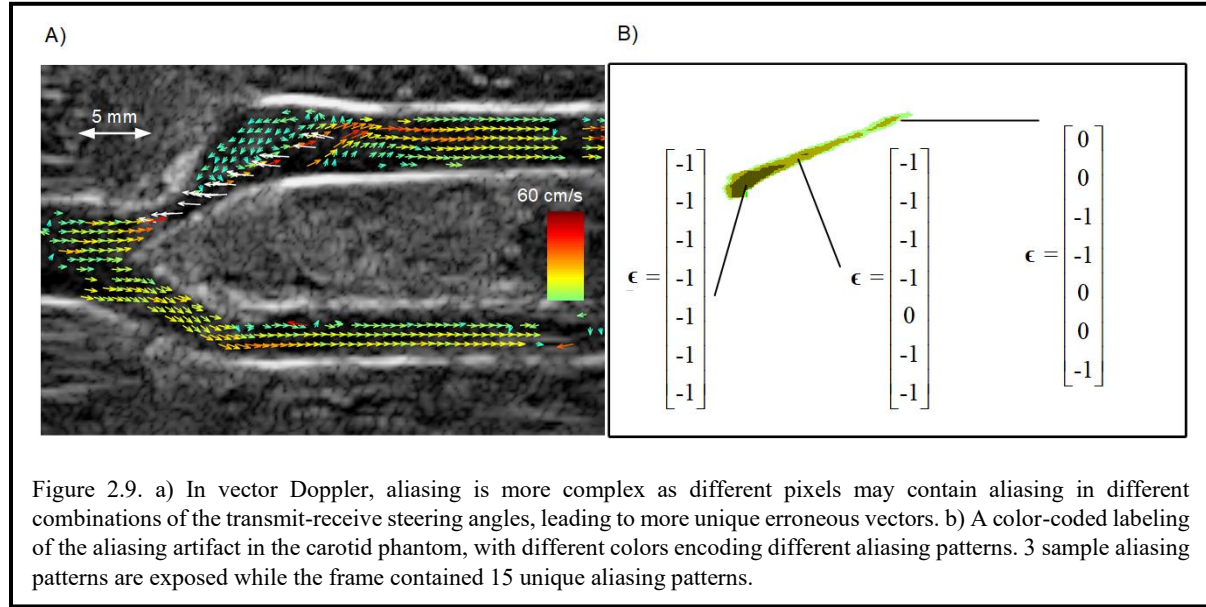
In multi-angle vector Doppler imaging, aliasing can occur in each of the transmit-receive steering angles combinations. Specifically, the measured normalized Doppler frequency vector $\hat{\mathbf{f}}$ used in Equation (2.5) is related to the true $\mathbf{f}$ by the following relation:

$$\hat{\mathbf{f}} = \mathbf{f} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix} \quad (2.9)$$

where $\boldsymbol{\epsilon}$ is an aliasing vector where each component denotes the discrete aliasing error ϵ_n for the n^{th} Doppler equation in Equation (2.5). When least squares velocity estimation is applied such as in Equation (2.6), the resulting velocity estimate is erroneous. Unlike in CFI, aliasing artifacts in vector doppler are multidimensional and are harder to interpret (Godidi et al., 2018). An example of this is shown in Figure 2.9a which demonstrates the vector flow output on the carotid bifurcation model with stenosis. Flow in the upper branch is aliased and depicts a number of different aliasing patterns (Figure 2.9b).

It is evident that aliasing is exceptionally problematic for vector Doppler and this is a major obstacle in the face of adopting these techniques for imaging flow in clinical applications. Specifically, any quantitative biomarkers that are derived from VFI are corrupted by these aliasing artifacts (Hansen et al., 2016; Hansen et al., 2017; Pedersen et al., 2014). Furthermore, the

requirement for multiple emissions from different angles effectively reduces the PRF and exacerbates the aliasing problem further.



The majority of efforts for addressing Doppler aliasing have focused on color flow imaging (Lai et al., 1997; Posada et al., 2016; Udesen et al., 2005). While dealiasing each individual Doppler frequency measurement used in vector Doppler measurement is theoretically a possible strategy to remove aliasing in multi-angle vector-flow, such an approach is unoptimized as it does not make use of all the information available from multiple transmissions. Speckle tracking has been used to correct aliasing artifacts in Doppler measurements in the axial direction by corroborating measurements from both vector flow modalities (Swillens, Abigail, Segers, & Lovstakken, 2010). Alternative solutions for expanding the PRF is to use packet transmissions (Jensen, J. A., 2019a; Jensen, J. A., 2019b) instead of alternating each transmit steering angle. For each transmit steering angle, a packet of consecutive plane waves steered with that angle is transmitted before proceeding to the subsequent angles. This transmission scheme overcomes the penalty that comes with more transmit steering angles but disrupts the continuity of data. Packet transmissions also limit the ensemble size available for Doppler processing and the imaging frame rate that can be achieved. Real-time aliasing-resistant vector Doppler (Ricci, Bassi, Dallai, Matera, & Tortoli, 2017) has been demonstrated in an ultrasound scanner using staggered transmissions (Madiena, Faurie, Porée, & Garcia, 2018) where velocity estimation exceeding 2 times the Nyquist limit was reported on a grid of 8×512 . Real-time functionality was achieved using programmable hardware and parallel beamforming circuits.

However, the requirement for additional transmissions at different PRFs comes with a significant load on the data transfer bandwidth and power consumption (Hu et al., 2021) that may not be supported by all scanners or prevent interleaving other sequences for B-mode imaging. The reported methodology used a cross-beam ($N=2$) approach for vector Doppler, which may be susceptible to significant variance (Yiu & Yu, 2016). The dual-wavelength approach (Porée et al., 2021) has also been used for aliasing resistant vector Doppler where velocity estimation exceeding 4 times the Nyquist limit was reported, without requiring interleaved transmissions at different PRFs. The needed broad-band signals however may impact imaging resolution.

Alternatively, aliasing in the multi-angle vector Doppler setting can be formulated as a least squares error optimization problem (Flynn, Daigle, Pflugrath, Linkhart, & Kaczowski, 2011), with the goal of estimating the correct aliasing correction vector $\mathbf{p} = -\epsilon$ that, when used to phase unwrap the frequency vector $\hat{\mathbf{f}}$, allows the estimation of the aliasing free velocity $\mathbf{v}_D$.

$$\mathbf{v}_D, \mathbf{p} = \underset{\mathbf{v}, \mathbf{p}_k}{\operatorname{argmin}} |\mathbf{A}\mathbf{v} - \mu(\hat{\mathbf{f}} + \mathbf{p}_k)|, \quad \mathbf{v}_D = \mu\mathbf{A}^\dagger(\hat{\mathbf{f}} + \mathbf{p}) \quad (2.10)$$

where $\mathbf{p}_k$ is the k^{th} aliasing correction candidate in a set of candidates being investigated. Assumptions about flow and the maximum number of cycles of aliasing in any component can be used to construct a reasonable set of candidates for investigation (Flynn et al., 2011). This approach was later developed further in the **extended least squares vector doppler** (ELS-VD) estimator (Ekroll, Avdal, Swillens, Torp, & Lovstakken, 2016). ELS-VD will form the basis of aliasing resistant velocity estimation in the proposed framework and thus will be described in the next section.

2.3.4 Extended Least Squares Vector Doppler

The objective of the ELS-VD estimator is to yield an aliasing resistant vector Doppler estimate $\mathbf{v}_D$ through a two-step process:

1) Discrete Squared Error Optimization of the Doppler System

A pen-optimal aliasing correction vector $\mathbf{p}$ is found through discrete squared error optimization:

$$\mathbf{p} = \underset{\mathbf{p}_k}{\operatorname{argmin}} |(\mathbf{A}\mathbf{A}^\dagger - \mathbf{I})(\hat{\mathbf{f}} + \mathbf{p}_k)|_2 \quad (2.11)$$

where $\mathbf{p}_k$ is an aliasing correction vector candidate in a discrete space of possible aliasing vectors. The space of aliasing correction vectors is constructed by considering that each component of $\mathbf{p}$ can take one value in the integer range of $[-L, L]$ where L is the maximum number of cycles of aliasing expected

in the signal. For a particular L , the space of aliasing vectors consists of $(2L + 1)^N$ candidates. Ideally, this optimization leads to a $\mathbf{p} = -\mathbf{c}$ but in practice, it cannot be used to differentiate between aliasing correction vectors $\mathbf{q}_i$ and $\mathbf{q}_j$ that are related by:

$$\mathbf{q}_i = \mathbf{q}_j + [l, l, \dots]^T \quad (2.12)$$

where $l \in \mathbb{Z}$. Therefore, a subsequent step is used to distinguish between $\mathbf{q}_i$ and $\mathbf{q}_j$.

2) Speckle Tracking to Derive Aliasing-free Velocity Estimate

The ambiguous candidates defined by Equation (2.12) are turned to velocity candidates by inverting the Doppler system:

$$\mathbf{v}_i = \mu \mathbf{A}^\dagger (\hat{\mathbf{f}} + \mathbf{q}_i) \quad (2.13)$$

The dealiased velocity candidate $\mathbf{v}_D$ is then the member of this set that correlates with speckle motion at the pixel of interest. Speckle tracking can be done via ensemble block matching on the speckle imaging (Section 2.2.1.1). The aliasing resistance of ELS-VD was demonstrated in simulated data and an *in vivo* scan of the carotid bifurcation (Ekroll et al., 2016). The framework was able to expand the measurable velocity rate by 5 times the Nyquist limit but was computationally demanding and introduced an overhead of 5 seconds per frame for $L=2$. This computational demand is problematic for point-of-care ultrasound applications (Moore & Copel, 2011).

2.4 Computing and Hardware Innovations in Ultrasound Imaging

One of the downsides in the aforementioned techniques for VFI and HiFRUS is the excessive processing times necessary to derive the flow maps. Various parts of the processing pipeline for ultrasound vector flow imaging, including beamforming, Doppler processing and aliasing resistant vector Doppler can be computationally demanding, and current scanners cannot realize these techniques for HiFRUS at the bedside.

To address these computing bottlenecks, one emerging trend in ultrasound imaging is parallel computing using graphical processing units (GPU)s. Specifically, components of the ultrasound processing pipeline have been reformulated to run on single or multi-GPU systems with orders of magnitude gains in processing throughput. For example, this has been applied to ultrasound image formation (Yiu, Tsang, & Yu, 2011; Yiu & Yu, 2015) where throughput of 3000 frames per second was achieved. This has also been applied to signal filtering (Bae et al., 2012; Chee, Yiu, & Yu, 2017), tissue tracking (Idzenga, Gaburov, Vermin, Menssen, & De Korte, 2014) and color flow imaging

(Chang, Hsu, & Li, 2009). One popular platform for parallel computing using GPUs is the compute unified device architecture (CUDA) (Garland et al., 2008), developed by Nvidia, which provides high-level GPU programming for ultrasound researchers.

In addition to computing innovations, new programmable ultrasound hardware is emerging to accommodate next-generation ultrasound imaging such HiFRUS. Open ultrasound platforms (Boni et al., 2018) are programmable ultrasound imaging systems that can be configured for novel transmission and reception sequences while allowing access to the raw ultrasound data for complex offline processing. Two such ultrasound open platforms were used throughout this thesis: (1) a Sonixtouch scanner modified with a custom Data acquisition board (Cheung et al., 2012), and (2) the US4US scanner (Yiu et al., 2019). With the custom acquisition board, the Sonixtouch can be configured for high frame rate imaging and provides access to the raw RF data for offline processing. This makes it suitable for algorithmic development when processing times are not important and was used in the development stages of this thesis.

The US4US (USPlatform; us4us Ltd., Warsaw, Poland) is a software-oriented platform that consists of 1) an Ultrasound module responsible for configuring and retrieving data from the ultrasound transducer, 2) a PC backend that gives the user control over the transmission and reception protocols of the ultrasound module and the ability to construct their own imaging pipeline and 3) a Processing module consisting of a GPU processor (GTX 1080Ti; NVidia Corporation, Santa Clara, CA, USA) connected by a Peripheral Component Interconnect Express (PCIe) backplane (BPG8032; Trenton Systems). With these modules, the US4US is therefore a highly configurable ultrasound imaging platform with strong computational capability. This makes it suitable for clinical translation as the novel algorithms can be integrated into the imaging pipeline with ease and may be realized in real-time for potential point-of-care solutions.

2.5 Deep Learning Applications in Ultrasound Imaging

Recently, neural networks have emerged as powerful tools in improving various aspects of ultrasound processing. Deep learning has been successfully used in improving the image quality of B-mode ultrasound (Gasse et al., 2017) and used for more advanced beamforming (Luchies & Byram, 2018). It has also been used for super-resolution imaging where the goal is mapping the micro-vasculature (Brown, Ghosh, & Hoyt, 2020) and some success has been seen in motion estimation for tissue (Østvik, Smistad, Espeland, Berg, & Lovstakken, 2018).

In this work, deep learning models were used for the purpose of semantic segmentation. A certain class of neural networks, fully convolutional encoder-decoder neural networks, have shown strong performance in segmentation for a variety of tasks (Badrinarayanan, Kendall, & Cipolla, 2017). The success of these networks in image segmentations can be attributed to 3 factors. 1) Being fully convolutional, this class of neural networks can exploit translation-invariant local patterns in the imaging view by breaking up the training set into small subsets. 2) The encoder-decoder architecture facilitates hierarchical pattern analysis by extracting and collating features at multiple scales (Badrinarayanan et al., 2017). 3) Later architectures are typically accompanied by skip connections (He, Zhang, Ren, & Sun, 2015) that transfer features from early stages in the network, pre-encoding, to the deeper parts. This has been shown to improve segmentation resolution, reduce overfitting and accelerate the training of large neural networks (Drozdzal, Vorontsov, Chartrand, Kadoury, & Pal, 2016). For the purpose of biomedical image segmentation, one such encoder-decoder, the U-net (Ronneberger, Fischer, & Brox, 2015), has become very popular and found success across multiple modalities. In ultrasound imaging, U-net has been used to segment tumors (Dong, Yang, Liu, Mo, & Guo, 2017; Yap et al., 2018), the ventricles of the heart (Smistad, Østvik, Haugen, & Lovstakken, 2017), lymph nodes (Zhang, Ying, Yang, Ahuja, & Chen, 2016) and more.

2.6 Potential Application for the Framework: Vascular Stenosis

One potential area of application for a robust bedside vector flow estimator is in imaging flow in conditions of vascular stenosis. Vascular stenosis is the narrowing of the lumen of a vessel and is symptomatic of vascular diseases, most notable of which is atherosclerosis but may otherwise be due to extrinsic factors such as tumors or a nearby vessel aneurysm (Thiriet et al., 2015). The reduction in the cross-sectional area in the lumen increases vascular resistance to flow and ultimately impairs blood supply to the organs at the end of the vessel. Common sites for vascular stenosis include the carotid artery (Goldstein, 2003) and the coronary artery (Ross, 1975) as well as peripheral arteries such as the femoral artery (Hansen et al., 2019). Regardless of the site, it is important that stenoses are detected and their severity defined as clinicians use this information to determine if and what intervention is needed to reduce morbidity and/or mortality. Stenosis severity and its risk are a function of its features, which can be classified into two: morphological and hemodynamical. Morphological features refer to the geometry and structure of the stenosis, such as the length of the lumen affected. Hemodynamical features refer to the differential blood flow dynamics resulting from stenosis. Both kinds of features are examined in practice, depending on the artery, the available

modality and the condition. Furthermore, including multiple features in the assessments often yields more diagnostic power.

2.6.1 Morphological Features

Vascular stenoses morphologically vary in their extent, length and shape (Thiriet et al., 2015), but the most common morphological feature used to assess the severity of stenosis is its degree, DoS, which is defined using the cross-sectional area at the throat (narrowest point) of the stenosis A_{sten} and the cross-sectional area in the open vessel (outside of the stenosed region) A_{open} . Specifically:

$$\text{DoS} = \left(1 - \frac{A_{sten}}{A_{open}}\right) \times 100\% \quad (2.14)$$

Vascular stenoses greater than 70% DoS are considered critical as blood supply is severely reduced in these cases. Stenoses less than 50% DoS are typically considered low-grade with a low risk of clinical events. In practice, the degree of stenoses often informs the clinicians' decision-making process by determining if surgical intervention is necessary to recover blood supply and/or minimize the mitigate an otherwise significant risk of a clinical event such as ischemic stroke or myocardial infarction. For example, the ECST (European Carotid Surgery Trialists' Collaborative Group, 1998) and NASCET (Ferguson et al., 1999) studies found that patients with a DoS in the carotid artery greater than 69% had significantly reduced stroke risk with a carotid endarterectomy compared to medical management, and this has been incorporated into clinical standards today. DoS may be estimated using B-mode ultrasound which can also reveal the location and extent of the plaque (Grant et al., 2003).

2.6.2 Hemodynamic Features

Vascular stenosis introduces complex geometries in the path of blood flow, ultimately impacting its dynamics, depending on how severe the stenosis is. In high-grade stenosis (>70%), blood is forced through a narrow opening (the constricted lumen) from a wider segment, akin to a flow nozzle, and this leads to the formation of a jet of high-speed flow, with velocities much higher than normal (Franz, Jump, Spalding, & Jenkins, 2013; Hood et al., 1996). Moreover, the stenosis typically introduces complex flow phenomena including vortices and turbulence in the vicinity of the stenosis and downstream from it. Currently, pulsed Doppler is used for detecting some of these hemodynamic changes. Studies have shown a positive correlation between the peak systolic velocity (PSV) obtained in pulsed Doppler and the degree of stenosis (Grant et al., 2003; von Reutern et al., 2012). Other

biomarkers can also be derived from Pulsed Doppler such as end-diastolic velocity (EDV) (von Reutern et al., 2012), pulsatility index (PI) (Granata et al., 2009) and spectral bandwidth (Johnston et al., 1986; Sheldon, Murie, & Quin, 1983).

Nonetheless, current ultrasound scanners cannot reveal the full picture of flow as they suffer from low spatial or temporal resolution and cannot capture multidirectional flow, which may occur in vascular stenosis (Hansen et al., 2017). Mapping flow speed and angle throughout the lumen of the vessel is important for understanding in detail the impact of flow dynamics on plaque formation and progression. One emerging biomarker derived using vector flow information is the vector concentration as a quantifier of flow complexity (Batschelet, 1981). Early studies show a correlation between vector concentration and DoS, suggesting a clinical role for vector flow imaging (Hansen et al., 2019).

2.7 Concluding Remarks

Ultrasound has a long history as the modality of choice for the non-invasive imaging of blood flow. Recent advances in high frame rate imaging and vector flow estimation have pushed the capabilities of this modality in tracking complex and transient flow dynamics. Nonetheless, there are currently limitations on the accessibility and utility of these next-generation techniques for clinical use, especially in pathological conditions such as vascular stenosis. System constraints and vector flow mapping considerations limit the maximum velocity that can be measured, and excessive processing time prevents point-of-care applications. Strategies involving GPU acceleration and deep learning are strong candidates for addressing these issues.

Chapter 3

A Deep Learning Approach to Resolve Aliasing Artifacts in Ultrasound Color Flow Imaging

Reprinted under Creative Commons Attribution License (CC BY), from Nahas, H.; Au, J. S.; Ishii, T.; Yiu, B. Y. S.; Chee, A. J. Y.; Yu, A. C. H. A Deep Learning Approach to Resolve Aliasing Artifacts in Ultrasound Color Flow Imaging, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, June 2020.

3.1 Introduction

Color flow imaging (CFI) (Merritt, 1987) has long been a popular modality for non-invasive mapping of blood flow *in vivo* (Oglat et al., 2018). It is particularly relevant to cardiovascular healthcare where CFI can be used to detect stenosis (Thiriet et al., 2015) and to monitor plaque formation through identifying abnormal flow phenomena such as jets, turbulence and flow reversal (Foley & Erickson, 1991; Merritt, 1987). However, CFI is known to be prone to aliasing artifacts that arise when the mean axial flow velocity at certain pixel positions exceeds the detectable limit (Terslev, Diamantopoulos, Dohn, Schmidt, & Torp-Pedersen, 2017). Such aliasing artifacts are fundamentally attributed to systemic limitations in its data acquisition paradigm, in which slow-time pulse-echo sampling is performed at each pixel position. Aliasing occurs whenever the pulse repetition frequency (PRF) used for slow-time sampling at each pixel position falls below the minimum rate (i.e. the Nyquist limit) needed to acquire an adequately sampled slow-time signal, whose mean Doppler frequency supposedly should correspond to the local average axial flow velocity according to the Doppler equation (Terslev et al., 2017). If aliasing is present at a pixel position, its resulting CFI pixel value would become misleading in different ways for the typical color encoding scheme that uses a red-blue bicolor hue to indicate the estimated mean velocity. First, the mapped color at a pixel position may be flipped to its complementary hue (i.e. from red to blue, or vice versa), thereby falsely indicating a change in the flow direction. Second, the mapped color may fall within a less saturated part of the same hue, such that a fast flow speed is deceptively rendered as slower flow in the same direction.

From a visualization standpoint, the presence of aliased color pixels in a CFI frame is known to jeopardize the proper interpretation of flow patterns in the image view. Indeed, in the presence of a curvy vessel geometry, the rendered CFI flow pattern often looks like a color mosaic even without aliasing (Gododi et al., 2018). Thus, if some aliased color pixels are present, gaining an intuitive understanding of the rendered flow profile becomes an extremely challenging task. Note that, from a flow quantification perspective, aliasing is an issue that affects the fidelity of not only CFI, but also vector Doppler that seeks to compute both axial and lateral velocity components (Jensen, J. A. et al., 2016a; Jensen, J. A. et al., 2016b). For the latter method, if flow vectors cannot be estimated accurately, they would not yield vector flow images with consistent visualization of the true flow profile (Cikes et al., 2014). Such an issue, in turn, would make it difficult to properly use the vector Doppler approach to effectively interpret complex flow dynamics that frequently appear in diseased vessels (Gododi et al., 2017; Hansen et al., 2016; Hansen et al., 2017).

To resolve aliasing artifacts in CFI data processing, several methods and algorithms have been proposed in the literature (Swillens, Abigail et al., 2010; Yiu, Lai, & Yu, 2014). For instance, an extended autocorrelation method has been proposed to double the maximum measurable velocity limit (Lai et al., 1997). Also, a staggered data acquisition sequence may be used to perform aliasing-free CFI below the Nyquist limit (Posada et al., 2016). These methods fundamentally involve changes to the system's front-end operations. Alternatively, aliasing artifacts may be tackled through image processing principles. For this type of dealiasing algorithms, a two-step process is commonly adopted: (i) identification and segmentation of aliased pixels and regions in the CFI frame, and (ii) correction of aliasing errors at the identified pixel positions based on flow profile properties. Accordingly, solutions based on fuzzy logic (Shahin et al., 2000) and unsupervised segmentation (Muth et al., 2011) have been proposed. These image-based dealiasing methods have succeeded in mitigating single aliasing errors, which correspond to cases when the actual Doppler frequency f_d exceeds the maximum unaliased frequency f_{max} by up to onefold of the unaliased frequency range (i.e. $f_{max} < f_d < 3f_{max}$, given that the unaliased frequency range equals to $f_{max} - f_{min} = 2f_{max}$). However, they are not effective in handling the more challenging double aliasing cases, which occur when f_d exceeds f_{max} by an amount between onefold and twofold of the unaliased frequency range; i.e. $3f_{max} < f_d < 5f_{max}$ (Muth et al., 2011). A more robust dealiasing technique is yet needed for CFI applications.

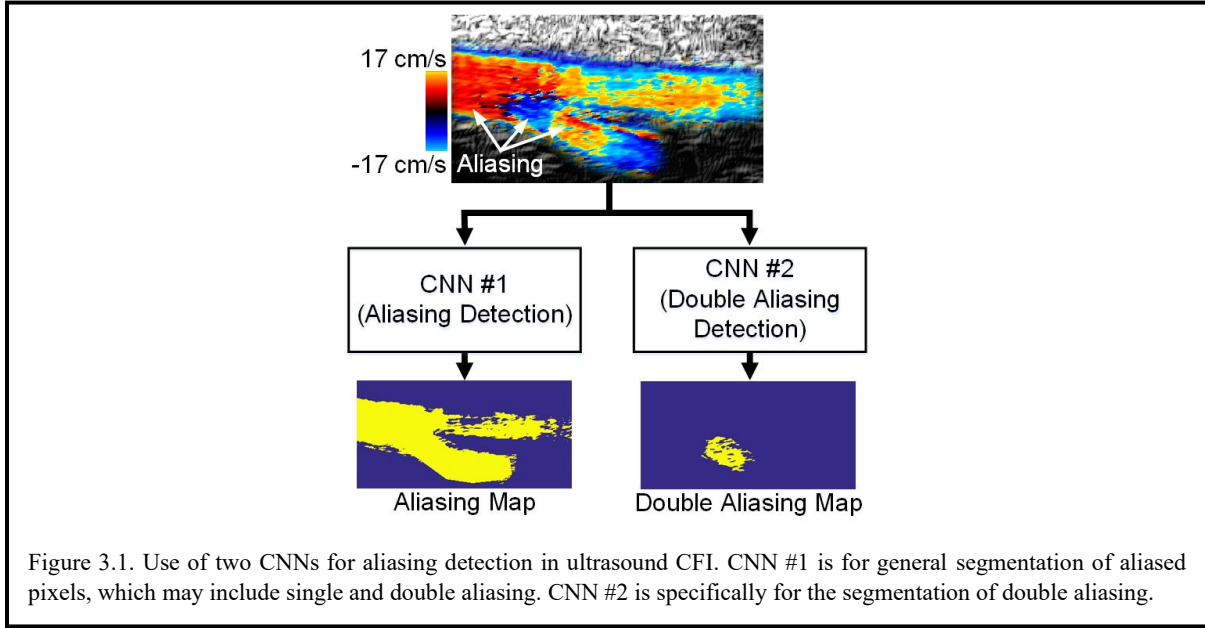
In this paper, we present a new dealiasing framework that is based on deep learning principles to suppress aliasing artifacts in CFI frames. We posit that aliased pixel positions and regions in a CFI frame can be effectively segmented by devising a convolutional neural network (CNN) that is trained to identify both single and double aliasing cases. We also posit that these aliased pixels may be subsequently removed via phase unwrapping. In approaching the CNN design, we have investigated the applicability of different ultrasound signal features as learning input for the CNN to segment aliased pixels in CFI frames. These features included data types derived from both Doppler spectral contents and flow speckle patterns.

Our dealiasing framework is built upon the recent success of CNNs in the domain of semantic segmentation (Garcia-Garcia, Orts-Escolano, Oprea, Villena-Martinez, & Garcia-Rodriguez, 2017), in which fully convolutional neural networks have been successfully applied on a variety of datasets (Long, Shelhamer, & Darrell, 2015). More specifically, our framework makes use of a class of CNN architectures known as encoder-decoder (Badrinarayanan et al., 2017), which has demonstrated superior performance in the segmentation of biomedical images (Dong et al., 2017; Yap et al., 2018). An encoder-decoder CNN is known to be potent in extracting and combining translation-invariant patterns at multiple scales (Ye & Sung, 2019), and these identified patterns can be refined to achieve high-resolution segmentation through skip connections (Drozdzal et al., 2016). This type of CNN is seemingly suitable for aliased pixel segmentation that may require information from a large neighborhood to disambiguate. Indeed, in a recent preliminary study (Nahas, Ishii, Chee, Yiu, & Yu, 2019), we have demonstrated the initial feasibility of devising an encoder-decoder CNN for single aliasing detection *in vitro*. Here, we shall expand the framework to address practical CFI scenarios *in vivo* that may include single and double aliasing.

3.2 Theoretical Principles

3.2.1 Overall Framework for Aliasing Segmentation

Our strategy for the identification of aliased pixels in ultrasound CFI was based on the use of two CNNs. As shown in Figure 3.1, CNN #1 was trained to detect aliased pixels (both single and double aliasing) in CFI frames, while CNN #2 was trained to specifically segment double aliasing cases. The second CNN was needed in this work in order to facilitate adaptive correction of double aliasing artifacts (to be discussed in Section 3.2.5). Training was done according to supervised learning principles, for which labelled datasets were generated.



3.2.2 Training Features for Aliasing Identification

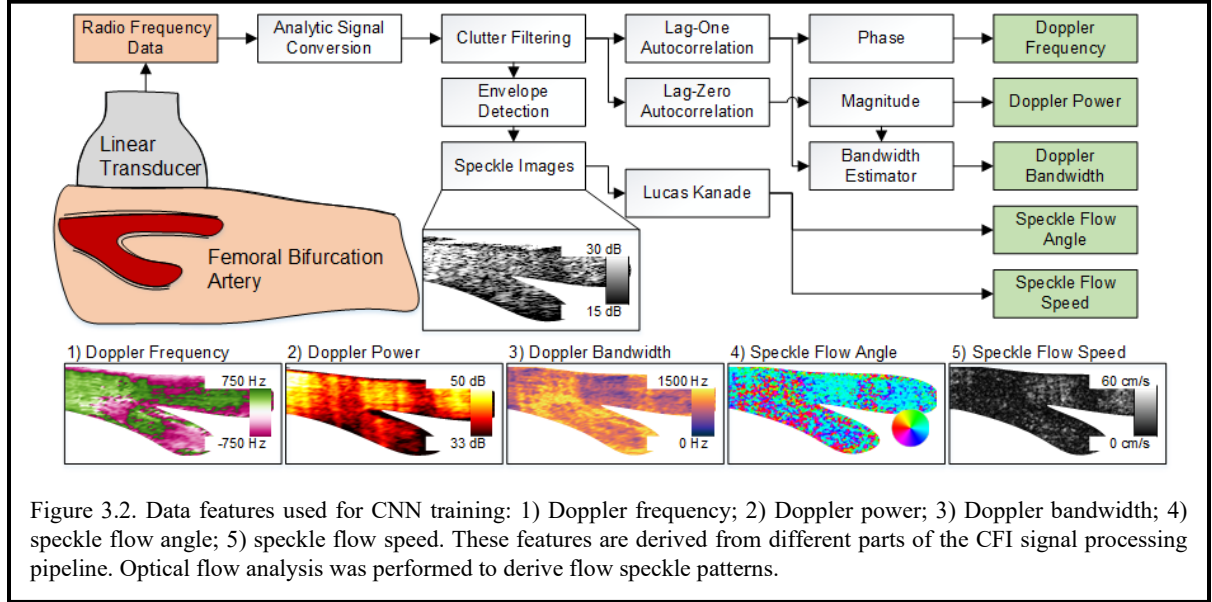
To successfully apply supervised learning to aliased CFI pixel identification, it is important to identify the relevant ultrasound information (or features) that can be used as training input. In this work, five different types of image-based datasets, as obtained from different steps in the CFI data processing pipeline as illustrated in Figure 3.2, were used to train CNNs for aliasing detection. These data features include: 1) Doppler frequency; 2) Doppler power; 3) Doppler bandwidth; 4) flow speckle speed; 5) flow speckle motion angle. For each image-based dataset, every pixel denoted the value of that data feature at the corresponding location in the image view. Further details on each training data type are given as follows.

1) Doppler Frequency

Aliasing typically emerges in CFI as abrupt disruptions and irregularities in the rendered color map in the image view. Since CFI maps are after all color-encoded maps of mean Doppler frequencies, the spatial distribution of mean Doppler frequency values is an obvious data type to include for the training of an aliasing detection CNN. In this study, the measured mean Doppler frequency $\hat{f}_d$ at each pixel position was calculated using the Kasai estimator (Kasai et al., 1985) (the phase of the lag-one autocorrelation function), as it is the standard algorithm used in clinical scanner development. $\hat{f}_d$ can be calculated as follows:

$$\hat{f}_d = \frac{f_{PRF}}{2\pi} \angle R_w(l) \quad \text{for } R_w(l) = \sum_{g=1}^{G-l} w^*[g]w[g+l] \quad (3.1)$$

where $w[g]$ is the g^{th} sample in the slow-time ensemble (which is an analytic signal), G is the ensemble size, $R_w(l)$ is the autocorrelation function at lag l , f_{PRF} denotes the PRF and $\angle$ denotes the phase of a complex data value.



2) Doppler Power

The clutter-filtered power Doppler map is often used to identify the presence of flow in the image view, and thus it could be useful for differentiating between tissue, flow and noise. As such, the spatial distribution of Doppler power values may serve as complementary input for aliasing prediction. Each Doppler power value $\hat{p}_d$ is simply the mean power of the slow-time ensemble after clutter filtering:

$$\hat{p}_d = \frac{|R_s(0)|}{G} \quad (3.2)$$

3) Doppler Bandwidth

Aliasing, especially at higher orders, can impact the Doppler bandwidth through smearing the Doppler spectrum, thereby resulting in high bandwidth values. Accordingly, Doppler bandwidth may

be another useful predictor of aliasing artifact. Based on the Kasai algorithm (Kasai et al., 1985), this quantity can be calculated as follows using the lag-one and lag-zero autocorrelation values:

$$\hat{b}_d = \sqrt{1 - \frac{|R_s(1)|}{|R_s(0)|}} \quad (3.3)$$

4) Flow Speckle Speed

Another potentially useful information for the detection of aliasing at a given CFI pixel position is the motion of speckle patterns at that pixel. From a signal processing perspective, flow speckle maps can be extracted from the log-compressed magnitudes of individual clutter-filtered slow-time ensembles. An optical flow algorithm can then be applied to these flow speckle maps to determine the motion vectors at different pixels. Subsequently, the flow speckle speed at each pixel can be determined as the magnitude of the corresponding motion vector. In this work, the Lucas-Kanade method (Baker & Matthews, 2004) was used as the optical flow algorithm.

5) Flow Speckle Angle

Based on the motion vectors derived from optical flow analysis, directional information about inter-frame flow speckle motion can be obtained by computing the angle of each motion vector. This information served as a complementary data feature for aliasing detection.

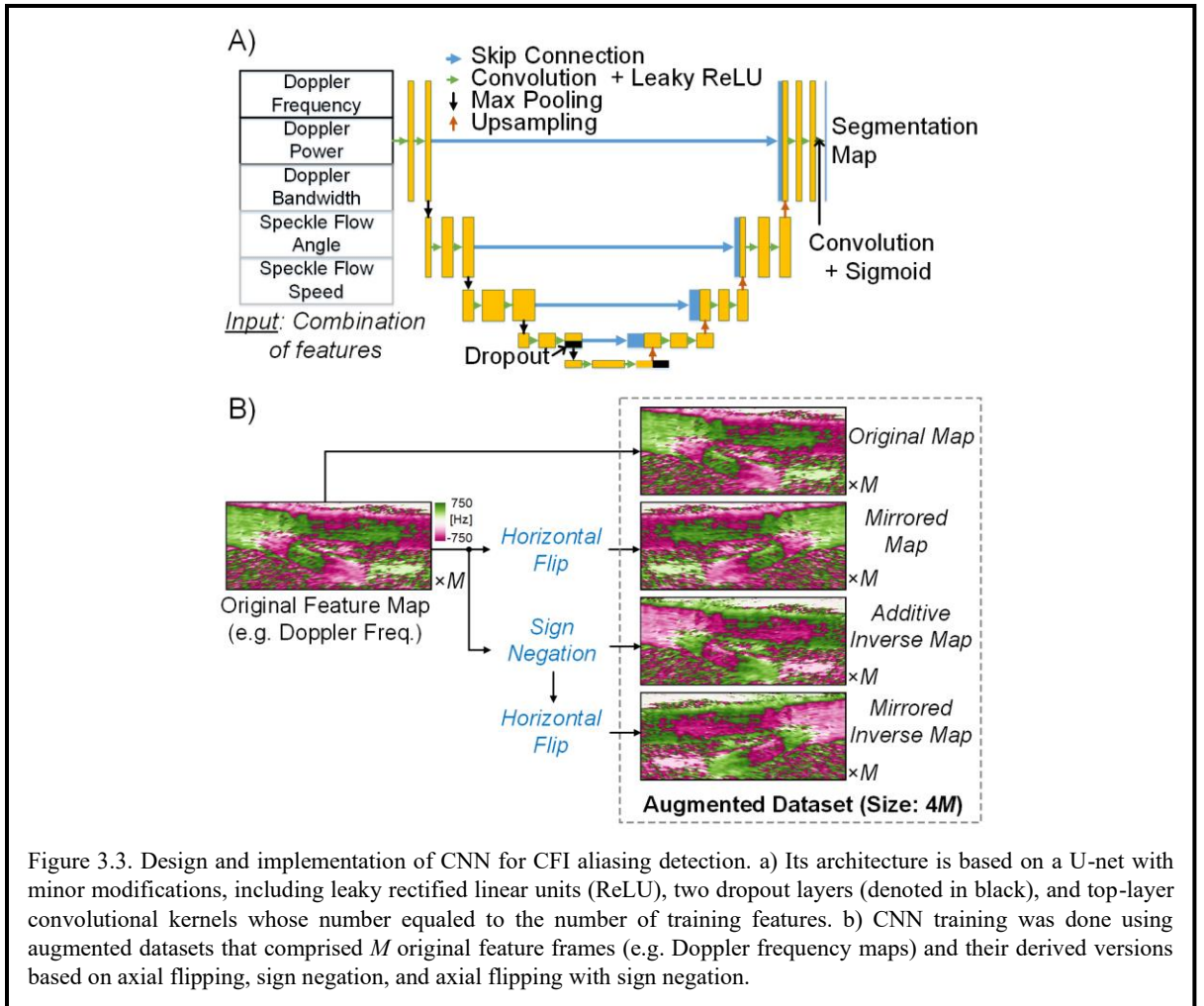
3.2.3 Candidate Combinations for Training Features

Four combinations of training features were investigated for aliasing identification in CFI. In all training combinations, Doppler frequency and Doppler power were included because they were respectively the source of CFI aliasing and a direct indication of the presence of flow at a CFI pixel position (which may be aliased). Doppler bandwidth and flow speckle motion characteristics were included in some training combinations to test their supplementary value to the training of CNNs for aliasing detection. Based on this arrangement, the tested training combinations are listed as follows. In each case, one frame of training sample consisted of multiple data feature channels.

- 1) **Doppler Frequency + Doppler Power:** This combination was the basic case for CNN training.
- 2) **Doppler Frequency + Doppler Power + Doppler Bandwidth:** This combination was used to test if the inclusion of Doppler bandwidth information in the CNN training process could improve the performance of aliased pixel identification.

- 3) **Doppler Frequency + Doppler Power + Flow Speckle Motion:** In this combination, the benefit of flow speckle motion information in aliasing segmentation was investigated. Flow speckle motion was rendered as two additional feature data channels (speed and angle). Thus, in total, including Doppler frequency and Doppler power, each training sample consisted of four data feature channels.
- 4) **Doppler Frequency + Doppler Power + Doppler Bandwidth + Flow Speckle Motion:** This combination integrated all data features. In principle, it is expected to give the best CNN training performance for aliasing detection.

3.2.4 CNN Architecture



The neural network architecture used in this work was based on the U-net, which is a specific implementation of an encoder-decoder CNN (Ronneberger et al., 2015). This architecture was used because of its known effectiveness in performing image segmentation at fine resolution (Drozdal et al., 2016). It expands beyond the CNN architecture devised preliminarily by our group to address single aliasing cases in CFI of flow phantoms (Nahas et al., 2019). As shown in Figure 3.3a, the CNN consists of an encoder part and a decoder part, each with 13 layers that are divided into 5 scales. Connecting the different scales are skip connections, where the output of an encoding layer is connected to a decoding layer of the same scale. The effective reception field of the encoder was 162×162 (pixels). The use of these network architecture parameters was empirically based on in-house pilot trials, and they were deemed to provide sufficient neighborhood information to discern aliased CFI pixels. Note that the use of multi-scale information is important in this CNN design because CFI aliasing can occur at varying regional sizes, from local to global discontinuities. Also, the use of skip connections is critical to improve both the training of deep networks and the segmentation resolution (Drozdal et al., 2016). These connections can facilitate the training of CNNs that can track the often irregular and sharp boundaries of aliased regions in the CFI image view.

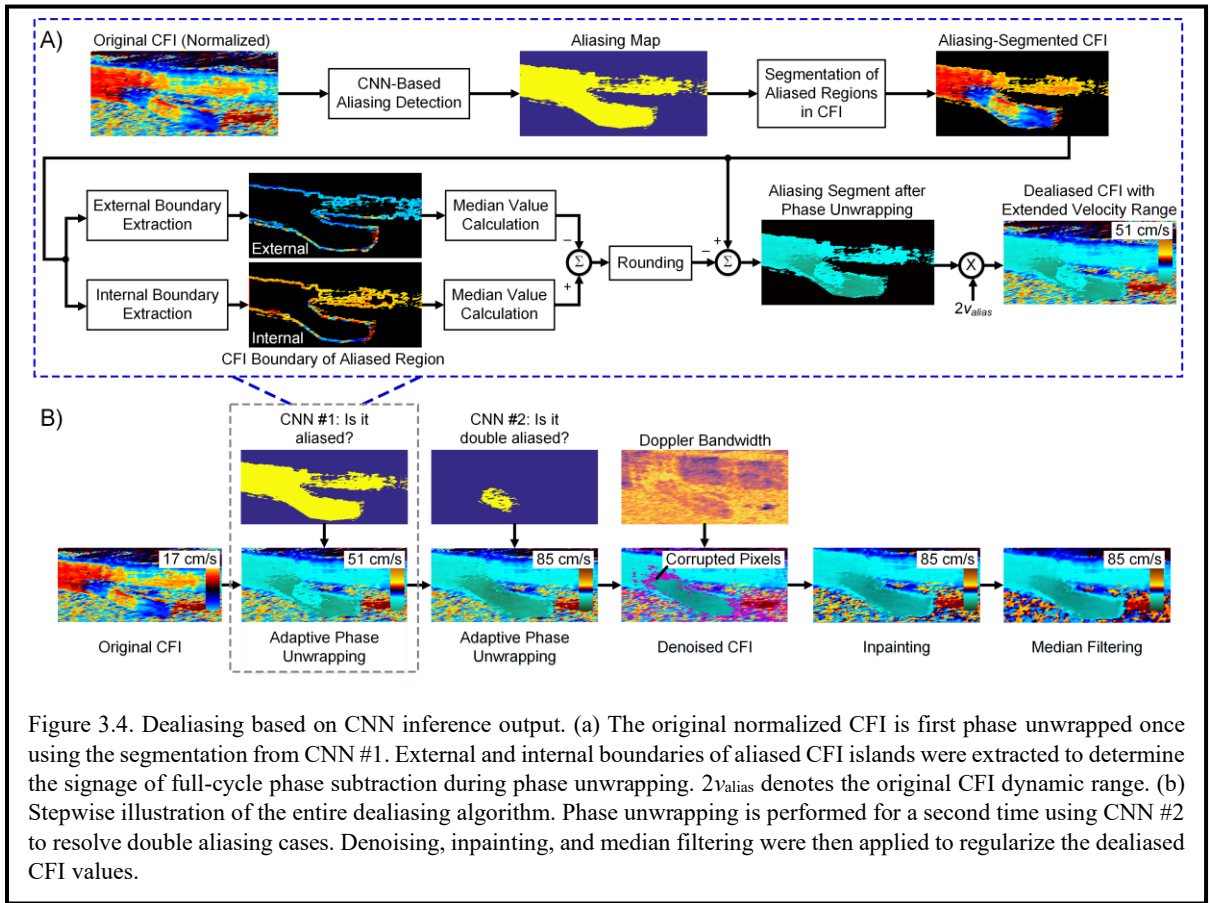
Since the CNNs used in this work were expected to process multiple channels of data features for each input frame, the first layer in the U-net architecture was modified from the original formulation (Ronneberger et al., 2015) to include more convolutional kernels that equaled to the number of data feature channels in each input frame. In addition, leaky rectified linear units (ReLU) were used, and the leak parameter was set to 0.3. This CNN design choice was made to improve learning and to reduce the occurrence of “dying ReLU” cases (Lu, L., Shin, Su, & Karniadakis, 2020). Moreover, two 50% dropout layers were added (denoted in black in Figure 3.3a) to improve the generalizability of the model. The size and number of filters were left as they were in the original U-net design (Ronneberger et al., 2015).

3.2.5 Dealiasing and Image Regularization

Correction of aliasing errors was performed using an in-house adaptive phase unwrapping algorithm that is inspired by the literature (Muth et al., 2011). In brief, it works by assuming that the segmented pixels were clustered in connected islands. As illustrated in Figure 3.4a, for each connected island, the internal and external boundary’s normalized CFI pixel values were extracted, and their median was evaluated. The difference between the internal and external boundary medians was then rounded to yield potential outcomes of -1 , 0 , or $+1$. In turn, this rounded difference outcome served to determine

whether a negative or positive full-cycle phase should be subtracted from the CFI pixels of that aliased island to achieve phase unwrapping (or to not perform phase unwrapping if the rounded difference was 0).

In our dealiasing procedure, phase unwrapping was applied twice as shown in Figure 3.4b. First, based on the segmentation results from CNN #1, phase unwrapping was applied once to the CFI pixel values of identified aliasing regions (i.e. either adding or subtracting full-cycle phase based on the algorithm in Figure 3.4a). The output of this step was then fed through the same algorithm for a second time, wherein the segmentation results from CNN #2 was used to perform another round of phase unwrapping on double aliasing cases. After these two rounds of phase unwrapping, the normalized Doppler frequency range effectively extended from $[-0.5, +0.5]$ to $[-2.5, +2.5]$.



At some pixel positions, aliasing may corrupt the pixels beyond utility. Indeed, our in-house spectral analysis showed significant folding due to aliasing and the normalized bandwidth approached unity in those pixels. To circumvent this issue, all pixels with a normalized Doppler bandwidth

exceeding 0.9 were sifted from the resulting CFIs. In replacement, those sifted pixels were filled with a post-hoc inpainting algorithm that consisted of averaging the nearest 8 pixels (fourth and fifth stages of Figure 3.4b). Finally, a 2D median filter with a 5×5 kernel was applied to eliminate spurious values in the resulting CFI (last stage in Figure 3.4b). Note that the size of this median filter kernel was chosen empirically based on in-house testing to adequately remove spurious CFI values without distorting the flow profile.

3.3 Experimental Methods

3.3.1 Data Collection

Table 3.1
Imaging Parameters Used in Data Collection.

Parameter	Value
Imaging Platform	
Number of Tx/Rx channels	192
Array pitch	0.245 mm
Pre-beamformed data sampling rate	25 MHz
Pre-beamformed data resolution	12 bits
Data Acquisition	
Imaging frequency	5 MHz
Tx pulse duration (number of pulses)	3
Transmit steering angle	-10°
Pulse repetition frequency	1500 Hz
Data acquisition duration	3 s
Beamforming	
Pre-beamformed data filter passband	3-7 MHz
Filter design method	Equiripple (30 th Order)
Receive F-number	1.0
Receive steering angle	-10°
Pixel spacing (axial and lateral)	0.1 mm
Image dimension (number of pixels)	400 (axial) $\times$ 368 (lateral)
Image dimension (mm)	40.0 $\times$ 36.8

To facilitate performance investigation on the CNN-based dealiasing framework, *in vivo* datasets were acquired from ultrasound long-axis acquisitions of the femoral bifurcation, which was chosen for its non-trivial geometry and triphasic flow characteristics with large variations in flow velocity

over a cardiac cycle (Hwang, 2017). Femoral CFI is generally susceptible to aliasing in the deep femoral artery branch, whose sloped vessel geometry naturally gives rise to smaller beam-to-flow angles that favor the generation of higher Doppler frequencies according to the Doppler equation. The femoral bifurcations from five healthy volunteers were imaged (with their written consent). To increase the variability of the data, the femoral bifurcation from each participant was also subjected to varying degrees of downstream pressure using a cuff set between 0 mmHg to 100 mmHg at increments of 20 mmHg. The data was collected as part of a femoral bifurcation hemodynamics study protocol as approved by the human research ethics committee at the University of Waterloo. Each data acquisition trial was performed over an entire cardiac cycle, so it included systolic and diastolic flow as well as flow reversal and recirculation.

Data acquisition was performed using an open-platform ultrasound research scanner (US4US; Warsaw, Poland) that was equipped with a linear array transducer (SL1543; Esaote, Genoa, Italy). The scanner was configured to perform plane wave data acquisition, similar to what was described elsewhere (Yiu et al., 2019). Note that, instead of conducting conventional scanline-based imaging, plane wave data acquisition was performed in this work because of two reasons related to its high-frame-rate imaging capability. First, it allows for time-resolved tracking of femoral flow dynamics with significant spatiotemporal variations over different phases of a cardiac cycle. Second, it benefits the efficacy of our deep learning framework by providing a diverse range of training data input that spans time instants at both flow systole and diastole as well as ones with antegrade and retrograde flow. Accordingly, raw channel-domain ultrasound datasets of plane wave data acquisition were collected for offline processing. Table 3.1 summarized the imaging parameters used in this investigation. Note that a -10° steering angle was applied during each plane wave pulsing event. Also, the pulse repetition frequency (PRF) was set to 1500 Hz, in line with what is used in clinical femoral CFI (Hwang, 2017). The use of such a low PRF was necessary to consistently track slow flow emerging during the flow transition phases of the cardiac cycle, thereby reducing color dropouts in the CFI cine-loops. Both single and double aliasing artifacts were observed in the resulting CFI frames.

3.3.2 Signal Processing

The raw datasets collected from the scanner were processed in Matlab (ver. 2016a; MathWorks, Natick, MA, USA) to generate the five training features described in Section 3.2.2. The processing platform was a computer server (SYS-4028-TRT; Super Micro, San Jose, CA, USA) that was

equipped with a Xeon E5-2620 central processing unit (Intel, Santa Clara, CA, USA) and an RTX-2080 graphical processing unit (GPU) (NVidia, Santa Clara, CA, USA). Using Matlab, beamforming was performed on this platform via a GPU-based delay-and-sum algorithm (Yiu et al., 2011), with Gaussian apodization window and dynamic receive focusing. The key processing parameters are listed in Table 3.1. Note that the receive aperture size (i.e. number of channels used to form each pixel) was automatically adjusted within GPU beamformer to consistently achieve an F-number of 1.0 for all pixel positions, thus yielding a lateral beam width of 0.68 mm (full-width at half-maximum value; derived from in-house field simulations). The receive aperture's lateral position was automatically centered about the corresponding pixel in all instances. Also, a steering angle of -10° was applied during this beamforming process to avoid instances of 90° beam-to-flow angle at different pixel positions.

From the beamformed data frames generated, slow-time clutter filtering was performed at each pixel position using an equiripple infinite impulse response filter with a cutoff of 0.1 normalized frequency (i.e. 150 Hz). Subsequently, the Doppler frequency, Doppler power, and Doppler bandwidth were computed at each pixel position using an ensemble size of 48 and the formulas in Section 3.2.2. Also, optical flow analysis (using the Lucas-Kanade algorithm) was performed on the log-compressed, clutter-filtered data frames to determine the flow speckle speed and angle at each pixel position. For this analysis, data for flow speckle speed and angle were averaged over 32 frames to reduce noise. For CNN training, the computed values for all these features were normalized with respect to the largest magnitude value of each map frame.

3.3.3 Data Organization for CNN Training

In total, 8 acquisitions were performed. Among all data frames, 284 CFI frame samples were generated for CNN training. The data feature frames used for the training of CNN #1 (for general identification of aliasing, including both single and double cases) were augmented to improve generalizability. Specifically, as illustrated in Figure 3.3b, the data feature frames were flipped about their axial axis to introduce flow conditions going in the opposite lateral direction; moreover, for Doppler frequency frames and optical flow angle frames, their additive inverse forms were derived to change the assumed flow direction. In total, the augmentations resulted in a final training set of 1136 frames (i.e. fourfold of the 284 original CFI frame samples). It is worth noting that the number of pixels affected by double aliasing constituted only a small portion of the entire dataset. As such, for

CNN #2 (intended to solely identify double aliasing), another training set was produced by excluding all frames without double aliasing. In turn, 476 frame samples were used to train CNN #2.

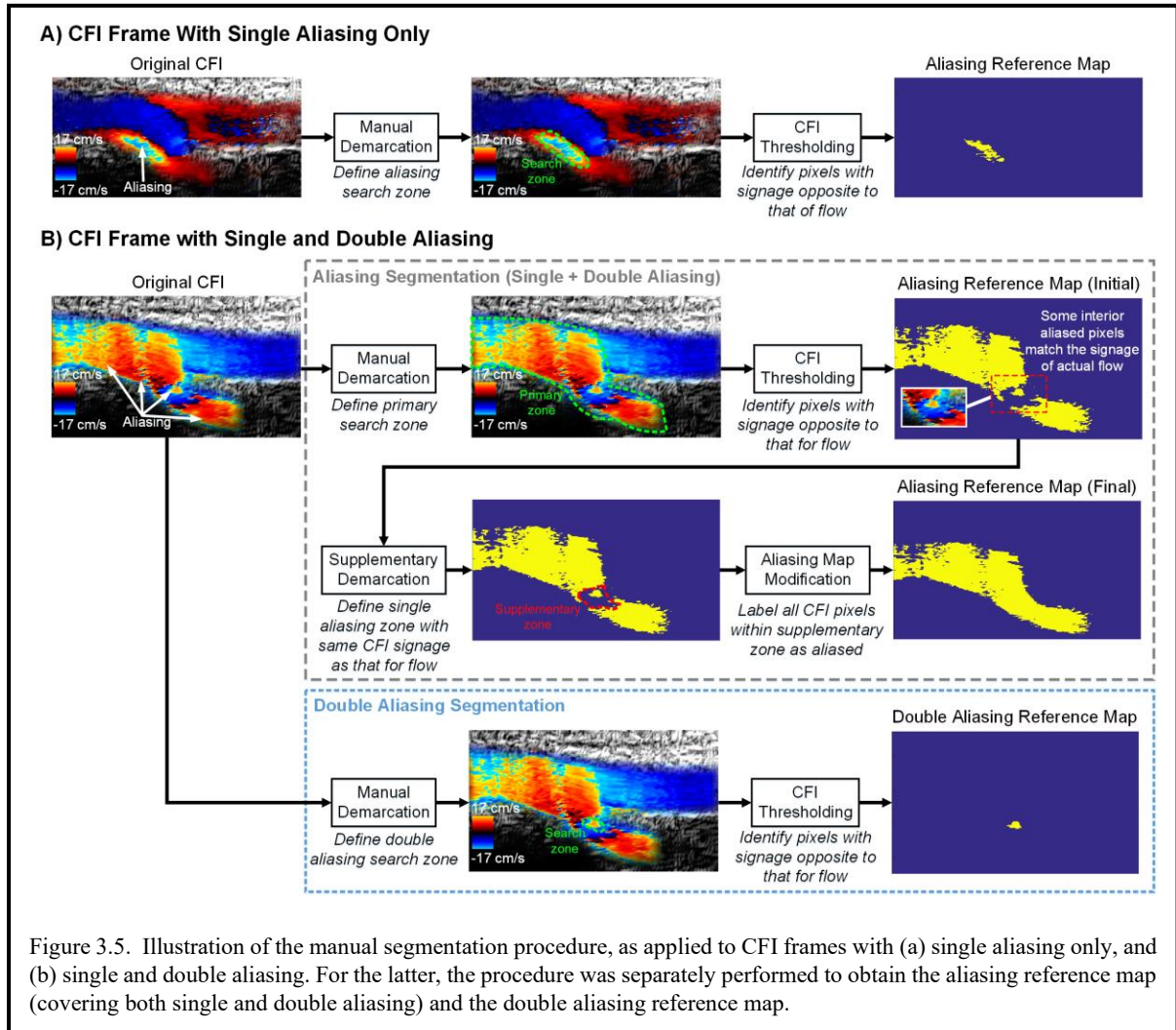


Figure 3.5. Illustration of the manual segmentation procedure, as applied to CFI frames with (a) single aliasing only, and (b) single and double aliasing. For the latter, the procedure was separately performed to obtain the aliasing reference map (covering both single and double aliasing) and the double aliasing reference map.

For each frame of training data, aliased regions were manually segmented to provide the CNN with information on where aliasing appeared in the image view. This manual segmentation task was performed using built-in Matlab functions. As shown in Figure 3.5a, it first involved, by inspection, manual demarcation of the aliased zone as a polygon based on the criterion that aliased CFI pixels were manifested as an anomalous color hue change (i.e. sudden change from red to blue, or vice versa). After that, an aliasing reference map was formed by finding, within the demarcated polygon, the CFI pixels with Doppler frequency signage that was opposite of that for the expected flow direction. For aliased frames with both single and double aliasing, manual segmentation was based on

a multi-step process. As illustrated in Figure 3.5b, it first identified a primary zone and searched for aliased CFI pixels in the same way as the simpler case shown in Figure 3.5a. Subsequently, the aliasing reference map was refined by performing another round of manual demarcation that delineated the boundaries of single aliasing regions with CFI pixels whose color was the same as that for flow. This supplementary demarcation was achieved based on manual identification of CFI map islands that were surrounded by single-aliased pixels with anomalous color hue change. All pixels within this supplementary zone were then labeled as aliased in the final aliasing reference map. Note that, following a similar manual segmentation procedure, double aliasing zones were separately identified on each CFI frame based on the principle that each of these zones would be engulfed within a single-aliased region that shared the same CFI color with flow.

3.3.4 Training Implementation

To facilitate CNN training and inference, Python (ver. 3.6.7) scripts were written for data fetching and for the execution of CNN-related operations on the computing server's RTX-2080 GPU. These scripts included calls to the Keras application programming interface (API) (ver. 2.1.6) that served to: 1) define the U-net CNN architecture parameters, and 2) invoke CNN computation operations on the RTX-2080 GPU via implicit calls to the TensorFlow-GPU (ver. 1.12) software back-end. Each CNN was trained by optimizing the binary cross-entropy loss between prediction and the reference outcome derived from manual segmentation. Such optimization was achieved using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 10^{-4} . To prevent overfitting, an early stopping mechanism was implemented such that training was terminated when loss on the validation set did not improve for 5 epochs. Moreover, only the model with the lowest validation loss was saved from each training procedure. Memory constraints in the GPU hardware (with only 8 GB of random-access memory) necessitated a batch size of 1. To account for the variability of initialization and data sampling, 7 models (i.e. 7 CNN training trials) with different random starting conditions were trained for each training feature combination, and their performance distribution was analyzed.

3.3.5 Testing and Segmentation Evaluation

With the trained CNNs, inferences on aliasing detection were made on 100 test frames that were acquired on a different subject. Phase unwrapping was performed using the algorithm described in Section 3.2.5 to recover the expected flow profile. The prediction results were compared against references derived from manual segmentation of aliased regions (realized using the same procedure

presented in Section 3.3.3). The efficacy of a CNN in identifying aliased pixel positions was evaluated using precision, recall, and intersection over union (IoU) metrics, which are respectively defined as follows:

$$Precision = \frac{|y \cap \hat{y}|}{|\hat{y}|} \quad (3.4)$$

$$Recall = \frac{|y \cap \hat{y}|}{|y|} \quad (3.5)$$

$$IoU = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} \quad (3.6)$$

where $|\cdot|$ denotes the size of the set, y is the reference segmentation map, as obtained from manual segmentation, and $\hat{y}$ is the predicted segmentation for aliasing.

An adjunct CNN training and inference experiment was performed to gain cursory insight into the effects of reduced training data size on the CNN's aliasing detection performance. This adjunct experiment was conducted in a subset of 159 CFI frames formed from 5 of the 8 human volunteer acquisitions. It made use of the same training data augmentation strategy and CNN implementation specifics as that described in Sections 3.3.3 and 3.3.4. The corresponding precision, recall, and IoU were derived for result analysis.

3.3.6 Dealiasing Performance Evaluation

The dealiased CFI frames obtained from the CNN-based framework were compared with the ones obtained from manual dealiasing, which involved manual segmentation of aliased pixel positions (as explained in Section 3.3.3) and the use of the same dealiasing algorithm presented in Section 3.2.5. Their difference between was quantified using the root-mean-squared difference (RMSD) metric, which is defined as:

$$RMSD = \sqrt{\frac{1}{C} \sum_{i=1}^C [x_{i(CNN)} - x_{i(manual)}]^2} \quad (3.7)$$

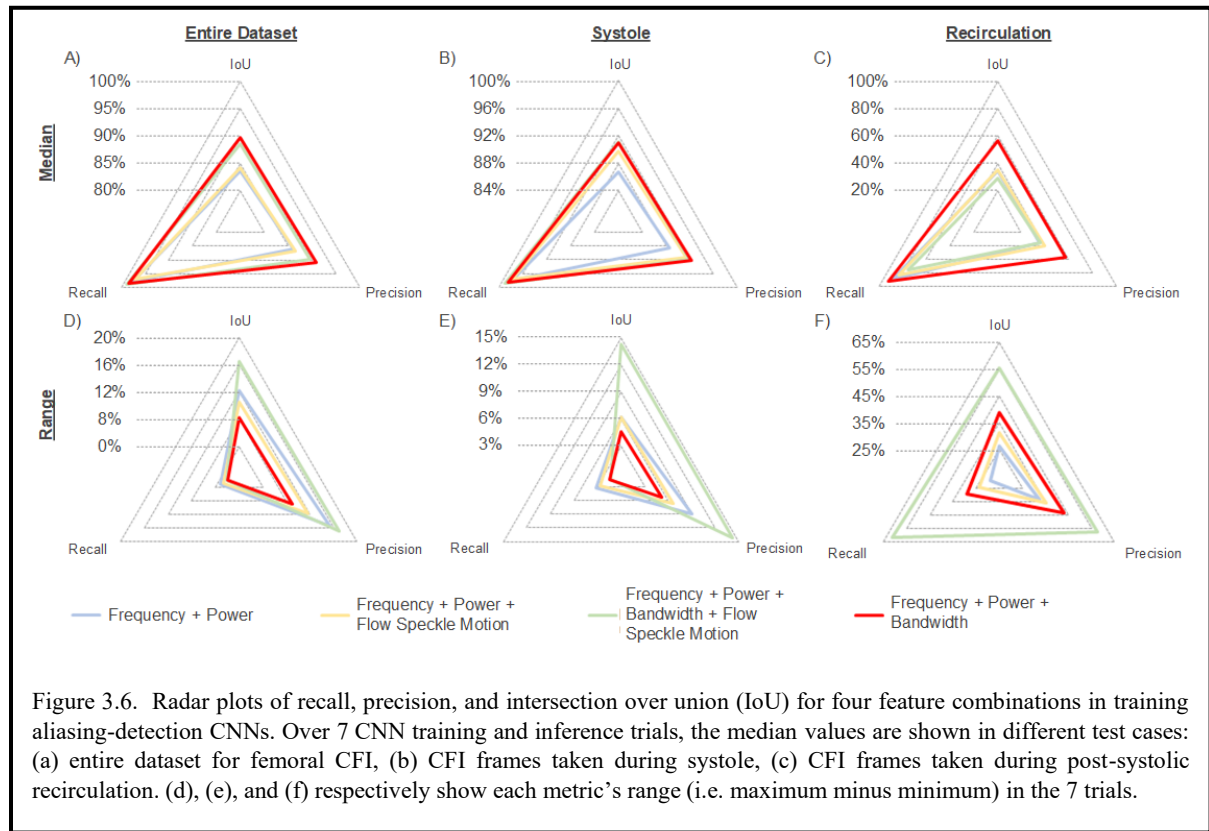
where C is the number of pixels being evaluated, while $x_{i(CNN)}$ and $x_{i(manual)}$ are respective the i^{th} pixel value in the CNN-based dealiased CFI and in the manually dealiased CFI. The results were normalized with respect to the post-correction, extended Doppler frequency limit (i.e. $5f_{max}$).

3.4 Results

3.4.1 CNNs Trained with Frequency, Power, and Bandwidth Features Yielded the Best Aliasing Detection Performance

For aliasing-detection CNNs, all candidate combinations of training features being tested (as presented in Section 3.2.3) had converged in training, and they yielded CNNs with varying degrees of success in identifying aliasing (whether it is single or double aliasing) at different pixel positions.

Figure 3.6 summarizes the aliasing segmentation performance for the CNN of each training feature combination by plotting the median and range (i.e. maximum minus minimum) of its precision, recall, and IoU derived from 7 CNN training trials. Results are shown for different test cases: over the entire test set (left column), for CFI frames taken during systole (middle column), and for CFI frames acquired in the presence of flow recirculation (right column).



As shown in Figure 3.6, the CNN exhibited the best aliasing detection performance when it was trained using the features of Doppler frequency, Doppler power, and Doppler bandwidth (red line in the plots). This training feature combination yielded the highest median recall, precision, and IoU for

the 7 CNN training trials in all test cases. Indeed, for the global test case, the recall, precision, and IoU were respectively 98.5%, 90.5%, and 89.7%. In comparison, for an adjunct inference experiment conducted with a CNN trained using frequency, power, and bandwidth features derived from a reduced dataset of 159 CFI training frame samples (versus 284 in the main experiment), the global test case had achieved a recall, precision, and IoU of 88.5%, 85.9%, and 88.2% respectively (data not shown).

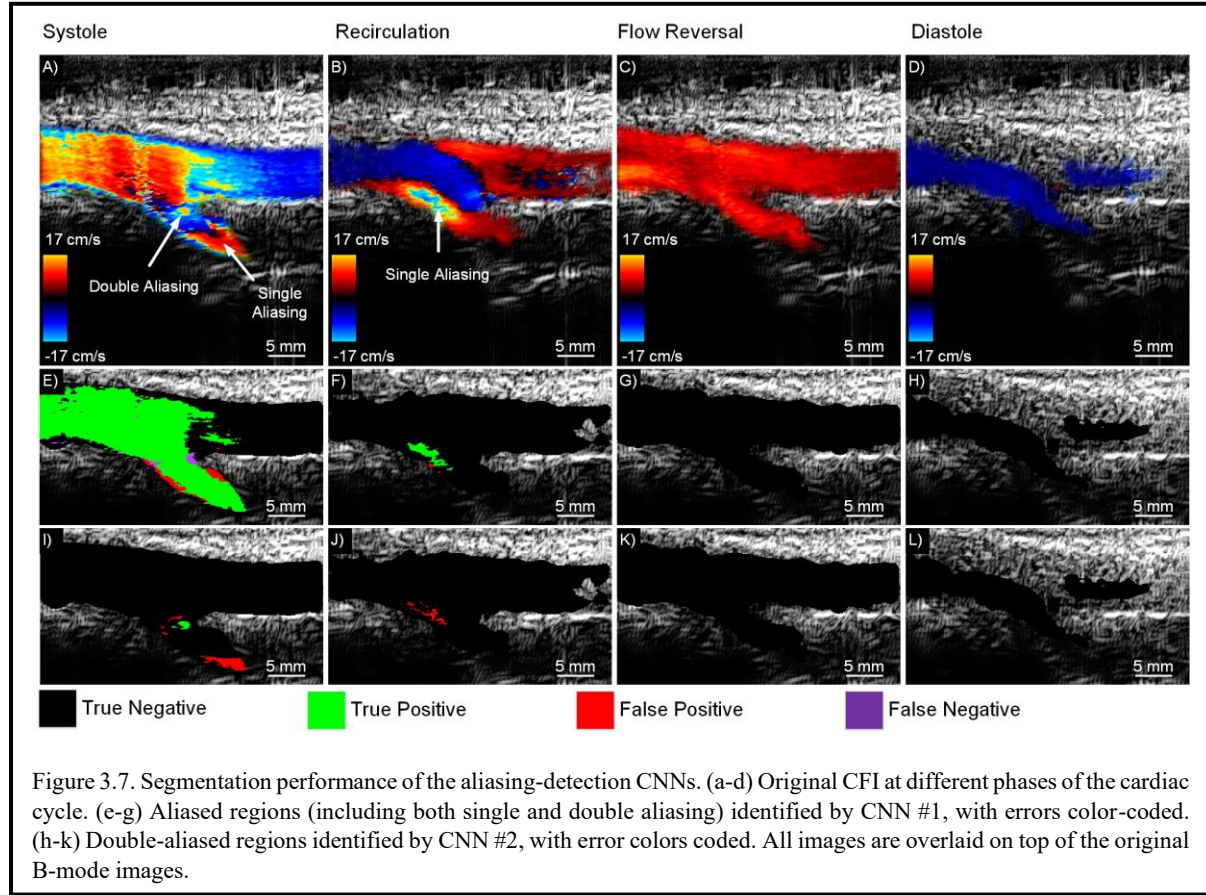
Another observation from Figure 3.6 is that the CNN of the frequency-power-bandwidth training combination generally yielded the smallest range for the three metrics in the global test case and the systolic frameset test case. While this CNN yielded a mediocre median precision and IoU of just above 50% in the test case involving CFI frames with flow recirculation, it was still significantly higher than those for the CNN of other training combinations by at least 15%. It is worth noting that, for the CNN of the frequency-power-bandwidth training combination, the inference time required to generate one frame of aliasing segmentation result was found to be 44 ms when the inference procedure was executed on the RTX-2080 GPU.

3.4.2 Aliasing-Detection CNN Can Accurately Identify Most of the Aliased Pixels

Using the three-feature training combination involving frequency, power, and bandwidth, the trained CNN was found to be potent in correctly identifying the majority of aliased pixels in the image view over different phases of a cardiac cycle. Figure 3.7 shows the corresponding results for four representative timepoints (systole, post-diastolic recirculation, flow reversal, diastole). For the CFI frames shown on the first row of Figure 3.7, the bottom two rows show maps of the detection performance for CNN #1 (for general aliasing identification, including both single and double aliasing cases) and CNN #2 (for double aliasing detection only). Four color-coded detection outcome regions are rendered in each map: true positive (green), true negative (black), false positive (red), and false negative (purple).

As can be observed from Figure 3.7, CNN #1 exhibited a strong performance in correctly identifying aliased pixels throughout the cardiac cycle. Specifically, it was able to identify most of the aliased pixels in the systole frame and in the CFI frame with post-systolic circulation. There were a limited number of false positive and false negative pixels near the tissue-flow boundary. In the flow reversal and diastole CFI frames, this CNN yielded true negative detection results with no detection error. On the other hand, for CNN #2, whose role was to solely detect double aliasing cases, it was able to correctly identify the double aliasing zone in the systole CFI frame (green zone in Figure

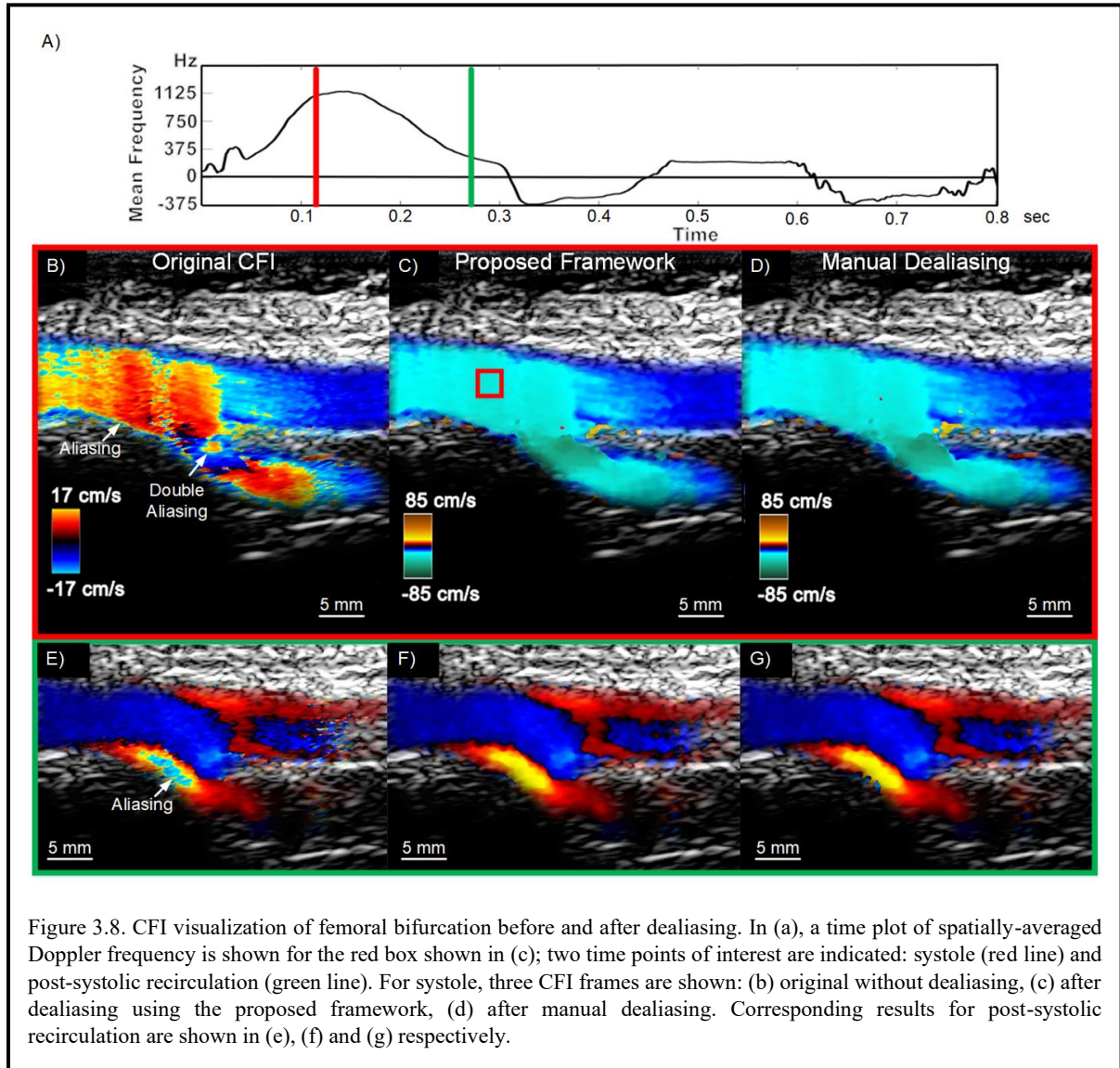
3.7h). Nevertheless, it also yielded some false positive instances in the form of scattered islands in certain CFI frames (red zones in Figure 3.7h and Figure 3.7i).



3.4.3 Visualization of Dealiased CFI Frames After Adaptive Phase Unwrapping

Based on the CNN-identified aliasing regions, adaptive phase unwrapping was found to be capable of resolving single and double aliasing artifacts in the CFI frames taken during systole and in the presence of recirculation. As an illustration, Movie 3.1 shows a cineloop (playback rate: 30 fps) of dealiased CFI over the systolic phase of the cardiac cycle in the femoral bifurcation. Shown in this movie is a side-by-side comparison between the original CFI (left column), the dealiased CFI (middle column), and the CNN-derived aliasing segmentation map (right column). Representative temporal snapshots of dealiased CFI are shown in Figure 3.8, specifically for frames taken during systole (red box), post-systolic recirculation (green box), flow reversal (blue box), and diastole (yellow box). Note that the velocity scale bars in and Figure 3.8, as derived from the Doppler equation assuming 0° beam-to-flow angle, are not the same for the original and the dealiased CFI because of their

difference in Doppler frequency range (see Section 3.2.5). Also, for reference, a time plot of post-correction mean Doppler frequency (spatial-averaged over the red box within the dealiased CFI) is shown in the top row of Movie 3.1 and Figure 3.8.



Three major observations should be noted in the dealiased CFI results. First, after resolving the aliasing artifacts, antegrade flow during systole can be consistently visualized in the dealiased CFI (red box; Figure 3.8). Second, during post-systolic recirculation, the dealiased CFI yielded a more consistent depiction of the flow recirculation zone as compared to the unaliased CFI (green box; Figure 3.8). Third, when comparing with the segmentation results in Figure 3.7, the extraneous pixels

that were misidentified as double aliasing did not significantly affect the visualization performance of the dealiased CFI, because connected bodies of misidentified pixels were not phase unwrapped as they did not exhibit a significant difference with respect to their neighborhood.

For the dealiased CFI frames, their RMSD relative to the reference based on manual segmentation and dealiasing are shown in Table 3.2. In all test cases (global, systole, and recirculation), NRMSD values were under 2.51%, indicating that the CNN-based dealiasing approach yielded similar CFI quality as the manual approach.

Table 3.2
RMSD Between the CNN-based Dealiased CFI and the Manually Dealiased CFI.

Test case type	RMSD (%)
Global	1.61
Systole	2.51
Recirculation	1.16

3.5 Discussion

3.5.1 Summary of Contributions

In recent years, the application potential of deep learning in medical imaging is serendipitously becoming established thanks to concurrent advances in high-throughput computing hardware such as the PU. Harnessing such emerging potency, this work represents the first attempt to address the aliasing problem in ultrasound CFI scenarios *in vivo* using a deep learning approach. In particular, we have devised and trained CNNs to identify and segment aliasing artifacts in CFI using multiple data features derived from different steps of the CFI data processing pipeline (Figure 3.2). The CNNs were trained based on a U-net architecture (Figure 3.3), and inferences drawn from the trained CNNs were used to correct for aliasing errors (both single and double aliasing cases) through an adaptive phase unwrapping strategy (Figure 3.4). Our *in vivo* femoral bifurcation experiments showed that the new dealiasing framework was able to detect aliased CFI pixels (Figure 3.7) and, in turn, resolve them to generate unaliased CFIs with Doppler frequency dynamic range that was five times wider than the conventional range (Movie 3.1 and Figure 3.8).

Compared to existing CFI dealiasing methods (Muth et al., 2011; Shahin et al., 2000), our CNN-based segmentation approach made no assumptions on the size of aliasing region(s) in the image view

and incorporated multiple CFI data features to achieve more consistent aliasing detection performance. In doing so, our framework was able to identify single and double aliasing zones even in CFI frames taken where the majority of flow region was aliased (Figure 3.7; top row). In addition, our framework did not falsely identify CFI regions with arbitrary discontinuities as aliased pixels (such as those emerging from recirculation zones). As well, it was able to extrapolate aliasing segmentation to regions where its boundaries were ill-defined.

It is worth reiterating that aliasing occurs in CFI whenever the PRF is lower than the Nyquist limit, such that the actual Doppler frequency is greater than the maximum unaliased frequency. On this issue, our CNN-based dealiasing framework has served well to relax the Nyquist constraint in CFI by identifying and correcting CFI pixels with aliased Doppler frequency estimates such that they would not disrupt CFI's flow map appearance. In achieving so, CFI can ascertain itself as a flow mapping modality that is resilient against aliasing artifacts even when a relatively low PRF is used. This notion has been demonstrated in our femoral imaging experiments. Bolstering CFI's resilience against aliasing would also be of practical significance to other flow mapping scenarios *in vivo* (to be elaborated in Section 3.5.3).

3.5.2 Insights on CNN Training for Aliasing Detection

As part of this investigation, different combinations of data features were used to train CNNs for aliasing detection. It was found that the combined use of Doppler frequency, Doppler power, and Doppler bandwidth led to the best aliasing detection performance as quantified from the metrics of recall, precision, and IoU (Figure 3.6). Note that, as substantiated by the strong performance metrics for this training feature combination, the size of training datasets used for CNN development can generally be regarded as adequate in our experiments. Even if the CNN was trained with 56% of the original dataset size, the resulting aliasing segmentation performance was only reduced slightly as mentioned in Section 3.4.1.

Doppler frequency and power were obvious to include as training features given their respective roles as the origin of CFI aliasing and the primary flow indicator. In contrast, Doppler bandwidth's utility in aliasing segmentation can be rationalized by noting that Doppler spectral contents exceeding the Nyquist limit were unpredictably rendered as aliased spectral contents in the sampled spectrum, thus adding variability to the estimated Doppler bandwidth. In turn, these spurious bandwidth estimates manifested themselves as spatial outliers over a 2D frame of bandwidth estimates, so they served well as complementary training input for the CNNs.

Flow speckle motion, as derived from optical flow analysis, was considered as a candidate data feature for CNN training. Nonetheless, it did not seem to improve aliasing detection. The reason is likely because speckle tracking is prone to be inconsistent in scenarios with low signal-to-noise ratios as was the case in flow estimation (Bohs, Geiman, Anderson, Gebhart, & Trahey, 2000). If flow speckle motion is to become a practically useful training feature for CNN-based aliasing detection, more advanced speckle tracking algorithms probably need to be implemented (Fadnes, Ekroll, Nyrnes, Torp, & Lovstakken, 2014).

3.5.3 Applications of Interest

In practice, there are at least three scenarios *in vivo* where aliasing is inevitably prone to occur. The first scenario is when a relatively low PRF is used to track slow flow in a pulsatile flow cycle with fast systolic flow. For instance, we used a PRF of 1,500 Hz for femoral bifurcation flow imaging. The use of this relatively low PRF value, which was in accordance with that used clinically in femoral CFI (Hwang, 2017), was necessary to capture slow but dynamic flow patterns that occurred during the flow phase transitions of the artery's triphasic pulsatile flow cycle. However, as a result of using this low PRF value, aliasing unavoidably emerged during systole with fast flow velocities. After performing dealiasing with our algorithm, the resulting CFI flow visualization became more visually intuitive, as rendered in Movie 3.1 and Figure 3.8.

The second applicable scenario arises when a relatively low PRF is needed to monitor fast flow dynamics away from the surface. As known in pulse-echo imaging, the maximum PRF that can be used is inversely proportional to the maximum intended imaging depth (since $f_{PRF} \leq c_0/2z_{max}$). Because the choice of PRF is limited by the depth resolvability constraint, aliasing would be inevitable when flow is relatively fast. Accordingly, an effective dealiasing algorithm is needed to resolve aliasing artifacts originating from pixels with fast flow. Practical examples of this imaging scenario include: (i) intraventricular flow imaging (Mehregan et al., 2014), and (ii) trans-perineum urinary flow imaging (Ishii, Yiu, & Yu, 2017). In these cases, flow speeds can be well above 2 m/s and the beam-to-flow angles may be small; in turn, the actual Doppler frequency may be relatively high. Nevertheless, the imaging depth can be up to 20 cm (for cardiac imaging) and 8 cm (for peritoneal urinary flow imaging), thus precluding the use of a high PRF like 10,000 Hz.

The third application scenario of interest is when relatively fast flow speeds emerge in the CFI image view. A prime example of this application scenario is the high-speed flow jet that arises at a carotid artery stenosis site. These flow jets can readily yield a peak systolic velocity of over 3 m/s for

stenotic sites whose narrowing level is 70% or higher (von Reutern et al., 2012). In such a peripheral flow mapping scenario, CFI may be performed with a relatively high PRF of 10,000 Hz. However, aliasing is still prone to occur since, according to the Doppler equation, the actual Doppler frequency tends to be high (and greater than the maximum unaliased frequency) due to the fast flow speed.

Aside from CFI applications, the CNN-based dealiasing algorithm may be applied to vector Doppler estimation. Aliasing is an inherent concern in vector Doppler estimation because the use of multiple transmit angles necessitates the use of lower PRF values for Doppler estimation at each angle (Yiu & Yu, 2016). To tackle the aliasing problem in vector Doppler, different CNNs can be trained for each angle to remove aliasing from that angle. Note that the segmentation maps obtained here may be used with more advanced aliasing resistant algorithms (Ekroll et al., 2016) suited for vector flow imaging to minimize their errors.

3.5.4 Limitations of Investigation

A few caveats should be acknowledged about the current body of work. First, when assessing the aliasing detection performance of our CNN-based framework, we have primarily made reference to manually identified aliased CFI regions. It would be beneficial in the future to more comprehensively validate the efficacy of our proposed algorithm by comparing its dealiased CFI maps with flow profiles obtained from another intraluminal flow measurement technique, such as magnetic resonance angiography.

Another limitation to be noted is that, since the application focus of this work was exclusively on femoral CFI, we have yet to evaluate the efficacy of CNN-based dealiasing in other CFI scenarios, such as the ones mentioned in Section 3.5.3. To generalize our dealiasing algorithm in other CFI scenarios, the CNNs should be retrained to include sample input from these imaging scenarios. Along this line of investigation, it would also be worthwhile to verify the robustness of the new dealiasing framework for different training dataset sizes and when applying it to CFI data acquired using different imaging parameters.

3.6 Conclusion

Aliasing has been a long-standing nuisance in CFI. With the use of our CNN-based dealiasing algorithm, CFI pixels suffering from single or double aliasing were identified, and they were resolved through adaptive phase unwrapping to effectively extend the Doppler dynamic range by fivefold. In doing so, CFI flow visualization quality was significantly improved. Through creating properly

trained CNNs for aliasing detection, the proposed framework is expected to help disentangle aliased-induced color mosaics in CFI frames. In turn, it can potentially improve CFI's utility in clinical vascular diagnostics, especially in cases with tortuous vessels with significant temporal variations in flow profile.

Chapter 4

A GPU-Based, Real-Time Dealiasing Framework for Vector Doppler Imaging

4.1 Introduction

High frame-rate vector flow imaging is an emerging ultrasound technique for visualizing complex and transient *in vivo* flow dynamics in real-time (Goddi et al., 2017). One recognized way of realizing this technique is to perform multi-angle Doppler estimation where a set of plane wave pulses is transmitted sequentially from multiple steering angles (Jensen, J. A. et al., 2016b). For such an approach, which is often referred to as vector Doppler, the flow vector at each pixel position in the imaging view is derived via a two-step process. First, for each steering angle, pulsed Doppler estimation is carried out on every pixel position by individually processing the corresponding slow-time ensemble. Second, at each pixel position, the Doppler frequency values derived from different steering angles are used as input to a least squares fitting operation that computes the resulting flow vector from triangulation arguments and multiple instances of the Doppler equation (Jensen, J. A. et al., 2016a).

Although performing vector flow imaging via Doppler principles has demonstrated robust estimation performance (Jensen, J. A. et al., 2016a), it is known to be prone to Doppler aliasing errors where, for any steering angle at a given pixel position, the Doppler frequency estimate may not correspond to the actual value induced by blood flow. The root cause for the emergence of Doppler aliasing is insufficient slow-time sampling, in which the pulse repetition frequency (PRF) used to acquire slow-time ensembles was lower than the Nyquist limit (i.e. two times of the maximum Doppler frequency induced by blood flow) (Pellett et al., 2005). If left unaddressed, aliased Doppler frequency estimates at certain steering angles would ultimately lead to the derivation of spurious flow vectors due to erroneous least squares fitting. In turn, they would significantly distort the appearance of the flow profile rendered in the imaging view (Goddi et al., 2017). Aliasing is particularly problematic in live implementations of high-frame rate vector Doppler. In such situations, the maximum PRF that can be used is constrained by the available data transfer rates on the scanner (Boni et al., 2018) or the use of multiple steering angles for robust flow vector estimation (Yiu & Yu, 2016). Consequently, aliasing is more prone to occur since it is well possible for the maximum detectable flow speed to be lower than the actual flow speeds found in major arteries, especially in

stenosed arteries where flow speeds can exceed 2 m/s (Au et al., 2020; Goddi et al., 2018). Note that the choice of PRF may also be limited by imaging depth constraints inherent in the pulse-echo sensing process. As such, aliasing artifacts tend to emerge more easily when performing vector Doppler mapping of fast blood flow in deep vessels (Au et al., 2019).

Resolving aliasing errors in vector Doppler is after all a challenging task, particularly if it needs to be done in real-time. On this topic, one proposed solution is to apply, for each steering angle, a dealiasing algorithm on the Doppler frequency estimates at every pixel position in the image frame (Lai et al., 1997; Muth et al., 2011; Posada et al., 2016). For vector Doppler, staggered (Madiena et al., 2018; Ricci et al., 2017) and dual-wavelength transmissions (Porée et al., 2021) have been successfully used to achieve aliasing-resistant vector Doppler. Another aliasing correction technique for vector Doppler is to holistically take into account all Doppler frequency estimates from all steering angles (Flynn et al., 2011). Transcending beyond these initial solutions for dealiasing, an extended least squares vector Doppler (ELS-VD) algorithm (Ekroll et al., 2016) was recently introduced as a more robust dealiasing framework for vector Doppler. It works by synergizing least squares fitting (Yiu & Yu, 2016) and speckle tracking (Bohs et al., 2000) principles to effectively identify and resolve aliased Doppler frequency estimates. This solution has demonstrated efficacy in suppressing aliasing artifacts from *in vivo* vector Doppler maps acquired from the carotid bifurcation. However, the algorithm’s reported processing throughput of 5 s per frame (Ekroll et al., 2016) is seemingly inadequate for real-time realization. As ultrasound is typically regarded as a real-time imaging modality that may be used in point-of-care settings (Moore & Copel, 2011), excessive offline processing would hinder clinical translation of vector Doppler as a potent vector flow imaging technique that is resilient against aliasing artifacts.

In this article, we present a new real-time dealiasing framework for vector Doppler estimation performed on high-frame-rate imaging datasets acquired from multiple plane wave steering angles. In devising our framework, we have worked with the guiding proposition that an effective dealiasing algorithm with real-time feasibility can be developed by harnessing the theoretical merit of the ELS-VD method and reformulating its execution using parallel computing principles and graphical processing unit (GPU) processing. We have particularly sought to exploit the multidimensional data independency inherent in the ELS-VD method to devise a single-instruction, multiple threads (SIMT) parallel computing pipeline that can be readily executed with high processing throughput on the GPU that has many compute cores in its hardware. We shall demonstrate our framework’s real-time

performance while reaping the theoretical benefits of ELS-VD to achieve robust vector flow imaging via Doppler principles.

4.2 Technical Principles

The objective of our algorithmic framework is to derive aliasing-resistant vector flow maps with real-time processing throughput. It achieves so by making use of SIMT-based parallel computing principles, which form the essence of GPU programming, to execute ELS-VD's signal processing chain. In brief, the ELS-VD algorithm (Ekroll et al., 2016) works by deriving, for each pixel, a dealiased velocity vector estimate through two computational stages:

- 1) Identify the pen-optimal aliasing correction vector $\mathbf{p}$ that yields the least squares fitting error (Yiu & Yu, 2016) among a system of multi-angle Doppler equations;
- 2) Find the aliasing-free velocity vector $\mathbf{v}_D$ by generating dealiased velocity vector candidates with the pen-optimal aliasing correction vector and then using speckle tracking (Bohs et al., 2000) to find the one with the highest flow speckle similarity.

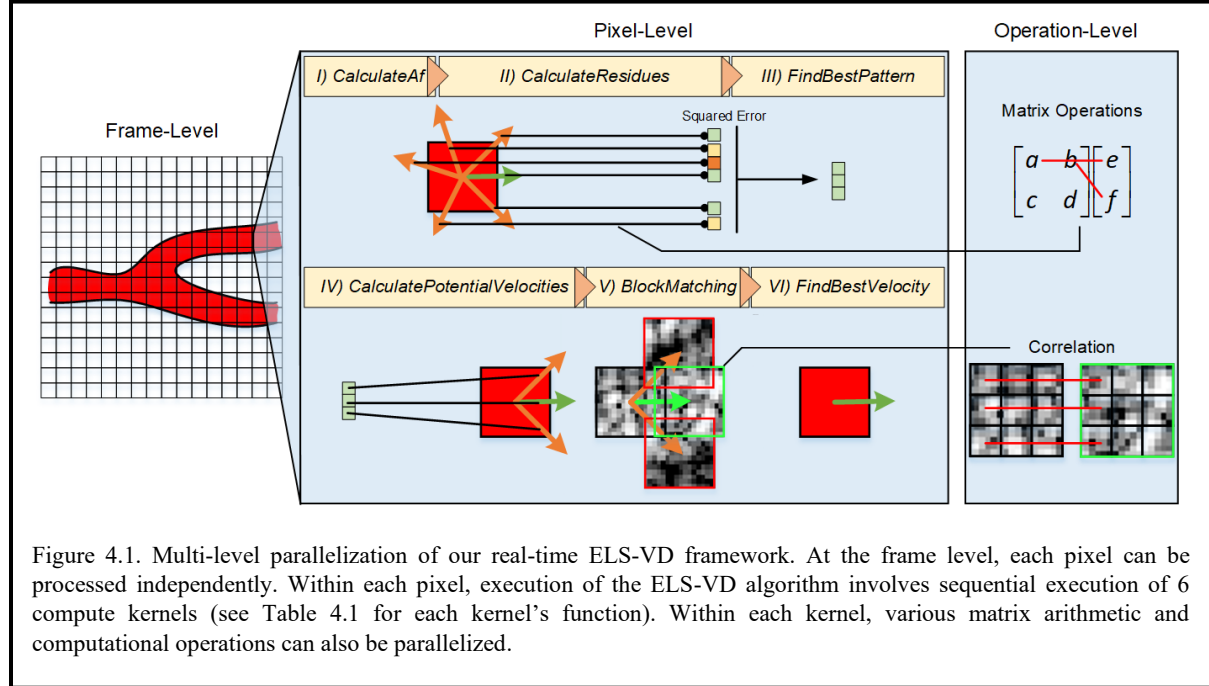
This algorithm can in principle resolve aliasing artifacts with higher aliasing orders (i.e. multiple wraparound cycles), in which for a given aliasing order L (a natural number), the actual Doppler frequency may exceed the maximum unaliased frequency by up to L -fold of the unaliased frequency range. The following subsections shall explain how our algorithmic framework can significantly accelerate the computation of the above algorithm to achieve real-time ELS-VD. Note that our methodological description is not intended to restate all the theory behind the ELS-VD algorithm (Ekroll et al., 2016). Instead, our focus is on explaining the parallelization solution that we have newly devised to realize the notion of real-time ELS-VD.

4.2.1 Key Parameters of Interest

From a computational standpoint, real-time execution of the ELS-VD technique is bottlenecked by a few major parameters. One major influencing parameter is the number of pixels C that requires ELS-VD to be performed in each image frame. Another key parameter is the number of transmit-receive (Tx-Rx) angle pairs N , since a larger N yields more Doppler frequency measurands, each of which may be prone to aliasing that needs to be rectified. A third governing parameter is the aliasing order L , because a larger L naturally creates more instances of aliasing correction vector candidates that need to be analyzed in the ELS-VD algorithm. Fourthly, the parameters used in speckle tracking

analysis would influence the required processing time, particularly when using a larger spatial window dimension B and a larger slow-time window size M for block matching. In our framework, various steps in ELS-VD's processing pipeline have been parallelized to reduce the computational burden imposed by these major influencing parameters.

4.2.2 Overall Parallelization Strategy



We have organized our real-time computing framework in a three-level (frame, pixel, operation) hierarchy as illustrated in Figure 4.1 to parallelize various ELS-VD computing stages. On an image frame level, ELS-VD is concurrently applied to estimate individual pixel position's dealiased flow vector estimate. Such a pixel-by-pixel parallel processing approach is inherently executed without interpixel dependence. Note that frame-based pixelwise parallelization itself is inadequate to achieve real-time processing, as we have discovered in our in-house preliminary tests. It is necessary to further parallelize the computational tasks when processing the set of slow-time ensembles for each pixel.

In our framework, pixel-level parallelization was achieved by exploiting the concurrency of various steps within ELS-VD when processing each pixel. Specifically, it is known that ELS-VD involves two algorithmic stages (Ekroll et al., 2016), in which different instances of the same computational operation are required and some precalculated constants are repeatedly accessed. As such, our

framework has sought to achieve parallelization by allocating multiple threads to handle different instances of calculations and by making frequently accessed data values conveniently accessible to all threads that need them.

The third level of parallelization in our framework pertains to specific computational operations in the algorithm. For instance, the execution of least squares fitting calculations (Yiu & Yu, 2016) involves matrix operations that can be accelerated via parallel computing. Accordingly, these operations have been parallelized through element-wise operations that can run concurrently via a SIMT approach.

With three levels of parallelization, our framework for SIMT has been structured into 6 compute kernels, each of which leverages a hierarchical organization of threads to achieve different parallel computation tasks. The function of each kernel is summarized in Table 4.1. Note that Kernels 1-3 focus on handling the first algorithmic stage of ELS-VD, while Kernels 4-6 address the second stage. Specific parallelization strategies are discussed in the following two subsections.

Table 4.1
Kernels Used in the Framework.

Kernel	Purpose
1. CalculateAf	Performs the matrix multiplication operation $\mathbf{A}^P \hat{\mathbf{f}}$
2. CalculateResidues	Calculates all K instances of the residue r_k
3. FindBestP	Finds the pen-optimal aliasing correction vector $\mathbf{p}$ according to minimum squared residue
4. CalculatePotentialVelocities	Generates the potential velocity candidates from the best pattern
5. BlockMatching	Performs block matching comparisons
6. FindBestVelocity	Determines the final velocity estimation

4.2.3 Stage One: Parallelized Derivation of Pen-Optimal Aliasing Correction Vector

At each pixel, with the set of slow-time ensembles acquired from all N Tx-Rx angle pairs, the first stage of ELS-VD seeks to identify the pen-optimal aliasing correction vector $\mathbf{p}$ with N entries, where each entry indicates the number of frequency unwrapping cycles (within the integer range $-L$ to $+L$) to be applied to the Doppler frequency estimate (normalized to PRF) of a given Tx-Rx angle pair (Ekroll et al., 2016). To perform this task, a set of normalized Doppler frequency measurands $\hat{\mathbf{f}}$ (with N elements) is first computed. Subsequently, aliasing correction vector candidates $\mathbf{p}_k$ are formed, each

of which is one combination of the set of frequency unwrapping cycle values to be applied to all but one Tx-Rx angle pairs (Ekroll et al., 2016). Note that, by deduction, the total number of aliasing correction vector candidates K is equal to $(2L+1)^{N-1}$. For each instance of $\mathbf{p}_k$, its fitting residue (i.e. squared error) r_k in least squares velocity vector estimation is computed as follows for a given speed of sound c_o , ultrasound frequency f_o , and PRF value f_{PRF} :

$$r_k = \left| \mathbf{A}\mathbf{v} - \mu(\hat{\mathbf{f}} + \mathbf{p}_k) \right|_2, \text{ for } \mu = \frac{c_o f_{PRF}}{f_o} \quad (4.1)$$

where $\mathbf{A}$ is the $N \times 2$ angle matrix in the system of Doppler equations from the Tx-Rx angle pairs (Yiu & Yu, 2016), $\mathbf{v}$ is the 2×1 velocity vector estimate at the pixel, $\hat{\mathbf{f}}$ is the $N \times 1$ ensemble of normalized mean Doppler frequencies obtained from all angle pairs, and $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T \mathbf{x}}$ is the l_2 norm ($\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$). The instance of $\mathbf{p}_k$ with the minimum fitting residue is then set as the pen-optimal aliasing correction vector $\mathbf{p}$. In other words:

$$\mathbf{p} = \arg \min_{\mathbf{p}_k} (r_k) \quad (4.2)$$

In pursuit of a parallel implementation of Equations (4.1) and (4.2), we note from linear algebra (Moon & Stirling, 2000) that r_k is essentially equal to the l_2 norm of the product between the orthogonal projection of $\mathbf{A}$ and the dealiased measurand vector $(\hat{\mathbf{f}} + \mathbf{p}_k)$. This operation can be mathematically expressed as:

$$r_k = \left| \mathbf{A}^P \hat{\mathbf{f}} + \mathbf{A}^P \mathbf{p}_k \right|_2 \quad (4.3)$$

where $\mathbf{A}^P$ is the $N \times N$ projection matrix that is equal to $(\mathbf{A}\mathbf{A}^\dagger - \mathbf{I})$, with $\mathbf{A}^\dagger$ and $\mathbf{I}$ respectively being the pseudoinverse of $\mathbf{A}$ ($2 \times N$) and an identity matrix ($N \times N$). Note that, in Equation (4.3), $\mathbf{A}^P$ and $\mathbf{A}^P \mathbf{p}_k$ are algebraic quantities that remain constant for all pixels and can thus be precalculated. In contrast, $\mathbf{A}^P \hat{\mathbf{f}}$ is the only part of the equation that needs to be calculated differently for each pixel. Based on this notion, we have devised three SIMT compute kernels in our framework to handle specific computational tasks, as elaborated in the following.

1) CalculateAf Kernel: We have devised a compute kernel to perform the $\mathbf{A}^P \hat{\mathbf{f}}$ operation for each pixel. This kernel uses N^2 threads per pixel to parallelize the prescribed computation process on an element-wise basis. Specifically, each thread with a unique identification (i,j) is responsible for

calculating the product between one matrix element and the corresponding element in the vector $\hat{\mathbf{f}}$ as follows:

$$\psi_{i,j} = a_{i,j}^P \hat{f}_j \quad (4.4)$$

where $(.)_{i,j}$ denotes the element in the i^{th} row and the j^{th} column in that matrix and $(.)_{i,j}$ denotes the j^{th} element in the vector. Subsequently, N threads are used to form the vector $\mathbf{A}^P \hat{\mathbf{f}}$, whose i^{th} element is the result of the following summation that is handled by the i^{th} thread:

$$(\mathbf{A}^P \hat{\mathbf{f}})_i = \sum_{j=1}^N \psi_{i,j} = \sum_{j=1}^N a_{i,j}^P \hat{f}_j \quad (4.5)$$

2) *CalculateResidues Kernel*: The output from the first kernel is fed as input into a second kernel that aims to compute the fitting residue for all aliasing correction vector candidates. The latter process involves a parallelized solution to Equation (4.3), whereby parallel addition of $\mathbf{A}^P \hat{\mathbf{f}} + \mathbf{A}^P \mathbf{p}_k$ and computation of the resulting l_2 norm are performed for each aliasing correction vector candidate. For this computation, K threads per pixel are used to perform the element-wise vector addition tasks needed to compute the residue for all aliasing correction vector candidates. The k^{th} thread is specifically responsible for calculating the following squared sum:

$$r_k^2 = \sum_{i=1}^N (\mathbf{A}^P \hat{\mathbf{f}} + \mathbf{A}^P \mathbf{p}_k)_i^2 \quad (4.6)$$

Note that, since finding the minimum r_k is equivalent to finding the minimum r_k^2 , the square root operation is omitted in Equation (4.6) to reduce the computational burden.

3) *FindBestP Kernel*: Given the calculated residues, the third kernel then proceeds to search for the pen-optimal aliasing correction vector $\mathbf{p}$ that has the minimum residue (among a set of K candidates). It achieves so by performing tree-based parallel reduction with multiple threads. Specifically, the set of K residue values are first partitioned into equally sized subsets, and each thread is used to search for the minimum of one subset. These local minima are then repartitioned into another cluster of subsets, and parallelized search for subset minima is again performed. This reduction process is repeated until the global minimum is obtained as per Equation (4.2).

4.2.4 Stage Two: Parallelized Computation of Optimal Dealiased Velocity Vector

After obtaining the pen-optimal aliasing correction vector $\mathbf{p}$, the ELS-VD algorithm proceeds to determine the dealiased velocity vector $\mathbf{v}_D$. This task is achieved via multiple steps. First, since $\mathbf{p}$ covers the frequency unwrapping cycle values for all but one Tx-Rx angle pair (Ekroll et al., 2016), a set of $2L+1$ dealiased velocity vector candidates is derived to cover possible aliasing cases (spanning from $-L$ to $+L$) that may arise for the measured Doppler frequency of the remaining Tx-Rx angle pair not covered in the previous ELS-VD algorithmic stage. Each dealiased velocity vector candidate $\mathbf{v}_l$ can be found from the least squares solution to the system of Doppler equations (Yiu & Yu, 2016), as given by:

$$\mathbf{v}_l = \mu \mathbf{A}^\dagger (\hat{\mathbf{f}} + \mathbf{q}_l) \quad (4.7)$$

where $\mathbf{q}_l$ is a candidate for the optimal aliasing correction vector, and it is one instance of the pen-optimal vector $\mathbf{p}$ with a specific frequency unwrapping cycle value (i.e. an integer from $-L$ to L) for the unaccounted Tx-Rx angle pair. In the second step, for each dealiased velocity vector candidate, inter-frame block matching performed to determine the corresponding sum of absolute difference (SAD) value. At last, the velocity vector candidate with the minimum SAD is defined as the aliasing free velocity vector $\mathbf{v}_D$ for that pixel. In our framework, one compute kernel is devised for each of the three steps. Their specific operations are presented as follows.

1) *CalculatePotentialVelocities Kernel*: To derive the set of dealiased velocity vector candidates $\mathbf{v}_l$, this kernel first allocates a block of threads to execute all $2L+1$ instances of Equation (4.7) on an elementwise basis. Given the dimensions of the matrix and vectors in Equation (4.7), $2N$ groups of threads are assigned per pixel, each group of which is responsible for handling the matrix multiplication between one element of the $2 \times N$ matrix $\mathbf{A}^\dagger$ and the corresponding element in the $N \times 1$ vector $(\hat{\mathbf{f}} + \mathbf{q}_l)$. Each thread group is allocated with $2L+1$ threads to parallelize the calculations for all instances of $(\hat{\mathbf{f}} + \mathbf{q}_l)$. Accordingly, the two elements of the i^{th} dealiased velocity vector candidate (i.e. its axial and lateral velocity components) are found to equal to the following sum of elementwise products corresponding to the same matrix row of $\mathbf{A}^\dagger$:

$$(\mathbf{v}_l)_i = \sum_{j=1}^N a_{1,j}^\dagger [\hat{f}_j + (q_l)_j] \quad (4.8)$$

$$(\mathbf{v}_l)_2 = \sum_{j=1}^N a_{2,j}^\dagger [\hat{f}_j + (q_l)_j] \quad (4.9)$$

2) *BlockMatching Kernel*: This kernel takes the $2L+1$ instances of dealiased velocity vector candidates $\mathbf{v}_l$ as input to perform speckle tracking to determine which instance of $\mathbf{v}_l$ yields the highest interframe flow speckle similarity (Ekroll et al., 2016). For each velocity vector candidate, the kernel first calculates its corresponding interframe displacement (equals to $\mathbf{v}_l$ multiplied by inter-frame interval) across a sequence of M consecutive flow speckle frames in a given time window of interest (with M typically smaller than or equal to the slow-time ensemble size used to calculate Doppler frequencies). For each pair of adjacent frames, a block of $B \times B$ pixels centered about the pixel of interest is formed in the former frame, and the corresponding displaced block location is identified in the latter frame. After that, the flow speckle similarity is evaluated between the former frame's $B \times B$ block and the displaced $B \times B$ block in the latter frame. This measure is calculated as SAD of the two blocks, and the cumulative SAD over all $M-1$ adjacent frame pairs is evaluated. Such a sequence of calculations for the entire set of velocity vector candidates is handled in parallel by the kernel via the use of B^2 threads. The $(i,j)^{\text{th}}$ thread is responsible for finding the cumulative SAD of one pixel in the block [denoted as $\Delta_l(i,j)$] over all $M-1$ adjacent frame pairs:

$$\Delta_l(i,j) = \sum_{m=1}^{M-1} \left| s_{D,m+1}^l(i,j) - s_m(i,j) \right| \quad (4.10)$$

where $s_m(i,j)$ denotes the $(i,j)^{\text{th}}$ block pixel in the former flow speckle frame in the m^{th} adjacent frame pair, while $s_{D,m+1}^l(i,j)$ denotes the same block pixel in the latter frame's displaced block for the l^{th} velocity vector candidate $\mathbf{v}_l$. It follows that the SAD for the l^{th} dealiased velocity vector candidate (denoted as Δ_l) is given by:

$$\Delta_l = \sum_{i=1}^B \sum_{j=1}^B \Delta_l(i,j) \quad (4.11)$$

When calculating Equations (4.10) and (4.11), non-flow pixels in the block, as determined via power thresholding akin to power Doppler imaging, are excluded from the SAD computation. Also, tree-based parallel reduction, similar to what has been described in Section 4.2.3.3, is implemented in the kernel to accelerate the addition process in Equation (4.11). Note that, in lieu of SAD, cross-correlation (i.e. sum of squares) may also be used to evaluate inter-frame flow speckle similarity. Yet,

it has not been used in this kernel because it is more computationally intensive due to the involvement of multiplications, and it has been suggested that SAD outperforms sum of squares in block matching (Idzenga et al., 2014).

3) *FindBestVelocity Kernel*: This last kernel in our real-time ELS-VD framework works to identify the optimal velocity vector $\mathbf{v}_D$ that is ideally not subject to aliasing. It achieves so by analyzing the $2L+1$ values of SAD calculated for all dealiased velocity vector candidates. $\mathbf{v}_D$ is then defined as the dealiased velocity vector with the minimum SAD, which corresponds to the highest interframe flow speckle similarity. Computationally, this task is similar to the one in the *FindBestP* kernel (see Section 4.2.3.3). The only difference is that the minimum SAD search process can be executed trivially without the need to implement parallel reduction strategies since L is typically not large.

4.3 Implementation of Real-Time ELS-VD

In this work, the compute unified device architecture (CUDA) platform (NVidia, Santa Clara, CA, USA) was used to implement our parallelized ELS-VD framework. CUDA is an application programming interface (API) developed for general-purpose computing on GPUs, and it relies on a SIMT architecture. The ELS-VD framework was realized as a standalone processing pipeline on a computing platform that was equipped with a Xeon E5-2620 central processing unit (Intel, Santa Clara, CA, USA) and an RTX-2080 GPU (NVidia). Our framework was coded in C++ using the CUDA API (ver. 8.0.60), and it was compiled for use with MATLAB (ver. 2020a; Mathworks Inc., Natick, MA, USA) through the use of the MEX functionality. Its implementation has also adopted different optimization strategies to enhance the framework's efficiency, as described in the following.

4.3.1 General Considerations for Fast Execution

In implementing our real-time ELS-VD framework on the GPU, we have optimized its computational operations for single-floating point precision. Specifically, all variables have been stored and processed as `floats` data type. Also, we have leveraged CUDA's `fast maths` compilation setting whenever possible to execute arithmetic operations such as division and power.

Besides optimizing the arithmetic operations, we have taken into consideration various key features of the GPU when implementing real-time ELS-VD. These features include:

- 1) Each GPU is equipped with thousands of compute cores (4,352 on the RTX-2080 model) to facilitate concurrent execution of multiple threads;

- 2) CUDA-based computing kernels are executed with thread blocks (indexed in 2-D entities) that have access to the GPU's on-chip memory, whose access latency is much shorter than the GPU's off-chip global memory;
- 3) There is a maximum thread block size that is supported by the GPU (1,024 for the RTX-2080 model), beyond which run-time overhead arises and it negatively impacts the computing throughput.
- 4) GPU's on-chip memory is limited in capacity; on the RTX-2080 GPU, between a group of 64 threads, only 96 kB of on-chip memory (64 kB shared memory and 32 kB L1 cache) is available for sharing;
- 5) Texture memory is available on the GPU to cache frequently accessed data in the GPU's global memory.

The following subsections shall describe how we have carefully organized the kernel's thread blocks and operations to make appropriate use of the GPU's available compute resources.

4.3.2 Fast Memory Access Based on the Use of One Thread Block Per Pixel

Various kernels in our real-time ELS-VD framework have suitably leveraged the GPU's fast on-chip memory by executing their SIMT computations on a one thread block per pixel basis. For example, the *CalculateAf* kernel (Section 4.2.3.1) requires N^2 threads for each pixel with N (number of Tx-Rx angle pairs) typically less than 32, so accordingly an $N \times N$ thread block is allocated to execute this kernel for each pixel. In doing so, the matrix elements of $\mathbf{A}^P$ and $\hat{\mathbf{f}}$, which are repeatedly accessed according to Equation (4.4), are copied to the GPU's shared memory during run-time to significantly shorten the resulting data access latency. A similar strategy for thread block organization is used in the *CalculatePotentialVelocities* kernel (Section 4.2.4.1) and the *FindBestVelocity* kernel (Section 4.2.4.3).

4.3.3 Optimization of the CalculateResidues Kernel

For compute kernels that require more processing threads than the maximum block size of 1,024 threads, we have adopted alternative implementation strategies to ensure that only one thread block is allocated to process each pixel. Specifically, for the *CalculateResidues* kernel (Section 4.2.3.2) that in principle uses K threads (where K is typically much larger than 1,024 as per Section 4.2.3), it has adopted a $\min(1024, K)$ thread block structure so that the number of threads per block would not

exceed 1,024. If K was greater than 1,024, the threads were instructed to compute Equation (4.6) for multiple instances of $\mathbf{p}_k$ (with an even distribution of computing load between threads). This implementation strategy was chosen in lieu of initiating more thread blocks so that thread organization can be simplified to reduce the number of read-write operations between the GPU's global memory and on-chip memory.

4.3.4 Optimization of the FindBestP Kernel

For the *FindBestP* kernel (Section 4.2.3.3), depending on the actual values of N and L , it is also prone to use more threads than the maximum block size supported by the GPU. To alleviate this issue, we have frontloaded parts of the minimum search operation to the *CalculateResidues* kernel. In particular, since each thread within *CalculateResidues* is already tasked to calculate r_k^2 for multiple instances of $\mathbf{p}_k$, we have revised its operation further so that it would also directly identify the instance with the minimum r_k^2 value within its subgroup. This strategy is computationally effective, because the values for multiple instances of r_k^2 already exist within the thread block's allocated on-chip memory. In turn, the number of memory access operations is reduced to improve the processing throughput, and the *FindBestP* kernel's parallel reduction task is naturally simplified. Note that, for the *FindBestP* kernel's minimum search among the remaining candidates, we have divided them into subsets of 32, so that at most 32 subsets can be processed by a single block of 1,024 threads. The key benefit of doing so is that the same thread block's on-chip memory can be leveraged to effectively reduce memory access latency of repeated r_k^2 value access requests during the minimum search. Only the final global minimum result would be copied back to the GPU's global memory for further processing.

4.3.5 Optimization of the BlockMatching Kernel

For the *BlockMatching* kernel (Section 4.2.4.2), there is repeated access of flow speckle image values in the current frame when executing Equation (4.10), and some flow speckle pixels are common between the spatial block matching window of adjacent pixels. Accordingly, the kernel has been devised to copy these frequently accessed flow speckle image values to the GPU's texture memory to reduce the data access overhead involved in retrieving data from the GPU's global memory. Note that the number of threads required to compute Equations (4.10) and (4.11) depends on the size of the $B \times B$ spatial window used for block matching. As such, give a maximum thread block size of 1,024,

the window dimension B was limited to 32 or smaller so that a single thread block can be allocated to handle the block matching calculations for one dealiased velocity vector candidate.

4.4 Experimental Methods

4.4.1 Experimental Hardware

Our real-time ELS-VD framework's efficacy, including both its aliasing resistance and processing throughput, was investigated through a series of phantom data processing trials. The experimental hardware consists of an anthropomorphic phantom and a high frame rate imaging platform. The phantom was a wall-less carotid bifurcation model with 50% stenosis, which was chosen because of the prevalence of aliasing in stenosed conditions. It was fabricated using an investment casting protocol reported elsewhere (Ho et al., 2017). Its tissue-mimicking material was composed of a mixture of 10% polyvinyl alcohol (341584; SigmaAldrich, St. Louis, MO, USA), 3% graphite scatterers of particle diameter less than 20 μm (282863; Sigma-Aldrich), and 0.3% potassium sorbate preservatives (85520; Sigma-Aldrich). The phantom was connected to a pulsatile flow pump that generated carotid pulses with a peak flow rate of 5 ml/s and a pulse rate of 72 beats/min. The blood mimicking fluid was matched to the acoustic and viscous properties of human blood. Details on the fabrication of the flow pump and the blood mimicking fluid have been reported elsewhere (Ho et al., 2017).

The imaging platform was a research-purpose ultrasound scanner (SonixTouch; Analogic Corporation, Peabody, MA, USA) with a linear probe L14-5 linear that was configured for plane wave transmission, and a pre-beamformed data acquisition system (Cheung et al., 2012) was used to acquire the channel-domain radiofrequency (RF) data needed for processing. Imaging and system parameters used in this experiment are shown in Table 4.2. In brief, transmission was performed with two steering angles (-10° , $+10^\circ$) and the raw data was then beamformed on receive to form 7 Tx-Rx steering angle combinations. These combinations were chosen for their effectiveness in resolving aliasing artifacts as reported earlier (Ekroll et al., 2016). Offline processing was then performed on the acquired data via a custom MATLAB and CUDA-based pipeline for beamforming (Yiu et al., 2011) and Doppler processing (Yiu & Yu, 2013). Accordingly, Doppler frequency measurands, flow detection maps, and flow speckle maps are derived for each image frame, and they were used as input to analyze the efficacy of our real-time ELS-VD framework.

Table 4.2
Imaging Parameters Used in the Experiments.

Parameter	Value
Imaging Platform	
Number of Tx/Rx channels	128
Array pitch (mm)	0.3048
Data sampling rate (MHz)	40
Data sample resolution (bits)	12
Data Acquisition	
Imaging frequency (MHz)	5
Tx pulse duration (number of pulses)	3
Tx steering angles (°)	−10 and +10
Pulse repetition frequency (Hz)	3333 (for each Tx angle)
Data acquisition duration (s)	2
Beamforming	
RF filter passband (MHz)	2.5–7.5
RF filter order	31
Filter design method	Equiripple
(Tx, Rx) angle pair combinations	(−10°,−10°), (−10°,−3°), (−10°,6°), (−10°,10°), (10°,−6°), (10°,3°), (10°,10°)
Image dimension (cm)	Lateral: −1.9 to 1.9, Axial: 0.0 to 4.0
Image dimension (number of pixels)	256 (lateral) × 256 (axial)
Slow-Time Data Processing	
Clutter filter cut-off (normalized)	0.05
Clutter filter order	75
Filter design method	Equiripple
Slow-time ensemble size (samples)	30 (sliding window processing)
Sliding window step size (samples)	1

4.4.2 Dealiasing Performance Analysis

Using our real-time ELS-VD framework, aliasing-resistant flow vector estimation was performed on a frame-by-frame basis on the GPU. The results were then copied back to the CPU for visualization. Flow vector estimation was also performed using the conventional least squares vector Doppler estimator (Yiu & Yu, 2016) to determine the impact of aliasing on the resulting vector flow images if these artifacts were not resolved. To facilitate quantification of our framework’s efficacy in achieving dealiasing, we have generated a set of reference flow vector maps by performing inspection-based manual dealiasing on the flow vector maps obtained via least squares vector Doppler estimation. The

procedure for manual dealiasing has been presented elsewhere (Nahas et al., 2020). The root-mean-squared difference (RMSD) between the ELS-VD-derived flow vectors and the manually dealiased flow vectors were then calculated as a metric for our framework’s aliasing resistance efficacy, especially for the image frames corresponding to flow systole when the fastest flow velocities emerged.

4.4.3 Throughput Analysis

The throughput of our real-time ELS-VD framework was evaluated in terms of the processing time per frame under different imaging resolutions and different parameters. Each kernel’s execution time was tracked individually using CUDA timers to identify the framework’s computational bottlenecks. The measured time only considered the GPU processing time, while the time for auxiliary operations (e.g. data transfer between CPU and GPU, and CPU functions) was excluded because the practical use of our framework would involve an end-to-end GPU processing pipeline.

We specifically investigated the impact of the following factors on processing time: 1) number of flow pixels being processed C , 2) maximum aliasing order L , 3) block matching spatial dimension B , and 4) slow-time window size M used for block matching. In a practical application, the number of flow pixels processed depends on the imaging resolution and the flow dynamics of the scene, making it an important parameter to investigate. The maximum aliasing order is a function of the imaging scenario, with faster flow conditions requiring higher aliasing order. Moreover, this parameter has a significant impact on the number of aliasing patterns to investigate. The block matching parameters (spatial window size and slow-time window size), which influence the fidelity of the speckle tracking analysis, may well impact the processing time and thus their effects should be analyzed.

4.5 Results

4.5.1 Real-Time Aliasing-Resistant Flow Vector Mapping Achieved in a Stenosed Carotid Bifurcation

Aliasing-resistant flow vector estimation was demonstrated successfully in the stenosed carotid bifurcation phantom. Figure 4.2a, b and c respectively show the peak-systolic flow vector maps derived using the least squares vector Doppler estimator, our real-time ELS-VD framework, and manual dealiasing. The ELS-VD estimation results were obtained for $L=1$, $B=20$, and $M=30$. Note

that, in Figure 4.2a, the white-colored flow vectors indicate ones with aliasing artifacts as determined from the manually dealiased reference. The key observation to be noted from Figure 4.2a is that,

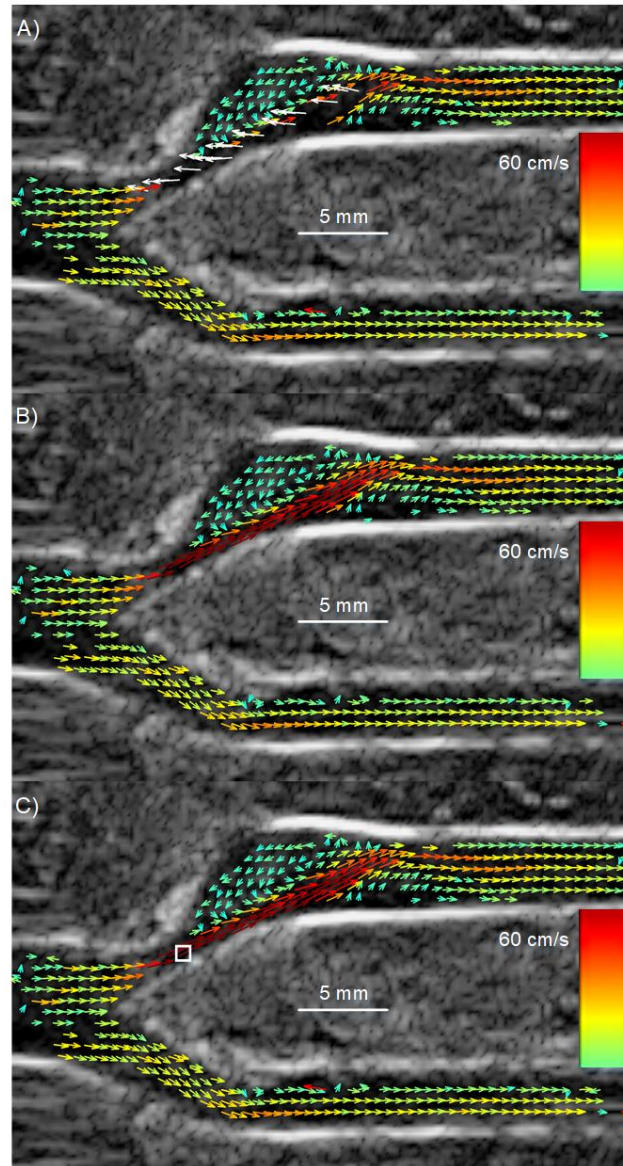
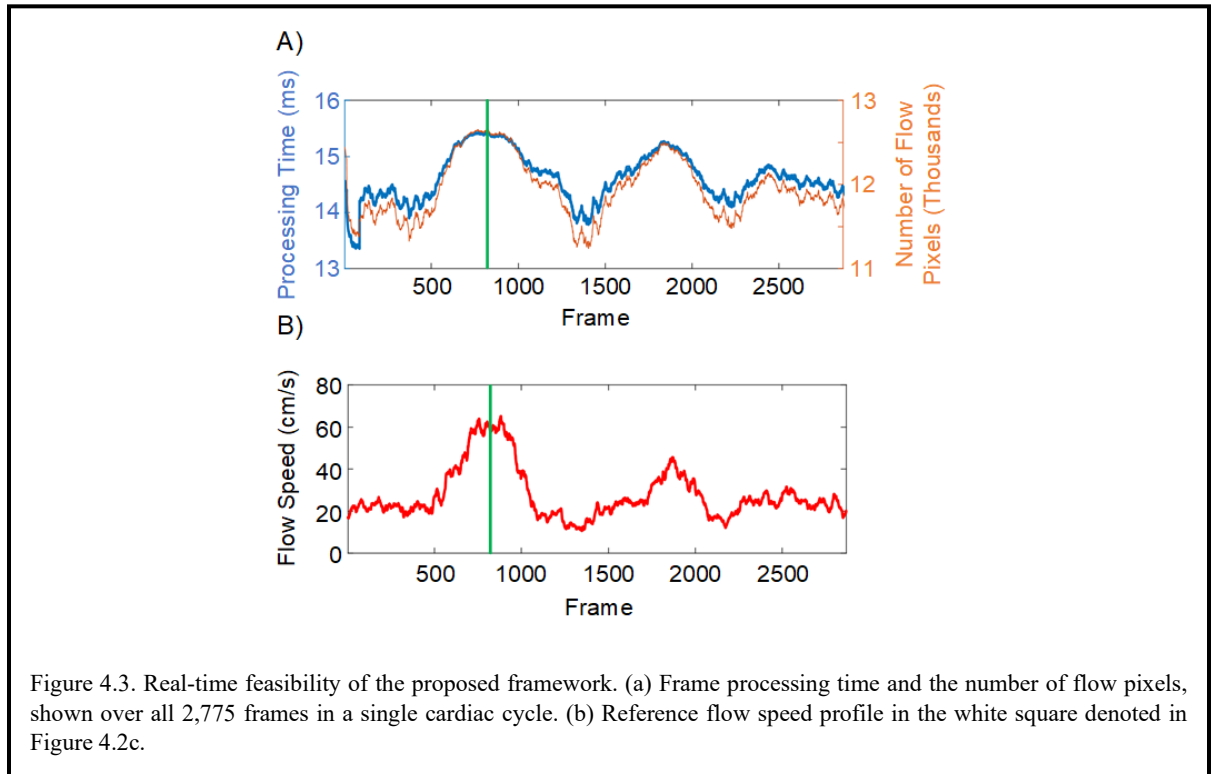


Figure 4.2. Peak-systolic flow vector profiles obtained from a carotid bifurcation phantom with 50% stenosis. Results are derived using: (a) least squares vector Doppler estimation, with aliasing artifacts highlighted in white; (b) the real-time ELS-VD framework; (c) manually dealiasing (shown for reference).

during peak systole, aliasing expectedly emerged in the carotid stenosis region because the narrowed lumen had induced faster flow velocities. These artifacts have been largely resolved by our real-time ELS-VD framework, as shown in Figure 4.2b. Indeed, over the 371 systolic frames where aliasing

artifacts emerged, real-time ELS-VD merely yielded an RMSD of 5 cm/s in the stenotic jet region with respect to the manually dealiased flow vector reference maps that have a systolic jet speed of 60 cm/s (as shown in Figure 4.3b). In contrast, the least squares vector Doppler estimator was found to yield a significantly higher RMSD of 44 cm/s in the stenosis zone.

As a substantiation of the notion of real-time processing, Figure 4.3a plots the frame processing time over one full cardiac cycle. For reference, the mean flow speed profile is also plotted in Figure 4.3b for the white square zone indicated in Figure 4.2c. It can be observed that, regardless of the cardiac cycle phase, the frame processing time was under 16 ms, which corresponds to a minimum processing throughput of 62.5 frames per second (fps). The frame processing time was found to vary slightly over different cardiac cycle phases. During systole, more flow pixels emerged because of flow-mediated dilation of the vessel lumen, and thus ELS-VD needed to be applied to more pixels to detect (and possibly correct for) aliasing artifacts. Nonetheless, real-time processing throughput was still achieved in this scenario.



4.5.2 ELS-VD Processing Throughput Influenced by Multiple Parameters

For practical combinations of processing parameters, our ELS-VD framework was generally found to be capable of achieving real-time processing throughput (with at least a video-range frame rate of 25 fps). Figure 4.4 shows a series of corresponding results. Specifically, Figure 4.4 shows the frame processing time as a function of the number of flow pixels C being processed and the maximum aliasing order L (for $B=20$ and $M=30$). This plot confirms that the frame processing time expectedly increased when there were more flow pixels or with the use of a larger aliasing order. For an aliasing order of 1, our ELS-VD framework can achieve real-time performance for frames with just over 40,000 pixels, or an image grid of 200×200 . In contrast, for an aliasing order of 3, our framework can yield real-time throughput for frames with 8,100 pixels.

As additional insight into the real-time performance of our ELS-VD framework, Figure 4.4b shows the number of pixels that can be processed at a 25 fps throughput under different block matching spatial window sizes and aliasing orders (for $M=30$). This plot shows that the use of a smaller spatial window for block matching tends to allow more pixels to be processed at real-time throughput. Figure 4.4c shows the pixel processing throughput (with 25 fps rate) as a function of the slow-time block matching window size M that sets the number of frames to be included in each block matching operation (for 20×20 spatial windows). As expected, the number of pixels that can be accommodated with real-time throughput was decreased at larger slow-time block matching window sizes.

4.5.3 Block Matching is the Primary Computational Bottleneck.

As shown in Figure 4.5, the *BlockMatching* kernel was found to be the bottleneck in every tested configuration of our real-time ELS-VD framework. In particular, Figure 4.5a shows the percentage of processing time occupied by each GPU kernel as a function of slow-time block matching window size M for $L=1$ and $B=20$. As the slow-time block matching window size increased to 30, the *Blockmatching* kernel apparently dominated nearly 90% of the frame processing time. When a smaller spatial window dimension B was used for block matching, the *Blockmatching* kernel was found to occupy a shorter fraction of the frame processing time, as shown in Figure 4.5b for $L=1$ and $M=20$. The only instance where another kernel, *CalculateResidues*, was identified as a major computational bottleneck was when using a high aliasing order of $L=3$. As shown in Figure 4.5c (for $B=20$ and $M=20$), in the case of $L=3$, 98% of the frame processing time was attributed to two kernels: *CalculateResidues* (51%) and *BlockMatching* (47%).

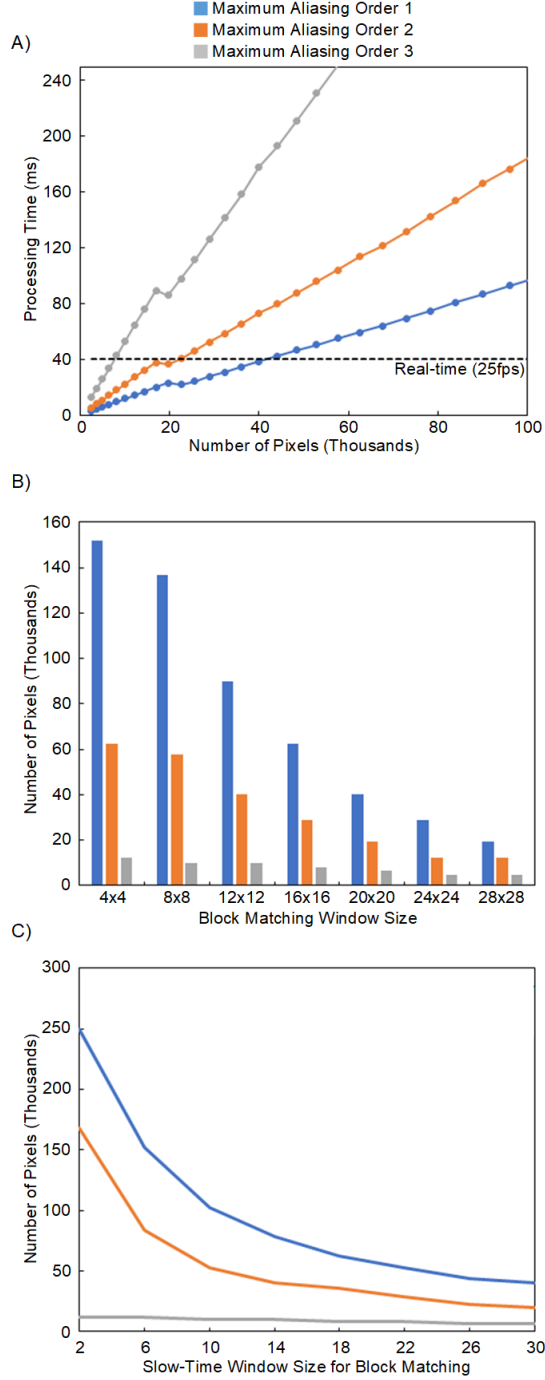


Figure 4.4. Computing performance of the real-time ELS-VD framework as a function of different parameters, including: numbers of flow pixels C , maximum aliasing order L , block matching window dimension B , and slow-time block matching window size M . (a) Frame processing time as a function of C for $L = 1, 2, 3$ (with $B = 20, M = 30$); (b) Number of flow pixels that can be processed in real-time (25 fps) for different $B \times B$ block matching windows ($L = 1, 2, 3; M = 30$). (c) Number of pixels that can be processed in real-time (25 fps) as a function of M ($L = 1, 2, 3; B = 20$).

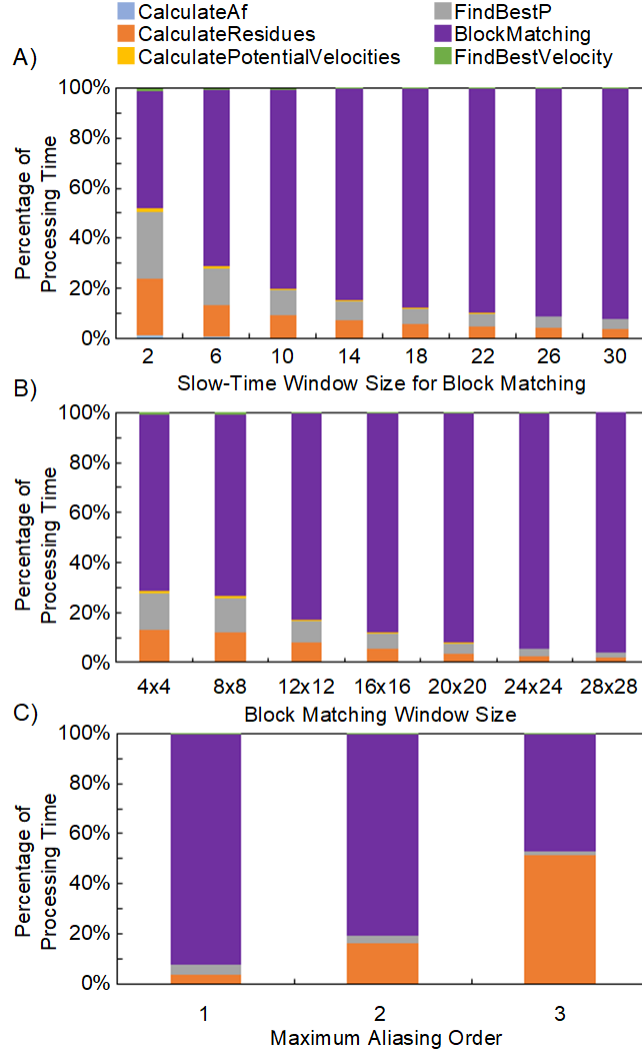


Figure 4.5. Processing time for each kernel in the real-time ELS-VD framework. Results are shown (in percentages) as a function of: (a) slow-time block matching window size M ($L = 1$; $B = 20$); (b) block matching spatial window size $B \times B$ ($L = 1$; $M = 20$); (c) Maximum aliasing order L ($B = 20$; $M = 20$). Number of flow pixels C was kept constant at 65,536 (256×256) in all cases.

4.6 Discussion

4.6.1 Summary of Contributions

There is significant interest in translating high frame rate vector flow imaging toward use in the clinic to facilitate the imaging of complex and transient phenomena (Godidi et al., 2017; Hansen et al., 2017; Hansen et al., 2019). Success in such a clinical translation drive can have ramifications on

cardiovascular health monitoring, especially in the diagnosis of diseased conditions such as atherosclerosis where complex hemodynamics are prone to emerge and are known to contribute to disease progression (Ku, Giddens, Phillips, & Strandness, 1985; Steinman & Taylor, 2005; Younis et al., 2004). For vector flow imaging to be favorably considered as a useful clinical diagnostic tool, it is important for this modality to be realized in real-time without aliasing artifacts. In response to such a technical need, we have sought to devise a GPU-based ELS-VD framework (Figure 4.1) with a set of SIMT computing kernels (Table 4.1) to derive aliasing-free velocity vector maps at real-time processing throughput. This framework has transcended beyond the original ELS-VD algorithm (Ekroll et al., 2016) by introducing parallel computing principles that favor real-time realization, and it has been implemented with the CUDA API and has been executed on commercial NVidia GPUs (RTX-2080).

Experimental results have shown that our GPU-based ELS-VD framework is effective in resolving aliasing artifacts in an anthropomorphic carotid bifurcation phantom with 50% stenosis (Figure 4.2) with short frame processing time that corresponds to real-time video display rates (Figure 4.3). Our framework has also succeeded in achieving real-time de-aliasing functionality for higher aliasing orders and for practical frame sizes with a reasonable number of flow pixels (Figure 4.4). Among all the computational operations of our ELS-VD framework, we found that block matching is generally the most time consuming (except in the case with an aliasing order of 3) and occupies the majority of processing time needed to derive each frame of dealiased vector flow map (Figure 4.5).

From an engineering standpoint, our GPU-based ELS-VD framework is of practical significance in that it represents the utility of GPU processing in achieving real-time aliasing-resistant vector Doppler estimation. Compared to the staggered PRF method (Madiena et al., 2018; Ricci et al., 2017), the proposed framework does not require additional transmissions at different PRFs which can strain data transfer bandwidth and limit the available ensemble length. This framework also does not require *a priori* knowledge of aliasing characteristics (i.e. no data training is needed), so it is fundamentally different from inference-based dealiasing solutions that rely on deep learning principles to correct for aliasing artifacts (Nahas et al., 2020). Note that the approach of accelerating various aspects of the ultrasound processing pipeline using GPU processing has been gaining momentum in the field (So, Chen, Yiu, & Yu, 2011). This processing strategy has been successfully applied to beamforming (Yiu et al., 2011; Yiu & Yu, 2015), Doppler processing (Chang et al., 2009; Chee et al., 2017) and strain imaging (Idzenga et al., 2014). It can be anticipated that our current work can directly contribute to

ongoing endeavors in realizing a full end-to-end GPU processing pipeline that can be implemented on the new generation of software-oriented research scanners (Yiu & Yu, 2016).

4.6.2 Insights on GPU Acceleration of ELS-VD

In this work, GPU acceleration of the ELS-VD framework was realized via SIMT parallelization with a three-level hierarchy of threads: frame level, pixel level, and operational level (Figure 4.1). This parallelization strategy naturally emerged from the data independencies of the ELS-VD framework, and it complemented the organization of threads used in GPU architecture. This thread organization, however, is dependent on specific ELS-VD computational steps being accelerated, and thus dedicated GPU kernels need to be devised for each part of the algorithm. In addition to thread organization, memory management is deemed to be another important consideration. Achieving real-time performance requires optimized use of the GPU’s on-chip memory, texture memory, and global memory when executing each kernel. While global memory is the largest in size, it is also the slowest to access. Variables that need to be accessed multiple times, such as the projection matrix $\mathbf{A}^P$ in Equations (4.4) and (4.5), should be transferred to the shared memory first before the matrix-vector operations are performed.

It is interesting to note that, although our ELS-VD framework has achieved real-time performance (Figure 4.3), we were not able to achieve full parallelization in some compute kernels because of practical constraints in the GPU’s hardware resources. Compromises need to be made during implementation (Secs. 4.3.3 to 4.3.5). Nonetheless, with the continuing development of more powerful GPUs, the impact of making these implementation compromises will likely be mitigated, and it is reasonable to expect that our framework can achieve an even higher processing throughput in the future.

As shown in our computing time analysis (Figure 4.5), the *BlockMatching* kernel was the primary computational bottleneck in our ELS-VD framework under almost all parameter combinations. This bottleneck stems from the repeated and random access of flow speckle image values from the GPU’s off-chip memory, and the number of these off-chip data accesses increased for larger block matching windows and slow-time window sizes. After all, a relatively large block matching window size (20×20) and a relatively large ensemble size (30 samples) were both needed to achieve robust dealiasing performance in the stenosed carotid phantom imaging scenario (or else significant spurious errors would arise, as we have discovered in our in-house preliminary tests). These findings are in accordance with the conclusions made in the original ELS-VD algorithm paper (Ekroll et al., 2016),

and they are the typical limitations of block matching in speckle tracking applications. Note that research has been done on the performance of speckle tracking in different settings and the impact of flow dynamics on its accuracy (Gao et al., 2015; Yeung et al., 1998). In the future, it would be worthwhile to incorporate these results to refine the performance of our real-time ELS-VD framework.

4.7 Conclusion

Aliasing is a known problem that affects the performance of vector Doppler estimation, and it in turn results in spurious rendering of velocity fields in vector flow imaging, especially when performed at high frame rates. To address such a problem, this paper has presented a real-time, GPU-based ELS-VD framework that can facilitate the derivation of aliasing-resistant flow vector maps. From a diagnostic efficacy standpoint, deriving aliasing-free vector flow images is critical to clinical cardiovascular diagnostics because complex flow patterns are typically marked by the existence of a wide range of flow speeds, for which slow speeds would emerge in recirculation zones while fast speeds would emerge during systole especially in a stenosed artery. Obtaining aliasing-free vector flow images in real-time, as demonstrated in this paper, can effectively facilitate on-site diagnostics at the bedside. In turn, the clinical value of vector flow imaging can be better substantiated.

Chapter 5

Fast Aliasing Resistant Vector Doppler Imaging Using Deep Learning and GPU Computing

5.1 Introduction

Visualization of blood flow dynamics *in vivo* is crucial for assessing cardiovascular health (Steinman & Taylor, 2005) and understanding the physiology of the circulatory system (Younis et al., 2004). For example, blood flow dynamics are significantly altered in conditions of vascular stenosis and flow speed is commonly used to grade the severity of the condition (Thiriet et al., 2015). A number of modalities have been used for blood flow imaging, including 4-D magnetic resonance imaging (Garcia, Barker, & Markl, 2019), optical tomography (Spaide, Fujimoto, Waheed, Sadda, & Staurenghi, 2018) and computed tomography with the aid of flow simulations (Lantz et al., 2018). More recently, multi-angle vector Doppler, an emerging ultrasound-based technique, has emerged as a powerful technique for the noninvasive visualization of complex flow dynamics *in vivo* at high frame rates (Du, Shen, Yiu, Yu, & Zhu, 2018; Goddi et al., 2017). Compared to the aforementioned modalities, the ultrasound-based vector Doppler is more accessible, has superior temporal resolution and uses fewer clinical resources with a smaller risk profile. Fundamentally, vector Doppler can be realized using a sequence of ultrasound plane wave transmissions with multiple steering angles (Jensen, J. A. et al., 2016b). For a given pixel in the imaging view, a flow vector is derived in a two-step process. First, pulsed Doppler estimation is applied on the data received from each steering angle. Second, the Doppler frequency estimates from each steering angle are fed to a least squares fitting operation to yield a flow vector (Yiu & Yu, 2016). With plane wave transmission, this process is applied on every flow pixel in a high frame rate vector flow cineloop (~3000 frame per second).

One pitfall in vector Doppler imaging is that vector estimation is jeopardized by Doppler aliasing (Pellett et al., 2005) in any of the slow-time signals from the different steering angles. Doppler aliasing occurs when the mean slow-time sampling rate, determined by the pulse repetition frequency (PRF) of the plane wave sequence, is lower than the maximum Doppler frequency induced by flow. This leads to erroneous measurements being used in the least square's estimator, and consequentially erroneous flow vectors in the flow profile. Doppler aliasing in vector Doppler jeopardizes the modality's ability to quantify and visualize flow *in vivo* (Yiu & Yu, 2016). Live vector Doppler applications at the bedside are particularly prone to aliasing because the available PRF

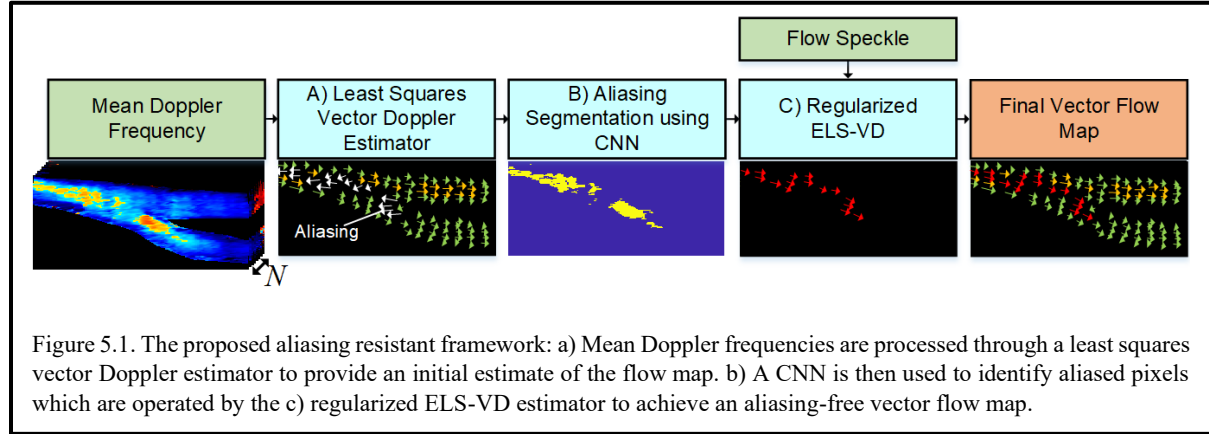
in bedside vector Doppler is limited by the number of transmitted steering angles needed to yield reliable results (Yiu & Yu, 2016) and the data transfer bandwidth available in the scanner (Boni et al., 2018; Yiu et al., 2019). Indeed, aliasing artifacts have been reported in Doppler based vector flow estimation in the carotid artery (Au et al., 2020; Goddi et al., 2018; Goddi et al., 2017), the femoral artery (Au et al., 2019) and in urinary flow phantom studies (Ishii et al., 2017). The situation is more severe in pathologies such as vascular stenosis where flow speeds can exceed 2 m/s (Thiriet et al., 2015), resulting in multi-cycle aliasing artifacts.

Correcting aliasing for vector Doppler is a challenging problem. Several solutions to Doppler aliasing exist in the literature, including postprocessing techniques such as image segmentation followed by adaptive phase unwrapping of the measured Doppler frequencies (Muth et al., 2011; Nahas et al., 2020). A staggered PRF has also been used to yield a six-fold increase in the maximum measurable Doppler frequency and used in real-time vector Doppler (Posada et al., 2016; Ricci et al., 2017). More recently, aliasing removal in vector Doppler has been achieved through a least squares estimator that incorporates all Doppler frequencies simultaneously (Flynn et al., 2011). This was then expanded to the extended least squares vector Doppler (ELS-VD) framework (Ekroll et al., 2016), where least squares estimation was developed further and speckle tracking (ST) (Jensen, J. A. et al., 2016b) was introduced to improve the generalizability of the framework. ELS-VD is successful in providing aliasing resistant vector flow estimation in the carotid artery, but excessive processing times prevent real-time bedside application. To address the processing throughput limitation, we previously presented a GPU accelerated pipeline for aliasing resistant velocity estimation using ELS-VD principles to achieve real-time performance (>25 fps) on a graphical processing unit (GPU) (see Chapter 4) but identified a bottleneck due to the number of flow pixels being processed.

In this study, we present an aliasing resistant estimator for vector Doppler imaging with high processing throughput and is suited for *in vivo* flow conditions. It achieves so by addressing a significant computational bottleneck in GPU accelerated ELS-VD (see Chapter 4), where the processing throughput is proportional to the number of pixels being processed. Here, we address this bottleneck through a high-speed inference neural network component that identifies in the imaging view the pixels affected by aliasing. We posit that a convolutional neural network (CNN) can be trained to identify aliased vectors in the flow map. The aliased vectors can then be removed rapidly by the accelerated ELS-VD framework applied exclusively on the affected pixels. This strategy minimizes the occurrence of false correction and accelerates the overall processing throughput of the

framework by reducing the number of pixels processing by the aliasing resistant estimator. The proposed framework builds on our previous work where we showed that CNNs can be trained to detect aliasing artifacts in color Doppler data and used to enhance the performance of aliasing removal techniques in that modality (Nahas et al., 2020). Processing throughput was validated using *in vivo* acquisitions from the femoral and carotid artery bifurcations.

5.2 Theoretical Principles



The objective of the framework is to achieve aliasing resistant vector Doppler estimation with high processing throughput. It achieves so by identifying aliasing-prone pixels in the flow map using a convolutional neural network, then applying the computationally demanding aliasing resistant velocity Doppler estimator on those aliased pixels only, thereby reducing the overall computational overhead per frame. To achieve this, the proposed framework consists of three main modules (see Figure 5.1):

1. A conventional least squares vector Doppler estimator (Yiu & Yu, 2016) that provides an initial estimate of the flow map with aliasing artifacts present (Figure 5.1a);
2. An *in vivo* trained CNN to segment aliased pixels in the vector Doppler map derived using module 1 (Figure 5.1b);
3. A regularized extended least squares vector Doppler estimator (Ekroll et al., 2016) (Chapter 4) that is exclusively applied on the aliased pixels (Figure 5.1c) as determined by module 1. The output of the final module is the aliasing free vector flow map.

The following sections define the imaging system and the aliasing problem in vector Doppler, then detail the principles of each module.

5.2.1 System and Problem Definition.

We define a high frame rate plane wave transmission and beamforming scheme consisting of the sequence of N transmit-receive steering angle pairs $[(\theta_1, \phi_1), (\theta_2, \phi_2), \dots, (\theta_N, \phi_N)]$ at a PRF per transmit angle of f_{PRF} and transmit frequency f_o . We further define the corresponding sequence of measured normalized mean Doppler frequency maps $\hat{u}_n(x, y, t)$ that can be obtained via beamforming and Doppler processing (Yiu et al., 2014), where x, y, t denote the lateral position, the axial position and the slow-time frame number respectively and $n=1 \dots N$ denotes the particular transmit-receive steering angle combination. This sequence of Doppler frequency maps is then used in deriving the vector flow map in vector Doppler theory (Yiu & Yu, 2016). The problem is that Doppler aliasing may occur in any $\hat{u}_n(x, y, t)$. This can be modeled as a discrete error:

$$\hat{u}_n(x, y, t) = u_n(x, y, t) + \epsilon_n(x, y, t) \quad (5.1)$$

where u_n is the true Doppler frequency and $\epsilon_n \in \mathbb{Z}$ is the discrete aliasing error and number of cycles of aliasing for the n^{th} transmit-receive steering angle. It is important to note that the PRF used in the transmission of the plane waves is reduced by a factor equal to the number of transmit angles. The more transmissions angles used, the greater the performance of the estimator (Yiu & Yu, 2016), but the reduced PRF results in increased likelihood of aliasing. The problem of aliasing removal in vector Doppler is then to determine ϵ_n for all n and remove it from the mean Doppler frequency measurements before use in vector Doppler estimation.

5.2.2 Module 1: Least Squares Vector Doppler Estimator

The objective of this module is to provide an initial estimate of the vector flow map using conventional least squares vector Doppler estimation (LS-VD) (Yiu & Yu, 2016) so that the subsequent CNN module can identify the aliased pixels. Doppler aliasing in any pixel in any $\hat{u}_n$ will manifest as disrupted flow regions in the output of the least squares estimator (Figure 5.1a). The velocity estimate at a given pixel (x, y, t) is the solution, $\mathbf{v}$, to the system of equations:

$$\mathbf{A}\mathbf{v} = \mu\hat{\mathbf{f}} \quad (5.2)$$

where:

$$\mathbf{A} = \begin{bmatrix} \cos(\theta_1) + \cos(\phi_1) & \sin(\theta_1) + \sin(\phi_1) \\ \vdots & \vdots \\ \cos(\theta_N) + \cos(\phi_N) & \sin(\theta_N) + \sin(\phi_N) \end{bmatrix} \quad (5.3)$$

, $\mathbf{f}=[\hat{u}_1(x,y,t),\dots,\hat{u}_N(x,y,t)]^T$ is vector of the measured normalized mean Doppler frequency and $\mu=c_0 f_{PRF}/f_0$ is the factor relating normalized Doppler frequency to flow speed. c_0 is the speed of sound in the system. A preliminary estimate at (x,y,t) can be obtained using the pseudo-inverse

$$\mathbf{A}^\dagger=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T:$$

$$\mathbf{v}_{\text{pre}}=\mu\mathbf{A}^\dagger\hat{\mathbf{f}} \quad (5.4)$$

Using these principles, a vector flow map for the entire imaging view can be constructed for each time point. There is no aliasing resistance at this stage so $\mathbf{v}_{\text{pre}}$ might be aliased if $\hat{\mathbf{f}}$ is corrupted by aliasing errors in any of its components.

5.2.3 Module 2: Aliasing Segmentation Using Convolutional Neural Networks

The purpose of this module is to identify regions in the vector flow map affected by aliasing artifacts so that they may be corrected in the final module. Initial vector maps produced by the first module will be fed to a CNN that is trained for aliasing segmentation. The output will be a probability map that denotes the likelihood that any given pixel is aliased.

1) Data Preprocessing

The vector flow maps obtained from the first module were processed to generate 2 channels images: flow speed $|\mathbf{v}|$ and flow angle $\angle\mathbf{v}$). This was done because high flow speeds and abrupt flow direction changes are strong indicators of aliasing. The channels were normalized to $[0,1]$ through linear mapping. Specifically:

$$|\mathbf{v}|_{\text{norm}}=\frac{|\mathbf{v}|}{\mu v_{\text{max}}}+0.5 \quad (5.5)$$

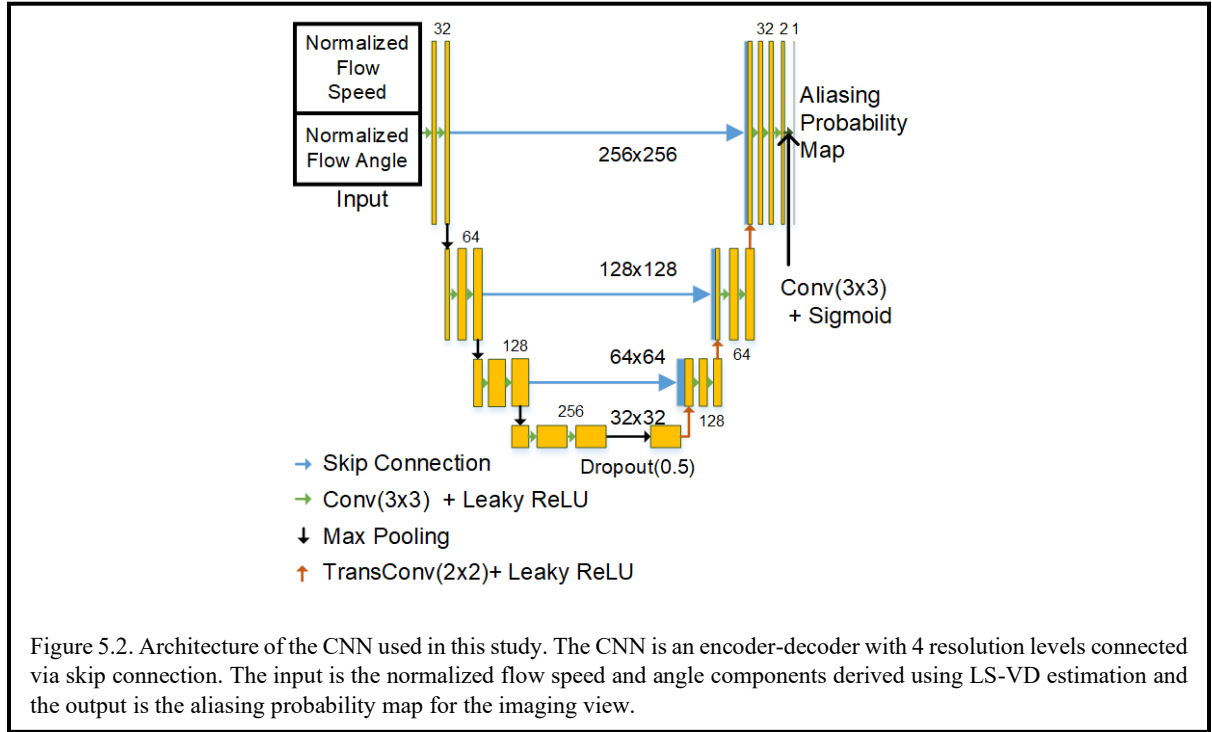
$$\angle\mathbf{v}_{\text{norm}}=\frac{\angle\mathbf{v}}{2\pi}+0.5 \quad (5.6)$$

Division by μ factors out the PRF related terms and keeps the trained network independent of that system parameter. v_{max} was turned so that $|\mathbf{v}|_{\text{norm}}$ was in $[0,1]$ in the training set.

2) Convolutional Neural Network Architecture

The neural network architecture (see Figure 5.2) used in this study was devised for the purpose of segmenting aliasing artifacts in the vector Doppler data. We devised the architecture using U-net features to benefit from its strengths in semantic segmentation (Ronneberger et al., 2015), but with simplifications to reduce inference time. Specifically, the network architecture is that of a fully

convolutional encoder-decoder, with skip connections connecting 4 resolution levels. Encoder-decoders are effective architectures for semantic segmentation (Badrinarayanan et al., 2017) and skip connections have been shown to improve segmentation resolutions (Drozdzal et al., 2016). Convolutions were done with a kernel of 3×3 and upsampling was performed using transpose convolutions with kernel of 2×2 . Activation in all but the last layer was done using a leaky rectified linear unit (RELU) with $\alpha = 0.3$. This CNN design choice was made to improve learning and to reduce the occurrence of “dying ReLU” cases (Lu, L. et al., 2020). The last layer was activated using a sigmoid function for segmentation. Figure 5.2 illustrates the architecture and exposes tensor sizes throughout the network. The use of these network architecture parameters was determined empirically based on prior trials, and the receptive field was deemed sufficient to discern aliased pixels. Note that the use of multiscale information is important in this case because aliasing can occur at varying regional sizes, from local to global discontinuities.



3) Reference Aliasing Segmentation Generation

The reference aliasing segmentation was automatically generated via the original ELS-VD framework tuned for high fidelity. This approach was utilized to label a large number of frames from many acquisitions to achieve a diverse training set. Specifically, the target aliasing-free vector flow map

was generated using the accelerated ELS-VD (Chapter 4) with a block matching size of 30×30 and an ensemble length of 30. We chose this strategy for two reasons: 1) This strategy allowed us to train on a large dataset of high frame rate acquisitions from multiple subjects as labeled data generation is automated. 2) There should be sufficient regularization in the neural network and its training process so that random dealiasing errors produced by this method are filtered out.

5.2.4 Module 3: Aliasing Resistant Vector Doppler

The purpose of this module (Figure 5.1c) is to derive a robust aliasing-free vector flow estimation in the affected pixels that are identified in the second module. This was done by applying a regularized version of the GPU accelerated aliasing resistant estimator previously reported (Chapter 4). In brief, this framework is based on the extended least squares vector doppler estimator (Ekroll et al., 2016) and its objective is to find the aliasing-free velocity estimate $\mathbf{v}_D$:

1) Discrete Squared Error Optimization of the Doppler System

A pen-optimal aliasing correction vector $\mathbf{p}$ is found by discrete optimization of equation:

$$\mathbf{p} = \underset{\mathbf{p}_k}{\operatorname{argmin}} |(\mathbf{A}\mathbf{A}^\dagger - \mathbf{I})(\hat{\mathbf{f}} + \mathbf{p}_k)|_2 \quad (5.7)$$

where $\mathbf{p}_k$ is an aliasing correction vector candidate in a discrete space of the aliasing vectors being considered. The space of aliasing correction vectors is bound by assuming that each component of $\mathbf{p}$ can take one value in the integer range of $[-L, L]$ where L is the maximum number of cycles of aliasing expected in the signal. Ideally, the addition of $\mathbf{p}$ to $\hat{\mathbf{f}}$ would negate all aliasing errors ϵ_n for all n . In practice, this optimization cannot differentiate between aliasing correction vectors $\mathbf{q}_i$ and $\mathbf{q}_j$ that are related by:

$$\mathbf{q}_i = \mathbf{q}_j + [l, l, \dots]^T \quad (5.8)$$

where $l \in \mathbb{N}$.

2) Speckle Tracking to Find the Aliasing Free Velocity Estimate

$\mathbf{p}$ is used to construct the set of candidates defined by Equation (5.8), which can be turned to velocity candidates using Equation (5.4) to form velocity candidates $\mathbf{v}_1, \dots, \mathbf{v}_{2L+1}$. The aliasing-free velocity vector $\mathbf{v}_D$ is then the member of this set that correlates with speckle motion at the pixel of interest. Speckle tracking is done in this study via ensemble block matching on a sequence of M consecutive flow speckle frames with a block matching window size of size $B \times B$ (Chapter 4). To reduce the

likelihood of spurious errors due to speckle decorrelation (Ekroll et al., 2016), we leverage the aliasing segmentation through a regularization mechanism that benefits of the flow continuity criteria. As flow is typically smooth and continuous in time and space within some finite spatiotemporal window, the correlation values for each velocity candidate are penalized by a penalization term:

$$\begin{aligned} \lambda M |\mathbf{v}_i(t) - \mathbf{v}_m(t-1)| \\ \text{for } i = 1 \dots (2L + 1) \end{aligned} \quad (5.9)$$

where λ is a forgetting factor that controls the level of regularization, $\mathbf{v}_i(t)$ is the i^{th} velocity candidate from the set being investigated at the current frame t and $\mathbf{v}_m(t-1)$ is the median velocity in a neighborhood of the location being investigated in the previous frame. The median filter kernel is chosen to be 3×3 . This technique biases towards non-aliased pixels as the regularized ELS-VD module is only applied on pixels that were flagged as aliased by the second module. Consequentially, spurious velocity candidates that are too spatiotemporally deviant from non-aliased pixels are more likely to be excluded.

5.3 Neural Network Development and Training

5.3.1 Dataset Collection

Data acquisition for the purpose of training, validation and evaluation was done using a research-purpose ultrasound scanner (US4US; Warsaw, Poland) that was equipped with a linear array transducer (SL1543; Esaote, Genoa, Italy). The scanner was configured for a sequence of two plane wave transmission angles (-10° , 10°). The scanner's live B-mode + color flow mode was used to guide the imaging of the arteries of interest. After an acquisition period of 2 seconds, the raw RF data is accessible and was copied to another system for offline processing. Details about the imaging platform and the scanning procedure are found in Table 5.1.

Table 5.1
Imaging Parameters Used in the Experiments.

Parameter	Value
Imaging Platform	
Number of Tx/Rx Channels	192
Array pitch	0.3048 mm
Pre-beamformed data sampling rate	25 MHz
Pre-beamformed data resolution	2 bytes per sample
Data acquisition	
Imaging frequency	5 MHz
Tx pulse duration (number of pulses)	3
Pulse repetition frequency	3000 Hz/angle
Data acquisition duration	2 s
Beamforming	
F-Number	2.0
Prefiltering	4-8 MHz bandpass, 64-tap finite impulse response.
Receive Apodization	hanning
Depth Range	0 – 4cm
Lateral Range	-1.9 – 1.9cm
Resolution	256 × 256
(Transmit, receive) steering angle combinations	(-10°, -10°), (-10°, -3°), (-10°, 6°), (-10°, 10°), (10°, -6°), (10°, 3°), (10°, 10°)

In vivo data sets were collected from long-axis acquisitions of the femoral bifurcation and carotid artery, which were chosen for their non-trivial geometry and triphasic flow characteristics with large variations in flow velocity over a cardiac cycle (Au et al., 2019; Au et al., 2020; Goddi et al., 2018). Flow in bifurcations is generally susceptible to aliasing in the deeper branches, whose sloped vessel geometry naturally gives rise to smaller beam-to-flow angles. The femoral bifurcation data were collected from 8 healthy adults, and the carotid artery data were collected from 11 older adults. 6 of the 11 older adults had clinically documented heart failure and plaques were seen in two carotid arteries from two different subjects, which were divided between the training and test sets. Another carotid dataset was obtained from an additional healthy volunteer where the probe was tilted and a lower PRF of 2000 Hz/angle was used to induce two-cycle aliasing artifacts. In total, the collected data was sought to train and evaluate the framework in healthy and pathological conditions to represent a range of physiological blood flow conditions. Each data acquisition trial was performed over an entire cardiac cycle, so it included systolic and diastolic flow as well as flow reversal and

recirculation. Acquisitions were collected as a part of an *in vivo* hemodynamics study protocol approved by the human research ethics committee at the University of Waterloo (ORE 22794).

5.3.2 Dataset Organization

To increase generalizability and prevalence of high order aliasing, the datasets were downsampled in PRF before use. Specifically, datasets at 3000 Hz/angle were downsampled to 1500 Hz/angle and 1000 Hz/angle and added back to the training set. Only datasets with aliasing in some of the frames were included. From the collected data, a training set of 15000 frames was assembled from 5 femoral and 7 carotid arteries whereas the validation set consisted of 3750 frames from 1 femoral and 2 carotid arteries. The test set of 700 frames was constructed from 2 femoral arteries and 3 carotid arteries, including a carotid with plaque and a tilted carotid from the healthy adult. The aliasing segmentation of the test set was obtained using the ELS-VD estimator followed by the manual inspection of each frame to correct any segmentation error. All sub-datasets included both one and two-cycle aliasing artifacts in the Doppler measurements from at least one of the transmit-receive steering angle combinations.

5.3.3 Training Hardware and Implementation

The network architecture was set up and trained in Python (version 3.6.7) using the Keras application programming interface (API) (version 2.1.6) with a TensorFlow-GPU (version 1.12) software back end on a computing platform that was equipped with a Xeon E5-2620 central processing unit (Intel, Santa Clara, CA, USA) and a GTX-1080 GPU (Nvidia). The neural network was trained by optimizing the binary cross-entropy loss between prediction and the reference segmentation realized using high fidelity ELS-VD (see Section 5.2.3.3). The optimization was done via the Adam optimizer (Kingma & Ba, 2014) with a learning rate starting at 10^{-4} and decreasing by a factor of 0.2 every time the validation loss function plateaued for 5 consecutive epochs until a minimum of 10^{-7} is reached. An early stopping mechanism was also included to terminate training after 15 consecutive epochs with no improvement in validation loss. Memory constraints in the GPU hardware (with 8 GB of random-access memory) necessitated a batch size of 1. During the training process, the sampled training frames were randomly flipped horizontally and/or rotated at a random angle $[0, 360^\circ]$, both with a probability of 50% per frame to improve training diversity.

5.4 Processing Pipeline for Aliasing Resistant Vector Doppler Imaging

5.4.1 Processing Hardware and Environment

To achieve high processing throughput, the framework was realized in a C++-CUDA environment on a computing platform that was equipped with a Core i7-6850K CPU @ 3.60 GHz central processing unit (Intel, Santa Clara, CA, USA) and an RTX-2080 graphical processing unit (GPU) (NVidia, Santa Clara). CUDA is an API for interfacing with Nvidia GeForce GPUs for general-purpose computing. The input of the framework, the mean flow maps, power and speckle patterns were transferred from the 64 GB RAM to the GPU global memory. The main thread in the framework then interfaced with dedicated GPU kernels and other CPU sub-routines to ultimately realize the theoretical framework proposed in Section 5.2.

5.4.2 Image Formation and Doppler Processing

The RF data was prefiltered to remove out-of-band noise, Hilbert transformed to form the analytic signal then beamformed using delay-and-sum principles to form seven analytic signals $w_n(x, y, t)$ with seven transmit-receive steering angle combinations. Details on the processing of the steering angle combinations are listed in Table 5.1. The 7 steering angle combinations were chosen for their favorable aliasing-resistance as recommended in the original ELS-VD contribution (Ekroll et al., 2016) while accounting for the transducer pitch. At each pixel in the imaging view per transmit-receive steering angle combination, the analytic signal was clutter filtered using a 169-tap FIR filter at 0.09 normalized frequency. The measured mean normalized Doppler frequency map, $\hat{u}_n(x, y, t)$ was estimated from the clutter filtered and beamformed analytic signal $w_n(x, y, t)$ (Yiu & Yu, 2016) using lag one autocorrelation with ensemble length $G = M = 30$. For pixel location (x, y) :

$$\hat{u}_n(t) = \frac{1}{2\pi} \arg \left\{ \sum_{i=t-G}^{t-1} w_n^*[i] w_n[i+1] \right\} \quad (5.10)$$

For flow detection and speckle images, the analytic signals w_n from all transmit-receive steering angle combinations were coherently compounded to form w_{cc} . Flow detection was derived using Doppler variance, $\hat{\sigma}(x, y, t)$, computed using the lag-one auto correlation, followed by log compression. For a pixel location (x, y) :

$$\hat{\sigma}(t) = 20 \log_{10} \left| \frac{1}{G-1} \sum_{i=t-G}^{t-1} w_{cc}^*(i) w_{cc}(i+1) \right|_2 \quad (5.11)$$

Doppler variance, as opposed to Doppler power, was used for better flow detection (Tremblay-Darveau et al., 2017). Any spurious detection artifacts, such as due to reverberation, were manually removed. The flow speckle image $\hat{s}(x, y, t)$ was formed from the power of the last frame in the Doppler ensemble. At pixel location (x, y) :

$$\hat{s}(t) = 20 \log_{10} |w_{cc}(t)|_2 \quad (5.12)$$

5.4.3 Aliasing Resistant Vector Flow Estimation

The processing pipeline used in each module of the aliasing resistant estimator (Section 5.2) are described in the following subsections:

1) Least Squares Vector Doppler Estimation

The aliasing resistant vector flow imaging framework was realized using a dedicated GPU kernel with frame-independent variables computed on the CPU during initialization. Specifically, the $\mathbf{A}^\dagger$ is computed for Equation (5.4) in a CPU subroutine built using the Armadillo library (Sanderson & Curtin, 2016; Sanderson & Curtin, 2018), then copied to global GPU memory. In the GPU kernel, one thread is launched per pixel in the image. Using these threads, $\mathbf{A}^\dagger$ is copied from global memory to shared memory. Each thread then forms the frequency vector $\hat{\mathbf{f}}$ pertaining to the pixel being processed, then computes the matrix-vector multiplication of $\mathbf{A}^\dagger \hat{\mathbf{f}}$ using iteration. The resulting $\mathbf{v}_{pre}$ is also converted to the two channels (normalized flow speed + angle, Figure 5.2) for CNN inference then stored in global memory for subsequent processes in the framework.

2) Aliasing Segmentation using CNN

Aliasing segmentation is realized by inference of the pretrained CNN described in Section 5.2.3.2 using the TensorRT library (version 10.1; Nvidia Corporation, Santa Clara, CA, USA). TensorRT is a high-performance neural network inference optimizer and runtime engine using GPU processing that is suited for the deployment of deep learning solutions in a C++ application. In the initialization stage, TensorRT codec was used to configure the neural network model trained in Tensorflow to a back-end inference engine that can be used using high-level functions. The resulting aliasing segmentation map was combined with the flow detection map using a CPU subroutine to form a table of the pixel

coordinates that are aliased. This table is transferred to GPU memory for subsequent access by module 3.

3) Robust Extended Least Squares Vector Doppler

The implementation of this module utilizes the previously reported GPU accelerated pipeline for aliasing resistant velocity estimation using the CUDA API (Chapter 4). In the initialization stage, the matrices, $\mathbf{A}$ and $\mathbf{A}\mathbf{A}^\dagger - \mathbf{I}$ as well as the vectors $(\mathbf{A}\mathbf{A}^\dagger - \mathbf{I})\mathbf{p}_k$ for every k in the aliasing space being considered are computed and stored in a table in GPU memory. The computations for aliasing resistant vector doppler estimation are then realized using 6 dedicated GPU kernels exclusively applied on aliased pixels identified in aliased pixels' coordinates table. The kernels are optimized to perform the necessary matrix-vector operations and speckle tracking. A detailed description of the implementation of these GPU kernels can be found in the previous study (Chapter 4) and is omitted here for brevity. The only modification made in this study was the inclusion of the regularization factor described in Equation (5.9).

5.5 Experimental Validation

5.5.1 Evaluation of Aliasing Segmentation by the CNN

We evaluated the aliasing segmentation performance of the CNN model on the unseen dataset consisting of *in vivo* femoral and carotid acquisitions. The prediction results were compared against references derived from high fidelity ELS-VD with manual correction to remove spurious errors (Section 5.2.3.3). The efficacy of a CNN in identifying aliased pixel positions in the flow region was evaluated using average-precision. It is important that the segmentation does not miss out on aliased vectors, while extra segmentation may be tolerated by the aliasing resistant estimator. To this end, the segmentation threshold was tuned to maintain a high recall (>95%).

5.5.2 Aliasing Resistance and Processing Throughput Evaluation

The framework was evaluated for processing throughput and aliasing resistance, the ability to detect aliasing as well as choose the correct velocity estimate. This was compared to the performance of the accelerated ELS-VD framework applied on all flow pixels, with the same block matching window size and correlation ensemble length but without the regularization described in Section 5.4.3.3. The processing time per frame of the proposed framework and the accelerated ELS-VD framework was measured for all datasets described in this section.

Quantitative aliasing resistance evaluation was done on an anthropomorphic carotid bifurcation phantom with 50% stenosis (Chee, Ho, Yiu, & Yu, 2016) with a pulsatile flow of blood-mimicking fluid achieved using a programmable pump. The phantom includes a model of the common carotid artery as it splits into the superficial and deep arteries. The vessels were designed to generally have a diameter of 6 mm, apart from a lesion in the superficial artery where 50% stenosis is induced. This mimics the pathological condition of atherosclerosis in the carotid and creates a clinically relevant testing scenario. The phantom was imaged using our scanner at a PRF of 3000 Hz/angle. To create a test set with ground truth, the pump was configured to produce a carotid flow profile with a period of 3 seconds and a peak flow rate of 2.8 ml/s. At this flow rate and PRF, the peak flow speed is around 30 cm/s and no aliasing is found in the vector flow map derived using the conventional LS-VD estimator and thus vector can serve as a ground truth.

To generate the test set, we downsample this data to a PRF of 750 Hz/angle, creating two-cycle aliasing artifacts in the stenosed region. At this downsampled PRF, the period of the flow profile is equivalent to the number of frames that a flow profile with a 0.75 second period would occupy at a PRF of 3000 Hz/angle, leading to a reasonable heartrate of 80 bps. This allowed us to preserve the speckle correlation properties at the lower PRF to resemble those that would be seen at the higher more realistic PRF. Moreover, the two-cycle aliasing seen at the low PRF would translate to a flow speed of $30 \times 4 = 120 \text{ cm/s}$ at the higher PRF. Such flow speeds are indeed encountered in stenosed conditions (Thiriet et al., 2015). Aliasing resistance of the proposed framework will be quantified as the root mean squared error of the predicted flow map using the proposed algorithm at low PRF relative to the flow profile obtained LS-VD at high PRF. This will be performed on flow pixels only and on 1000 frames centered around the systolic phase.

The framework was also applied on three *in vivo* datasets for processing throughput analysis and performance demonstration. The datasets were selected from the test set and included: a healthy femoral bifurcation carotid artery at 3000 Hz/angle with 1 cycle aliasing only; a carotid artery with stenosis leading to jet formation that manifested as aliasing artifacts at PRF = 3000 Hz/angle; and the tilted carotid artery sample at PRF = 2000 Hz/angle with two-cycle aliasing. This provided a variety of flow conditions and PRFs for evaluation.

The parameters of the regularized ELS-VD module were tuned for lower speckle quality in the *in vivo* case and to accommodate the sharp flow gradients in the carotid phantom case. For the *in vivo* case, a block matching kernel size of $B=30$ and regularization factor $\lambda=0.5$ was used. For the carotid

bifurcation phantom dataset, block matching kernel size of $B=15$ and regularization factor of $\lambda=0.1$ were used. It is important to note that in scenarios with excessive aliasing and at low PRFs, some pixels may be corrupted by bandwidth aliasing which leads to erroneous mean Doppler frequency measurements as the signal is effectively ‘whitened’. These corrupted pixels cannot be processed by LS-VD estimation and were removed from the flow map in this study in vector flow visualization and quantitative analysis.

5.6 Results

5.6.1 Evaluation of Aliasing Segmentation by the CNN

The CNN for aliasing segmentation converged in training and demonstrated successful aliasing segmentation on all test scenarios. Specifically, segmentation scored an average-precision of 90.8% on the test set. Since recall was a higher priority as mentioned previously, a threshold of 0.1 was used for the rest of the paper. At that threshold, we report the recall, precision, specificity, and F-1 score in Table 5.2.

Table 5.2
Segmentation Performance with Threshold = 0.1.

Score	%
Recall	99.1
Precision	52.0
Specificity	87.4
F-1 score	68.2

Aliasing segmentation was achieved on vector Doppler output with 1 and 2 cycle aliasing. Figure 5.3 shows the corresponding segmentation maps on the different scenarios in the test set with 1-cycle and 2-cycle aliasing artifacts. The first column is a visualization of the vector flow maps computed using the conventional LS-VD, with the aliasing artifacts highlighted. The second column shows the aliasing segmentation performance on the different scenarios with the errors relative to manual segmentation color-coded by classification. The column on the right visualizes the mean Doppler frequency maps from one of the transmit-receive steering angle combinations using the color scheme typically used in color flow Imaging. This last column was included to emphasize aliasing artifact/cycle and contours more clearly, with the specific transmit-receive steering angle combination is denoted on each Doppler image. For 1 cycle aliasing, systolic frames from the femoral bifurcation

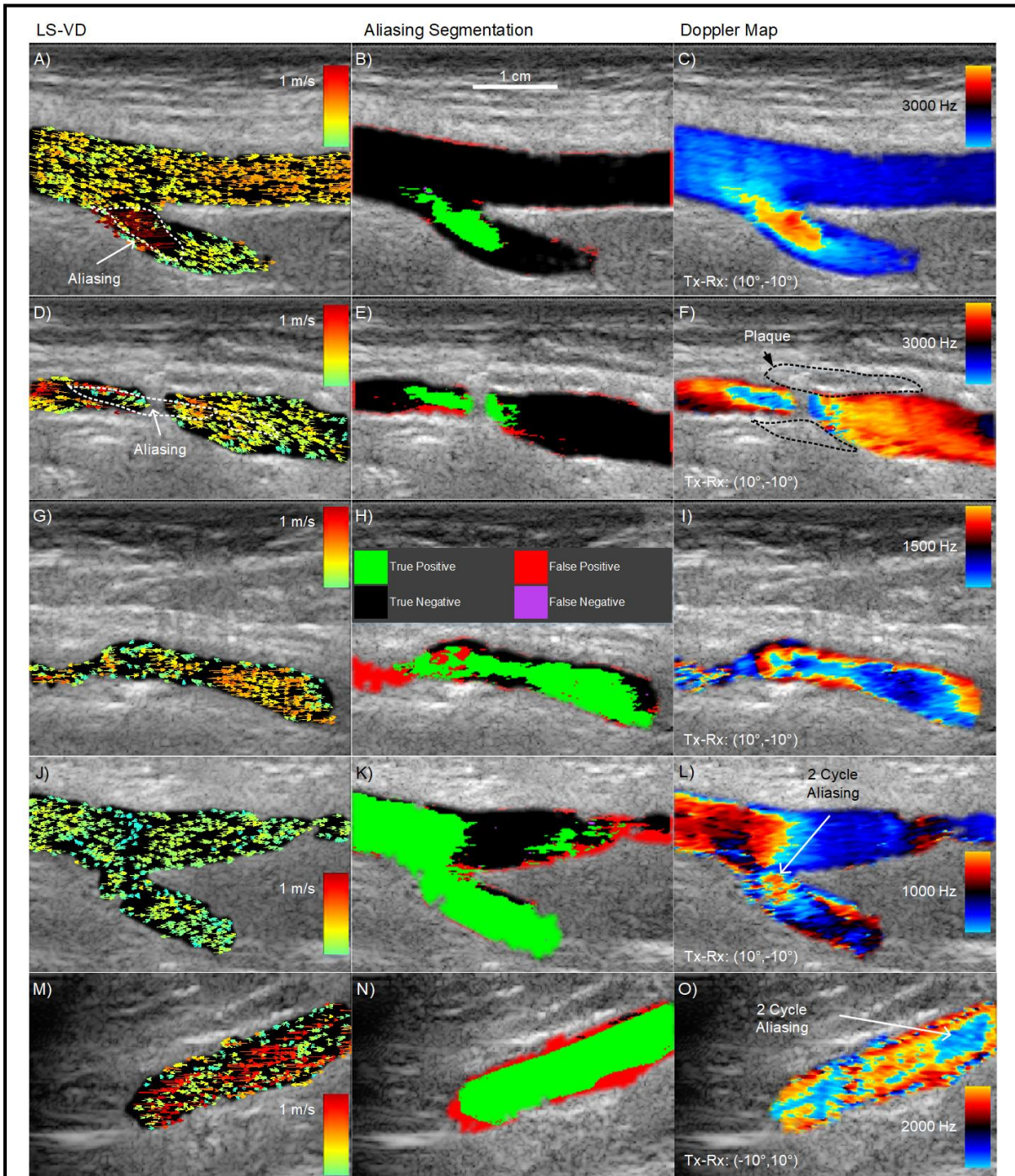


Figure 5.3. Aliasing segmentation performance on the test set: (a-c): Femoral bifurcation with 1 cycle aliasing; (d-f): Carotid artery with 1 cycle aliasing due to plaque; (g-i): Carotid artery with 1 cycle aliasing due to low PRF; (j-l): Femoral bifurcation with 2 cycle aliasing due to low PRF. (m-o): Carotid artery with two cycle aliasing due to low PRF and Probe tilt. The first column is a visualization of the vector flow map calculated using conventional LS-VD estimator. The second column showcases segmentation performance relative to the reference with errors color-coded. The right column is a visualization of the mean Doppler frequency maps from the one the transmit-receive steering angle combinations to emphasize aliasing cycle order and contours.

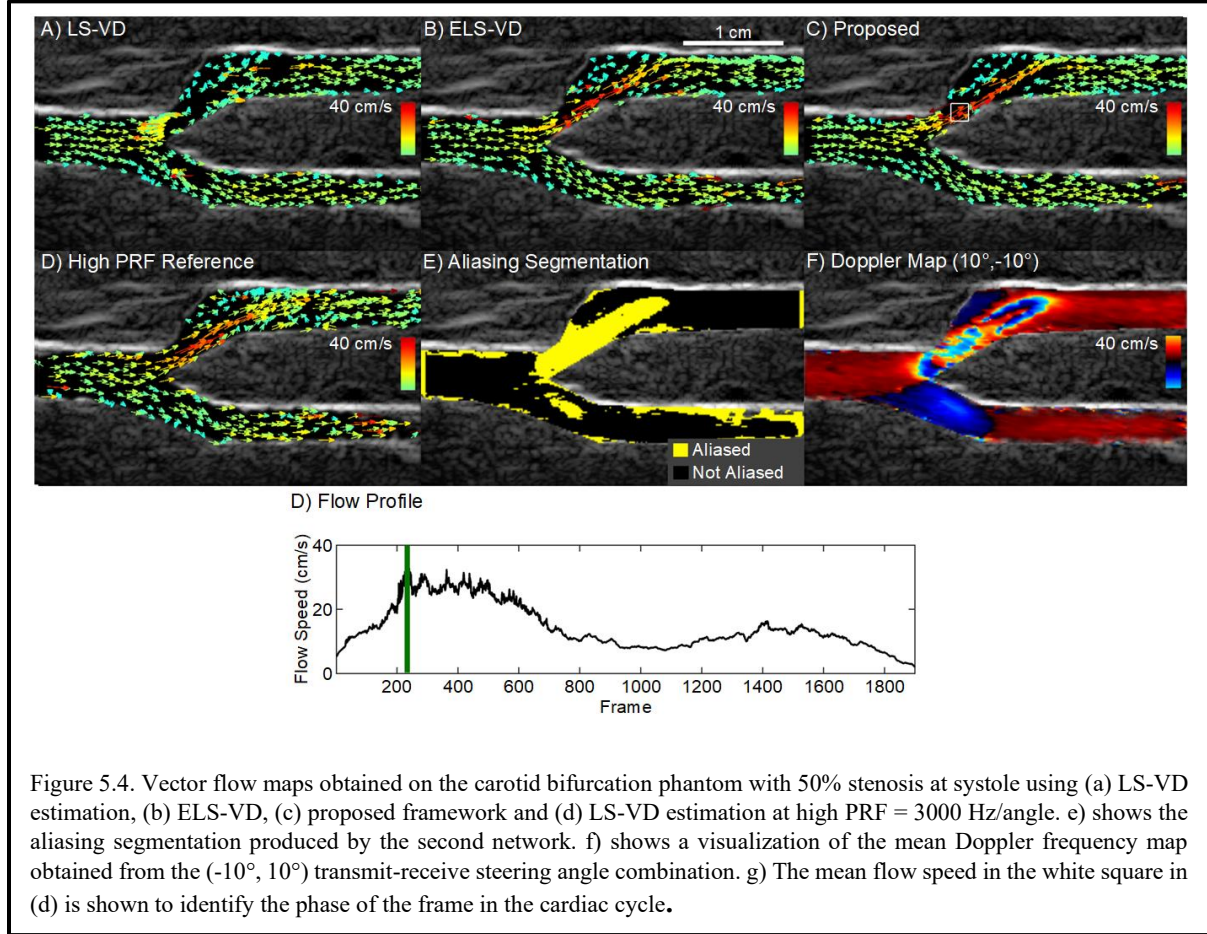
are shown in the top row. The second row showcases a pathological carotid artery with plaque build-up leading to the formation of a jet. This jet is captured as aliased pixels in the flow map that are successfully segmented by the CNN. The third row in Figure 5.3 showcases performance on a plaque-free carotid at a lower PRF of 1500 Hz/angle. For two-cycle aliasing, an acquisition from a femoral bifurcation at low PRFs of 1000 Hz/angle and a tilted Carotid artery at 2000 Hz/angle are shown in the last two rows. The maps rendered in Figure 5.3 are all overlaid over the corresponding B-mode image of that scenario using the flow detection map to convey the vasculature geometry to the reader. The key observation here is that aliasing segmentation was successful in a variety of vessel geometry and even when the entire lumen was aliased (Figure 5.3m-o). False-positive errors were encountered near the flow-tissue boundary, the edges of the frame or in pixels with low signal quality (Figure 5.3i).

5.6.2 Aliasing-Free Vector Flow Maps Realized Using the Proposed Framework.

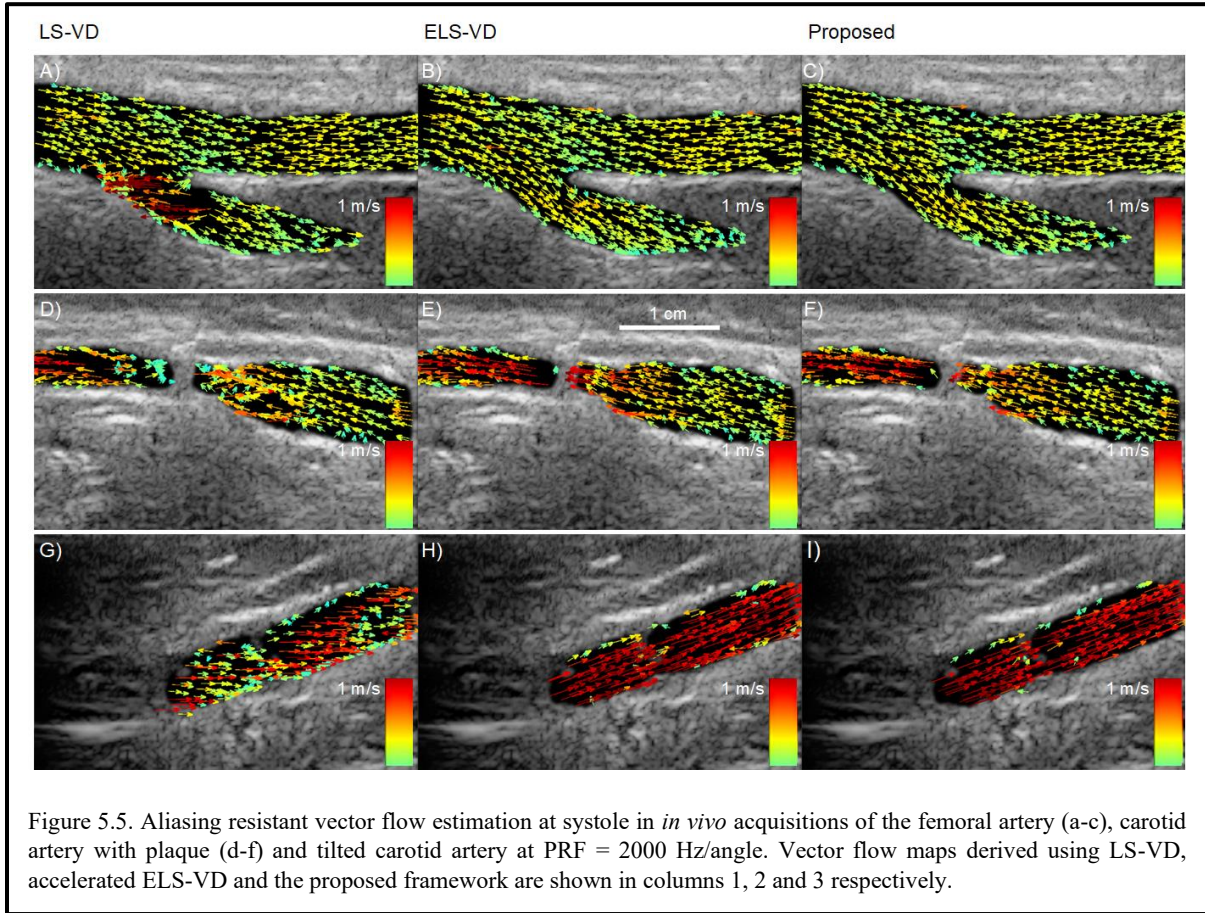
The overall framework succeeded in removing aliasing artifacts in the femoral and the carotid arteries containing 1 and 2-cycle aliasing. We showcase aliasing removal in Movies 5.1-4 for the carotid bifurcation phantom, femoral artery (Figure 5.3a-c), carotid artery with plaque (Figure 5.3d-f) and tilted carotid artery at PRF=2000 Hz/angle (Figure 5.3m-o) respectively. In the movies, we render the vector flow cineloop derived using conventional LS-VD, ELS-VD and the proposed framework. Vector flow visualization is done using vector projectile imaging principles (Yiu et al., 2014). In Movie 5.1, we also show the vector flow cineloop for the carotid phantom derived using LS-VD at high PRF= 3000 Hz/angle. The movies also include a cineloop of the aliasing segmentation by the CNN and a cineloop of the mean Doppler frequency map from one of the transmit-receive steering angle combinations to emphasize aliasing contours and order. The phase of the cardiac cycle can be tracked using a plot of the flow speed in the white square included with the movies. Snapshots at systole of the movies are shown in Figure 5.4 for the carotid phantom and in Figure 5.5 for the *in vivo* datasets.

On the carotid phantom, the proposed framework demonstrated successful aliasing-resistant vector flow estimation in the presence of 2 cycle aliasing in the jet (Figure 5.4f). Whereas the LS-VD estimator (Figure 5.4a) produced a highly corrupted flow profile, the output of the proposed framework matched the reference frame obtained at high PRF (3000 Hz/angle). Compared to the high PRF reference at systole, the output of the proposed framework achieved an RMSE of 5.07 cm/s, whereas the accelerated ELS-VD framework achieved an RMSE of 5.23 cm/s. For comparison, the

output of the LS-VD estimator at low PRF achieved an RMSE of 14.68 cm/s. In the body of the jet of peak systole (Movie 5.1), we did observe some signal dropout due to extreme spectral bandwidth aliasing at the low PRF.



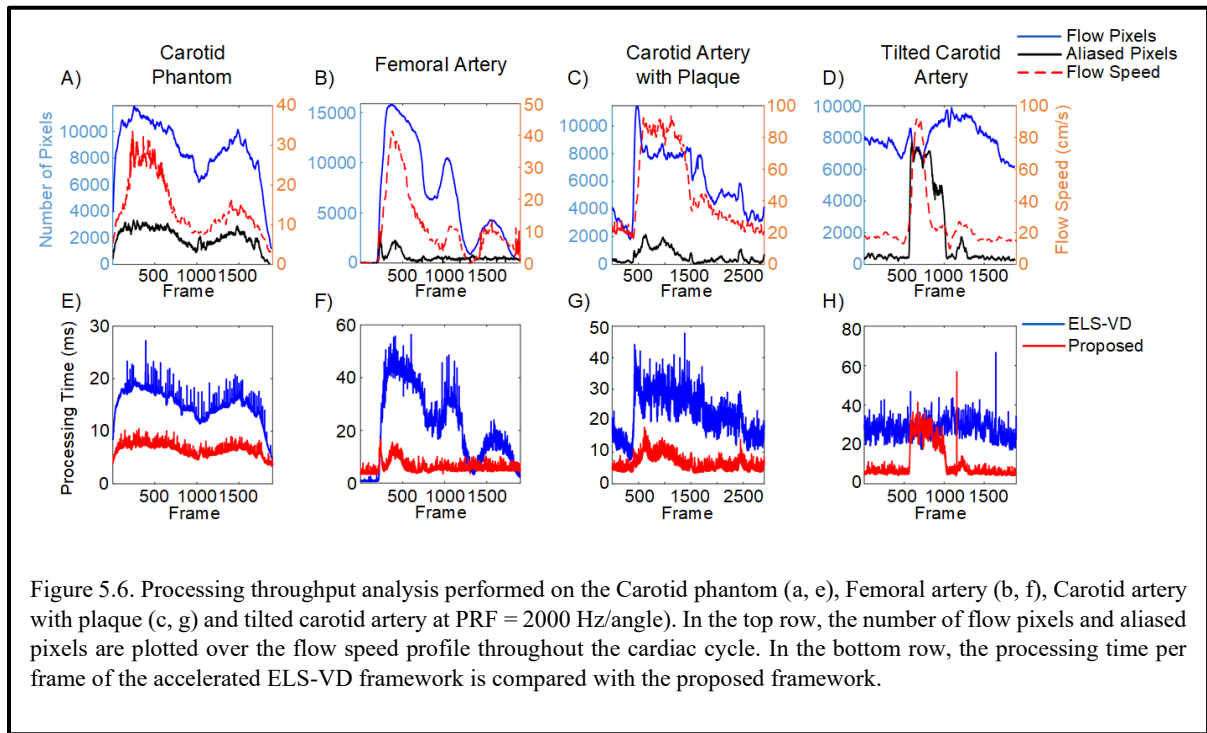
The key observation is that the output of the proposed framework matched the output of the accelerated ELS-VD framework applied on the entire flow region in the *in vivo* data sets from the femoral and carotid arteries (Figure 5.5, Movies 5.2-4). Aliasing resistant vector flow estimation was observed in the presence of 1 (Figure 5.5a-f, Movie 5.2, 3) and 2 cycle aliasing artifacts (Figure 5.5g-i, Movie 5.4). Successful aliasing removal was also seen in the pathological case with plaque (Figure 5.5d-f, Movie 5.2). Without aliasing resistance, the vector flow maps derived using LS-VD (Figure 5.5a, d and g, Movie 5.2-4) showcase significant aliasing, especially in the tilted carotid example at PRF = 2000 Hz/angle where the whole flow profile is corrupted (Figure 5.5g-i, Movie 5.4).



5.6.3 Processing Throughput Increased in the Proposed Framework

The proposed framework achieved higher processing throughput when compared to the GPU accelerated ELS-VD. In Figure 5.6, we plot the processing throughput analysis results for the carotid bifurcation phantom (Figure 5.6a, e), femoral artery (Figure 5.6b, f) from Figure 5.3a-c, carotid artery with plaque (Figure 5.6c, g) from Figure 5.3d-f and the carotid artery at 2000 Hz/angle (Figure 5.6d, h) from Figure 5.3m-o. The top row in Figure 5.6 shows the number of flow pixels detected in the image and the number of aliased pixels detected by the CNN per frame. These plots are overlaid on a plot of the flow speed from vector flow profiles derived using the proposed estimator to track the cardiac cycle. In the bottom row, we plot the processing time per frame for the GPU-accelerated ELS-VD and the proposed framework.

The major observation in this analysis is that the overall processing throughput of the proposed framework is significantly less than that of the accelerated ELS-VD GPU framework. While the accelerated ELS-VD had a processing time of around 20–50 ms per frame, the proposed framework had processing times under 20 ms for most cases. Time gains were clearest in the scenarios with lower aliasing pixel count relative to the number of flow pixels. (Figure 5.6a, e). In one instance, the tilted carotid artery at PRF= 2000 Hz/angle, almost all flow pixels were aliased and thus the proposed framework processing time approached that of ELS-VD at systole (Figure 5.6d, h). On the entire cardiac cycle, however, the processing time however was considerably lower in the proposed framework. With the proposed framework, the third module was typically the bottleneck in conditions of aliasing. While the processing throughput of the third module varied with the number of aliased pixels detected in the scene, the processing times for the first and second modules remained constant throughout the cardiac cycle at 0.1 ms and 2.7 ms per frame respectively.



5.7 Discussion

5.7.1 Summary of Contributions

Realizing vector Doppler imaging at the bedside reliably is important for the clinical adoption of the modality. One of the fundamental challenges preventing clinical uptake for vector Doppler imaging is Doppler aliasing which leads to erroneous estimation in applications with limited PRF or high velocities. By leveraging deep learning and GPU processing principles (Figure 5.1), we present an aliasing resistant vector flow estimator with high processing throughput (25-100 fps). This is achieved by addressing the computational bottleneck due to the number of flow pixels being processed by aliasing resistant estimation (Chapter 4). Specifically, we devised a two-step strategy where a CNN (Figure 5.2) segments aliased vectors in vector flow maps derived using conventional LS-VD estimation (Figure 5.1), so that the computationally demanding aliasing resistant estimator is only applied where necessary. The entire processing pipeline is implemented using a GPU codec with dedicated GPU kernels.

Experimental results have shown reliable aliasing segmentation (Figure 5.3) and aliasing resistant vector flow estimation in a stenosis phantom (Figure 5.4) and *in vivo* acquisitions from the carotid and femoral arteries (Figure 5.5). Successful aliasing resistance was reported in healthy (Figure 5.5a-c, g-i) and pathological conditions (Figure 5.5d-f) in the presence of 1 and 2 cycle Doppler aliasing (Figure 5.5g-i) and throughout the entire cardiac cycle (Movies 5.1-4). This was achieved with a low compute time that was under 20 ms in almost all cases (Figure 5.6).

Our work showcases the first application of deep learning to aliasing in vector Doppler imaging. Deep learning is an emerging strategy for signal processing in medical ultrasound, with applications reported in beamforming (Luchies & Byram, 2018) and flow estimation (Evain et al., 2020). Compared to previous work done in flow estimation, the presented framework employs deep learning in an integrated framework that includes other modules such as LS-VD estimation and speckle tracking with the ultimate purpose of improving processing throughput. This strategy allows us to simplify the deep learning problem and thereby reduce network and training complexity, while simultaneously reaping the benefits of machine learning for complex problems. Moreover, the chosen class of networks, the CNNs, is widely supported with high-level APIs that enable rapid on-site inference using GPU processing.

5.7.2 A Framework Suited for Clinical Translation

The proposed framework is suited to highly dynamic and fast flow that may be found in arteries. The proposed strategy leverages the fact that high velocities are typically bound to the systolic phase to a specific spatiotemporal window, by specifically directing the application of aliasing resistant vector flow estimation to those pixels and using the faster and adequate LS-VD estimation elsewhere, thereby reducing the overall processing time per frame (Figure 5.6). Moreover, the network used for aliasing segmentation is trained on real acquisitions from superficial arteries with varying vasculature and should therefore be suited to a wide range of pulsatile arterial flow.

For bedside application, vector flow estimation is one among several processing steps between raw RF data and on-screen vector flow visualization, which include beamforming, clutter filtering and vector rendering (Du et al., 2018). This means that the aliasing-resistant vector flow estimation process itself must be realized with high processing throughput using limited computing resources to maintain scanner accessibility. With the low processing time achieved per frame using a single commercial GPU, the proposed framework should readily fit in a GPU-powered and software programmable ultrasound scanner (Yiu et al., 2019). Compared to existing real-time aliasing-resistant vector Doppler that make use of staggered transmission (Ricci et al., 2017), the proposed framework uses fewer transmissions per frame, minimizing the necessary data transfer bandwidth, and has demonstrated velocity estimation exceeding 3 times the Nyquist limit.

The network in this work was trained and evaluated on *in vivo* acquisitions from the femoral and carotid arteries from a range of apparently healthy adults and older adults with document cardiovascular disease, with a focus on stenosis as a target application. In practice, there are other scenarios where aliasing in vector Doppler is of practical concern, such as in cardiac (Hansen et al., 2013) and deep vasculature imaging (Ishii et al., 2017). This generalizability limitation is typical of deep learning applications in ultrasound and requires expanded training and evaluation datasets.

5.8 Conclusion

Doppler aliasing jeopardizes vector Doppler estimation and is an obstacle to clinical adoption of the modality. Using deep-learning and GPU processing, an aliasing resistant vector flow estimator is realized for *in vivo* arterial flow with high processing throughput. By removing aliasing, the proposed framework significantly improves vector flow visualization in healthy and pathological conditions.

With its processing throughput and *in vivo* focus, the framework is suitable for integration with a programmable ultrasound scanner for bedside imaging.

Chapter 6

Aliasing Resistant Vector Doppler at the Bedside Using GPU Processing and Deep Learning

6.1 Introduction

Imaging blood flow in the human vasculature is a common procedure in the clinic for cardiovascular health monitoring in conditions such as vascular stenosis and stroke (Steinman & Taylor, 2005). However, these conditions are typically characterized by disturbed and complex blood flow dynamics which can be difficult to capture using existing flow imaging modalities in the clinic. High frame rate vector flow imaging (HiVFI) is a next-generation ultrasound-based modality that enables the visualization and quantification of complex and transient flow dynamics *in vivo* (Jensen, J. A. et al., 2016a; Jensen, J. A. et al., 2016b). A number of implementations of HiVFI exist in the literature, often through a mixture of unfocused transmissions and vector flow estimation techniques such as transverse oscillations (Jensen, J. et al., 2017; Lenge, Ramalli, Tortoli, Cachard, & Liebgott, 2015), speckle tracking (Nyrnes, Fadnes, Wigen, Mertens, & Lovstakken, 2020) and vector Doppler (Ricci et al., 2014; Yiu et al., 2014). High frame rate vector Doppler can be realized using plane wave transmissions with the velocity vector at each pixel in the imaging view then derived using a two-step process. First, pulsed Doppler estimation is performed on the slow-time signal at the pixel of interest for each steered plane wave. Second, the Doppler frequencies from each steering angle are used in least squares fitting based on the Doppler equation to estimate the velocity vector. This strategy has been employed in one commercial scanner with a reported framerate on the order of 100 fps (Du et al., 2018) but there is an overall limited adoption in the clinic.

One cause for the slow clinical adoption of high frame rate vector Doppler is the prevalence of Doppler aliasing-related flow estimation errors that render the modality unreliable for general use and practically ineffective in diseased conditions (Pellett et al., 2005). Vector Doppler estimation is corrupted with aliasing artifacts when the frequency content of any of the slow-time signals from any of the steering angles exceeds half the pulse repetition frequency (PRF) used for transmission (Flynn et al., 2011). This occurs in fast-flow conditions and leads to erroneous Doppler frequencies being measured (Goddi et al., 2017). These incorrect measurements then corrupt the least squares estimator in vector Doppler, producing invalid flow vectors. In on-site applications, PRF is limited by the available data transfer bandwidth (Boni et al., 2018; Yiu et al., 2019) and the number of steering

angles needed to yield reliable results (Yiu & Yu, 2016). The available maximum velocity under these PRF conditions is typically lower than flow speeds encountered in both healthy and diseased arteries (Goddi et al., 2018; Thiriet et al., 2015). Indeed, aliasing artifacts are routinely reported Doppler-based HiVFI from bifurcations (Au et al., 2019; Goddi et al., 2018) and in the presence of stenosis (Goddi et al., 2017; Hansen et al., 2019).

Several innovations have been made in addressing aliasing artifacts in Doppler-based HiVFI. Packet transmissions and time-domain correlation techniques have been used with TO based HiVFI to increase the PRF of the system (Jensen, J. A., 2019a; Jensen, J. A., 2019b) and staggered transmission has seen success in vector Doppler (Madiena et al., 2018; Ricci et al., 2017). Aliasing removal in vector Doppler has also been achieved through extending the conventional least squares error estimator to consider aliasing in the Doppler measurements (Flynn et al., 2011). This approach was later expanded in the extended least squares vector Doppler estimator (ELS-VD) (Ekroll et al., 2016), which improved the least squares estimator further and used speckle tracking to resolve ambiguities. While ELS-VD was successful against two cycles of aliasing, excessive processing times prevented real-time application. We previously presented a GPU-accelerated pipeline for aliasing resistant velocity estimation that utilizes convolutional neural network segmentation to reduce the number of pixels processed by computationally demanding aliasing resistant estimation (Chapter 5).

In this paper, we present a high frame rate vector Doppler imaging system for bedside use, with real-time aliasing resistant vector flow playback on demand. This is realized using the US4US platform, a GPU-equipped open platform that exposes RF data for a custom ultrasound processing pipeline (Yiu et al., 2019). We posit that raw RF data can be processed to aliasing resistant vector flow maps + B-mode in a frame rate over 20 fps for on-demand playback at the bedside. This is achieved in a vector Doppler imaging system that showcases an end-to-end GPU pipeline for beamforming (Yiu et al., 2011), filtering, Doppler estimation and visualization and makes use of an on-site deep learning module to achieve high processing throughput. The framework is demonstrated in pilot trials to image superficial arteries in the body of a human subject.

6.2 Framework for Real-time Aliasing Resistant HiVFI

The high frame rate vector Doppler imaging system is realized as a software program on a programmable ultrasound scanner configured for plane wave transmission (Figure 6.1). The program configures the scanner for multi-angle plane wave transmission, receives the raw pre-beamformed RF

data in real-time and processes it to a B-mode + flow speckle + vector flow map frame that is rendered in real-time. Two modes of operation are available:

- 1) Live mode: raw RF data is processed in packets through conventional least squares vector Doppler estimation principles to realize live vector flow imaging.
- 2) Replay mode: the raw RF data is continuously processed using aliasing resistant vector flow maps estimation, with the output played back in slow-motion.

The need for aliasing resistance emerges from the limited PRF available. The PRF is limited by the available data transfer bandwidth of the scanner and the need for multiple steering angles in plane wave transmission for good quality B-mode generation and robust velocity estimation (Yiu & Yu, 2016).

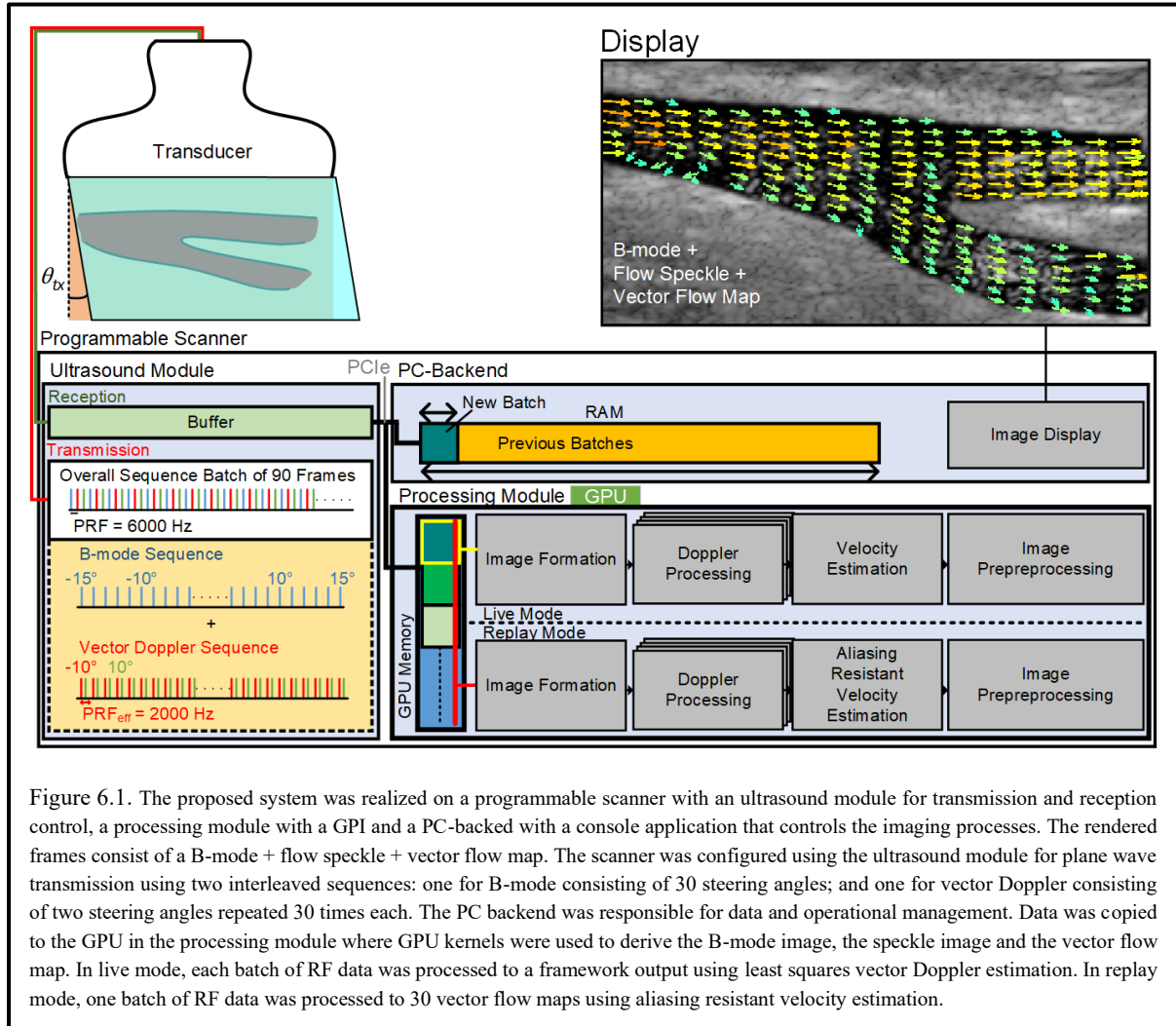


Figure 6.1. The proposed system was realized on a programmable scanner with an ultrasound module for transmission and reception control, a processing module with a GPU and a PC-backed with a console application that controls the imaging processes. The rendered frames consist of a B-mode + flow speckle + vector flow map. The scanner was configured using the ultrasound module for plane wave transmission using two interleaved sequences: one for B-mode consisting of 30 steering angles; and one for vector Doppler consisting of two steering angles repeated 30 times each. The PC backend was responsible for data and operational management. Data was copied to the GPU in the processing module where GPU kernels were used to derive the B-mode image, the speckle image and the vector flow map. In live mode, each batch of RF data was processed to a framework output using least squares vector Doppler estimation. In replay mode, one batch of RF data was processed to 30 vector flow maps using aliasing resistant velocity estimation.

6.2.1 Hardware

The framework was developed on a software-oriented open platform (USPlatform; us4us Ltd., Warsaw, Poland). A detailed description of the hardware in the scanner can be found in a previous publication (Yiu et al., 2019). The key features (Figure 6.1) include a 192-channel programmable ultrasound module, a GPU processor (GTX 1080Ti; NVidia Corporation, Santa Clara, CA, USA), and a personal computer back end that are all connected with a Peripheral Component Interconnect Express (PCIe) backplane (BPG8032; Trenton Systems). The scanner was equipped with a linear array (SL1543; Esaote, Genoa, Italy) and can be configured for custom plane wave imaging sequences. The raw RF data is transferred using the PCIe with a maximum bandwidth of 4.8 GB/s to the PC's 24 GB Random-Access Memory (RAM) in batches of frames. Data on the RAM is then processed using the C++ programmed software that interfaces with the GPU using the compute unified device architecture (CUDA) API (version 10.1; Nvidia Corporation, Santa Clara, CA, USA). Details on the imaging system are shown in Table 6.1.

Table 6.1
Array and Transmission Parameters.

Parameter	Value
Number of Tx/Rx channels	192
Number of samples/channel	1536
Pre-beamformed data sampling rate	25 MHz
Pre-beamformed data resolution	2 bytes per data sample
Transmit frequency	5 MHz
Tx pulse duration (number of pulses)	3
Pulse repetition frequency	6000 Hz

6.2.2 Transmission Protocol

The transmission protocol of the ultrasound module in the programmable scanner was configured to a repeating sequence of $P = 90$ steered planewaves (Figure 6.1) consisting of two interleaved sub-sequences: 1) A B-mode sub-sequence of 30 transmission events; and 2) A vector Doppler sub-sequence of 60 transmissions events. The B-mode sequence consisted of planewaves steered at $\theta_{tx} \in [-15^\circ, 15^\circ] \setminus \{0^\circ\}$ with a spacing of 1° for forming high-resolution B-mode images using coherent compounding. The vector Doppler sub-sequence consists of two alternating steering angles $\theta_{tx} \in \{-10^\circ, 10^\circ\}$ that were used for vector flow estimation. In the overall sequence (see Figure 6.1),

one B-mode related transmission event was performed, followed by two vector Doppler events that were steered at the two aforementioned angles. The sequence was transmitted using the full available aperture (192 channels) at the maximum available PRF= 6000 Hz, leading to an effective PRF of 2000 Hz per vector Doppler angle. Transmission configuration on the programmable scanner is done using a JavaScript Object Notation (JSON) file that specifies multiple additional parameters including transmission frequency, pulse length, transmission power and more. Table 6.1 shows the transmission parameters used in this study. During live-imaging mode, the received RF data was transferred to the PC backend in batches of $P = 90$ frames.

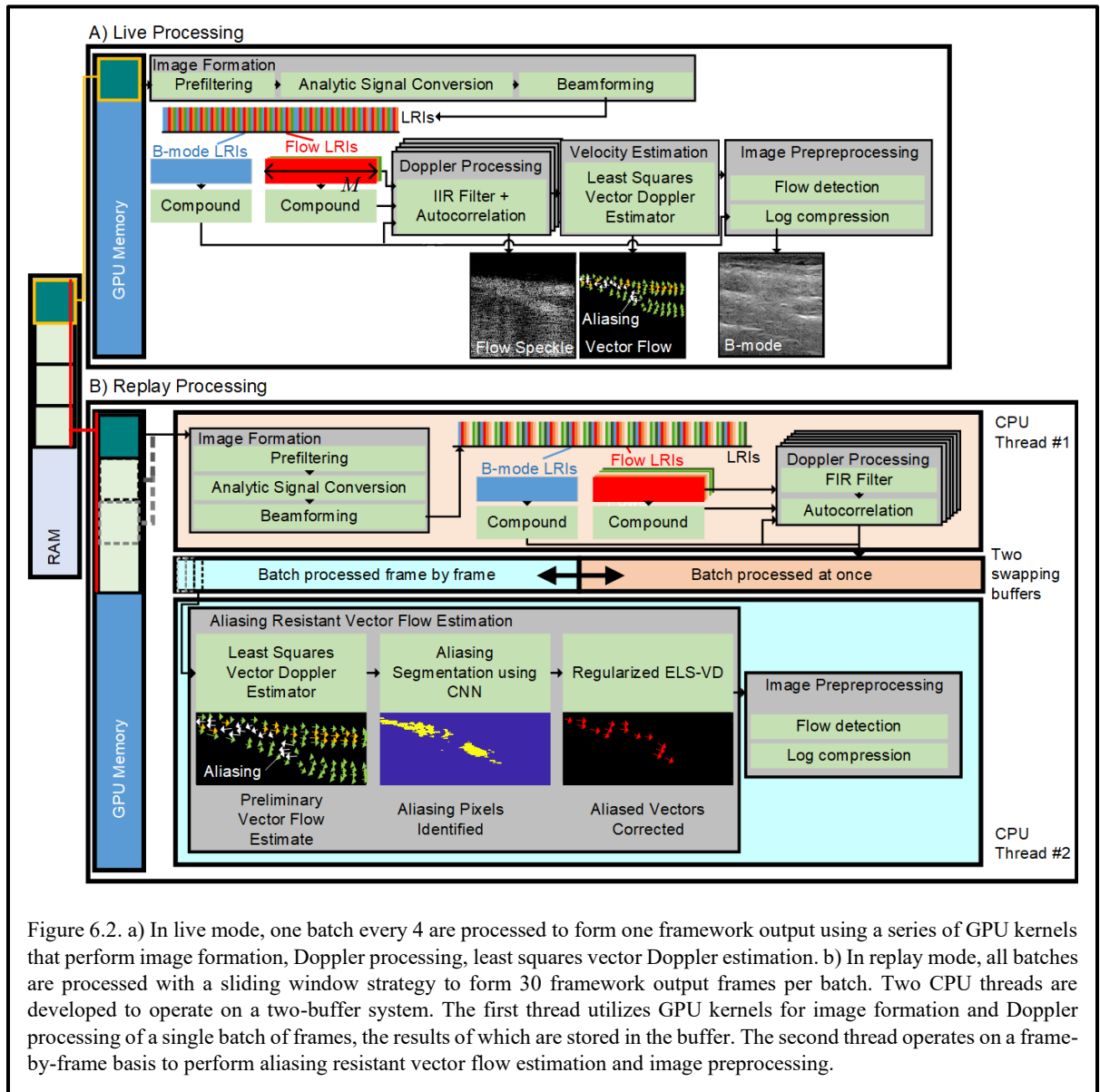


Figure 6.2. a) In live mode, one batch every 4 are processed to form one framework output using a series of GPU kernels that perform image formation, Doppler processing, least squares vector Doppler estimation. b) In replay mode, all batches are processed with a sliding window strategy to form 30 framework output frames per batch. Two CPU threads are developed to operate on a two-buffer system. The first thread utilizes GPU kernels for image formation and Doppler processing of a single batch of frames, the results of which are stored in the buffer. The second thread operates on a frame-by-frame basis to perform aliasing resistant vector flow estimation and image preprocessing.

6.2.3 Software Implementation Overview

The vector flow imaging framework was implemented in C++ with the bulk of signal processing performed on the GPU using CUDA kernels that were launched by the main program. All batches of data were copied to the RAM as they were received, with a maximum of 150 batches being stored. At the PRF of transmission, this translates to 2.25 seconds of acquisition time. An output frame was derived from GPU processing of the raw RF data using a series of operations: image formation, Doppler processing, velocity estimation, image preprocessing. The frame was then displayed using C++ OpenGL API (version 4.6; Khronos Group, Beaverton, OR, USA). The processing pipelines for both operational modes are detailed in the following two subsections:

6.2.4 Live Mode Implementation

In live mode, only one out of the $Q=4$ batches (Figure 6.2a) were processed to a vector Doppler + B-mode frame to match the frame rate of the imaging framework and the refresh rate of the screen which was naturally lower than the PRF. At the PRF of transmission, this translated to a live frame rate of 16.7 fps. To use GPU processing on a batch of frames, data from that batch was first copied from the RAM to the GPU's global memory using a callback function that is triggered after a batch of acquisitions. The batch was then processed synchronously after its acquisition using the following stages to yield an output frame.

1) Image Formation

The raw RF data were first filtered to remove out-of-band noise then Hilbert transformed to form the analytic signal using two dedicated GPU kernels (Figure 6.2a). Prefiltering was done using a 64-tap finite impulse response (FIR) (4-8 MHz) bandpass filter applied on each channel. The analytic signal was performed using another 51-tap FIR filter whose output was the imaginary part of the analytic signal. Each sequence of 90 analytic frames was delay-and-sum (DAS) beamformed to low-resolution images (LRI) in GPU global memory with different receive steering for subsequent B-mode and vector Doppler estimation. The beamforming grid was set to a resolution of 256×256 , an imaging depth of $[0, 4]$ cm, and a lateral axis of $[-1.9, 1.9]$ cm. The B-mode subsequence was all beamformed with 0° receive steering to form 30 LRI frames for B-mode computation. The vector Doppler sequence was beamformed to 90 LRI frames consisting of an ensemble of 30 frames for three transmit-receive steering angles denoted in Table 6.2a. Receive steering was done using a hanning apodization and a constant F-number = 1.0 and 2.0 for B-mode and vector Doppler frames

respectively. After beamforming, the B-mode related LRI frames were extracted from the buffer and coherently compounded to form the high-resolution image (HRI) for the B-mode image, which was then stored in the GPU global memory for subsequent visualization.

Table 6.2
Transmit-Receive Steering Angles Used.

A) Live Mode

Tx-Rx angle	1	2	3
Transmit Angle (°)	-10	-10	10
Receive Angle (°)	-7	8	8

B) Replay Mode

Tx-Rx angle	1	2	3	4	5	6	7
Transmit Angle (°)	-10	-10	-10	-10	10	10	10
Receive Angle (°)	-10	-3	6	10	-6	3	10

2) Doppler Processing

The LRI analytic frames pertaining to the vector Doppler sequence were divided into ensembles of 30 frames pertaining to each transmit-receive steering angle, then processed to find the mean Doppler frequency, variance and speckle pattern per transmit-receive steering for the whole image. These processes (Table 6.2a) were done using a single GPU kernel that was iteratively called three times, one for each transmit-receive steering angle to ultimately produce the mean Doppler frequency maps necessary for the vector flow estimation of one frame from one batch. A 10th order projection-initialized infinite impulse response filter was for clutter removal at u_c normalized frequency that can be controlled by the user. Lag-one autocorrelation was then performed to derive the mean Doppler frequency, variance and instantaneous speckle image for each transmit-receive steering angle. The programming of all kernels described in this section is reported in our previous work (Yiu et al., 2019; Yiu et al., 2011).

For the n^{th} transmit-receive steering angle, the measured mean normalized Doppler frequency map, $\hat{u}_n(x, y, t)$ with x, y, t being the lateral position, the axial position and the slow-time frame number respectively, was estimated from the clutter filtered and beamformed analytic signal $w_n(x, y, t)$ (Yiu & Yu, 2016) using lag one autocorrelation with ensemble length $G = M = 30$. For pixel location (x, y) :

$$\hat{u}_n(t) = \frac{1}{2\pi} \arg \left\{ \sum_{i=t-G}^{t-1} w_n^*[i] w_n[i+1] \right\} \quad (6.1)$$

For flow detection and speckle images, the analytic signals w_n from all transmit-receive steering angle combinations were coherently compounded to form w_{cc} . Flow detection was derived using Doppler variance, $\hat{\sigma}(x, y, t)$, computed using the lag-one autocorrelation, followed by log compression. For a pixel location (x, y) :

$$\hat{\sigma}(t) = 20 \log_{10} \left| \frac{1}{G-1} \sum_{i=t-G}^{t-1} w_{cc}^*(i) w_{cc}(i+1) \right|_2 \quad (6.2)$$

Doppler variance, as opposed to Doppler power, was used for better flow detection (Tremblay-Darveau et al., 2017).

The flow speckle image $\hat{s}(x, y, t)$ was formed from the power of the last frame in the Doppler ensemble. At pixel location (x, y) :

$$\hat{s}(t) = 20 \log_{10} |w_{cc}(t)|_2 \quad (6.3)$$

3) Vector Flow Estimation

Vector flow estimation in live mode was achieved using the conventional least squares vector Doppler estimator (Yiu & Yu, 2016) on the mean Doppler frequency maps derived from each batch. The velocity estimate at a given pixel (x, y, t) is the solution, $\mathbf{v}$, to the system of equations:

$$\mathbf{A}\mathbf{v} = \mu \hat{\mathbf{f}} \quad (6.4)$$

where:

$$\mathbf{A} = \begin{bmatrix} \cos(\theta_1) + \cos(\phi_1) & \sin(\theta_1) + \sin(\phi_1) \\ \vdots & \vdots \\ \cos(\theta_N) + \cos(\phi_N) & \sin(\theta_N) + \sin(\phi_N) \end{bmatrix} \quad (6.5)$$

θ_i and ϕ_j are the i^{th} transmit and receive steering angles respectively, $\hat{\mathbf{f}} = [\hat{u}_1(x, y, t), \dots, \hat{u}_N(x, y, t)]^T$ is a vector of mean Doppler frequencies, $\mu = c_0 f_{PRF} / f_o$, c_0 is the speed of sound and f_o is the transmit frequency. An estimate at (x, y, t) can be obtained using the pseudo-inverse $\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$:

$$\hat{\mathbf{v}} = \mu \mathbf{A}^\dagger \hat{\mathbf{f}} \quad (6.6)$$

Using these principles, a dedicated GPU kernel was devised to estimate the vector flow map for the entire imaging view for each time point. In the GPU kernel, one thread is launched per pixel in the

image, utilizing the maximum threads per block in CUDA (1024). Using these threads, $\mathbf{A}^\dagger$ is copied from global memory to shared memory. Each thread then forms the frequency vector $\hat{\mathbf{f}}$ pertaining to the pixel being processed, then computes the matrix-vector multiplication of $\mathbf{A}^\dagger \hat{\mathbf{f}}$ using iteration. The result is a vector flow map for the imaging view that can then be visualized.

4) Image Preprocessing

The final frame to be rendered is composed of three layers: the B-Mode image, the flow speckle map, and the vector flow map. In the preprocessing stage, GPU kernels were used to log compress the coherently compounded B-mode HRI and threshold the Doppler frequency variance map to form the flow detection map. Persistence, median and gauss filters were applied on the flow detection map to remove spurious flow detection and feather its edges.

5) Image Display and Vector Flow Visualization

OpenGL was used to composite the B-mode and speckle images on the screen with dynamic range management into a window of a resolution of 800×760 . Vector flow visualization was developed using C++ and OpenGL where each vector on the frame was rendered using two OpenGL polygons that were stacked in the shape of an arrow. In live imaging, a static grid of vectors is employed where the color and length of the vector are computed from the flow speed and color bar. The static grid is chosen in live mode over a projectile-based visualization due to the speed of the dynamics in real-time which would make it challenging to interpret the fast-moving. A screen update is performed once a new output frame has been produced. Since every fourth batch is processed, the frame update in live imaging is 16.7 fps given the transmission scheme described in Section 6.2.2.

6.2.5 Replay Mode Implementation

In replay mode (Figure 6.2b), all 150 batches stored in the RAM were continuously processed to vector flow + B-mode frames and played back in real-time, with a yield of 30 frames per batch. In contrast to the live mode, an asynchronous approach was used in the replay mode with two CPU threads running in parallel on a two-buffer system. One thread operated on a batch of RF data at a time for image formation and Doppler processing after copying it to GPU memory from the RAM, with the results stored in one buffer. In parallel, a secondary thread performed velocity estimation and image preprocessing on the stored results from the other buffer on a frame-by-frame basis. Once a complete buffer of frames has been processed and its contents displayed, the buffers swap to allow for

subsequent batches to be processed. The frame rate of this second thread can be controlled by the user to adjust the speed of the cineloop. In this paper, the frame rate of this thread was set to an average of 60 fps. Frame skipping and delays may be used by this thread to preserve a constant time evolution regardless of the processing time required to derive each aliasing-free vector map. The velocity estimation and preprocessing thread was given priority to the GPU resources so that the image formation and Doppler processing are carried out concurrently with minimal disruption to the velocity estimation and display.

1) Image Formation

Beamforming in replay mode (Figure 6.2b) differed to live mode in that the vector Doppler sequence was beamformed to 210 LRI frames consisting of an ensemble of 30 frames for 7 transmit-receive steering angles denoted in Table 6.2b. The 7 steering angle combination was chosen to reduce vector Doppler variance (Yiu & Yu, 2016) and because of their inherent aliasing resistance (Ekroll et al., 2016).

2) Doppler Processing

Replay mode Doppler (Figure 6.2b) processing was done using sliding window strategy where the new 30 frames were appended to older frames before processing. This strategy yields 30 sets of mean Doppler frequency maps to ultimately produce 30 vector flow frames from each batch of RF data. A GPU kernel was designed to extract the 30 LRI frames corresponding to a specific transmit-receive steering angle. These were then clutter filtered by a 169-tap FIR filter to remove clutter at u_c normalized frequency in a dedicated GPU kernel. A subsequent kernel performed lag-one autocorrelation to derive the mean and variance of the Doppler frequency for each transmit-receive steering angle as in live mode. This kernel was called iteratively for each of the 7 transmit-receive steering angles. A final call of the kernel was performed on the coherently compounded ensemble formed from the summation of the 7 transmit-receive steering angles to form a higher resolution speckle pattern and Doppler frequency variance map for flow detection.

3) Vector Flow Estimation

In replay mode, aliasing resistant vector flow estimation was realized using a previously reported framework (Chapter 5). In brief, the framework (Figure 6.2b) consists of three modules:

- 1) Least squares vector Doppler estimation to provide an initial estimate of the vector flow map with possible aliasing,
- 2) Aliasing segmentation of the artifacts in the initial vector flow map estimate by a convolutional neural network (CNN),
- 3) Aliasing correction using a regularized extended least squares vector Doppler estimator applied exclusively on the aliased pixels.

This strategy ensures that the computationally demanding aliasing resistant estimator is only applied where needed, reducing overall computational overhead. In this section, we summarize the key aspects of this framework and the GPU kernels used to realize it.

A) Module 1: Least Squares Vector Doppler

In the first module, the mean Doppler frequencies from all transmit-receive steering angles are used to derive an initial estimate of the flow vector using the same least squares vector Doppler estimator used in the live mode. There is no aliasing resistance at this stage so some estimates might be aliased. Processing here is identical to that reported in Section 6.2.4.3.

B) Module 2: Aliasing Segmentation

In the second module, the vector flow map from the first module is passed through a pretrained CNN to segment aliased pixels. The CNN is an encoder-decoder that was previously trained offline on *in vivo* vector flow maps from femoral and carotid arteries with varying extents of single and double-cycle aliasing artifacts (Chapter 5). Rapid inference was realized using the TensorRT library, a high-performance inference optimizer and runtime engine that is suitable for deploying pretrained deep learning models in GPU systems. Details about the CNN architecture, training and evaluation are reported in the previous study and omitted here for brevity (Chapter 5). In this work, we complement the segmentation with the morphological operation of binary closing is applied using two GPU kernels on the segmentation results (dilation followed by erosion with a 9x9 binary disk stencil) to prevent the occurrence of gaps.

C) Module 3: Aliasing Removal

After aliasing segmentation, an aliasing resistant vector flow estimator is applied on the affected pixels. This estimator is based on ELS-VD (Ekroll et al., 2016) and achieves real-time performance through GPU processing. The aliasing-free velocity estimate $\mathbf{v}_D$ is derived using a two-step process:

1. Discrete squared error optimization of the Doppler system to find a set of velocity candidates, one of which is $\mathbf{v}_D$.
2. Speckle tracking to find the aliasing free velocity estimate in the set of velocity estimates produced by the first step.

Speckle tracking was done in the form of block matching using the sum of absolute differences (SAD) on the instantaneous flow speckle images, with a window size $B \times B$ on an ensemble of M consecutive frames. This operation was computed in a dedicated GPU kernel. We eliminate spurious errors to speckle decorrelation in the second step by penalizing velocities $\mathbf{v}_i$ that disrupt the continuity of flow.

$$\lambda M |\mathbf{v}_i(t) - \mathbf{v}_m(t-1)| \text{ for } i = 1 \dots (2L+1) \quad (6.7)$$

where λ is a forgetting factor that controls the level of regularization, $\mathbf{v}_i(t)$ is the i^{th} velocity candidate being investigated at the current frame t and $\mathbf{v}_m(t-1)$ is the median velocity in a neighborhood of the location being investigated in the previous frame.

This regularized aliasing resistant velocity estimator is realized using 6 dedicated GPU kernels exclusively applied on aliased pixels identified in aliased pixels' coordinates table. The kernels are optimized to perform the necessary matrix-vector operations and speckle tracking. A detailed description of the velocity estimator and its implementation can be found in the previous Chapter 5 and is omitted here. In this work, we devised an additional kernel to remove vectors with poor speckle correlation as in (Ekroll et al., 2016) and replace them with an interpolated value. The vectors were removed if the mean SAD per pixel and frame exceeded 5.333 dB and the interpolated vector was derived as the mean of the velocity vectors in a 7×7 neighborhood around the affected pixel.

4) Image Preprocessing, Display and Vector Flow Visualization

Image preprocessing in replay mode was identical to that in live mode. For image display in replay mode, projectile visualization is used to present the vector flow output in accordance with the vector projectile imaging framework that was previously reported (Yiu et al., 2014). To accommodate the variability of the processing time of the vector flow estimator in replay mode and prevent variable time evolution in the cineloop, the visualization module was decoupled and run in a separate thread with a fixed update rate of 30 fps. Each update, the last available B-mode + speckle image is rendered on screen and the vector positions are updated according to the last available vector map and processing time for that vector map. A C++ class was devised with tables to keep track of all vector

positions and lifetimes in number of frames. At the start of the cineloop, the positions of the arrows are randomly sampled in the imaging view. In subsequent frames, the positions of the vectors are updated using the vector flow map and the temporal resolution of the cineloop. Vectors that do not reside in the flow map or exceed their adjustable lifetime are reset and spawned in a random new location. A consistent vector concentration is maintained by measuring the existing concentration of vectors before a vector is spawned in a particular pixel, then preventing the spawning if the vector concentration is too high.

6.3 Experimental Demonstration

6.3.1 Demonstration Protocol

The framework for real-time aliasing resistant vector flow was demonstrated using self-guided imaging by a healthy volunteer in compliance with COVID-19 regulations. The scanner has been approved for human use by the human research ethics committee at the University of Waterloo. The volunteer used the live mode of the framework to locate target arteries and achieve a desirable imaging window. Once this was found, the play-back mode was enabled and the aliasing resistant vector flow cineloop was computed and displayed frame by frame in real-time.

6.3.2 Imaging Scenarios

Four imaging scenarios were chosen for the demonstration of the efficacy of the framework for imaging flow in bifurcations, excessive aliasing and stenosis conditions. In the first two imaging scenarios, the volunteer scanned their carotid and femoral bifurcations using the scanner while in a seated upright position. The target bifurcations are known to contain non-trivial and transient flow dynamics with aliasing emerging typically due to a low beam-to-flow angle in the deeper branches. To simulate excessive aliasing (two-cycle aliasing in the Doppler frequency measurements from at least one of the transmit-receive steering angles), the volunteer scanned their common carotid artery while tilting the probe to minimize the beam-to-flow angle.

We also demonstrated the performance of the framework in a carotid stenosis phantom to evaluate the framework in pathological conditions. Quantitative aliasing resistance evaluation was done on an anthropomorphic carotid phantom with 50% stenosis (Chee et al., 2016) with a pulsatile flow of blood mimicking fluid achieved using a programmable pump. The phantom resembles the common carotid artery as it splits into the superficial and deep arteries. The vessels were designed to generally have a

diameter of 6 mm, apart from a lesion in the superficial artery where 50% stenosis is induced. This mimics the pathological condition of atherosclerosis in the carotid and creates a clinically relevant testing scenario. The pump was configured to produce a carotid flow profile with a peak flow rate of 10 ml/s at a heartrate of 72 bps. At these conditions and PRF, two-cycle aliasing is seen in the superficial branch near stenosis. Imaging parameters were tuned for each imaging scenario and are tabulated in Table 6.3. The clutter filter was tuned to minimize tissue motion in the rendered profile. In the carotid stenosis phantom, the sharp flow gradients required a smaller block matching kernel size and regularization factor.

Table 6.3
Imaging Parameters for the Four Scenarios.

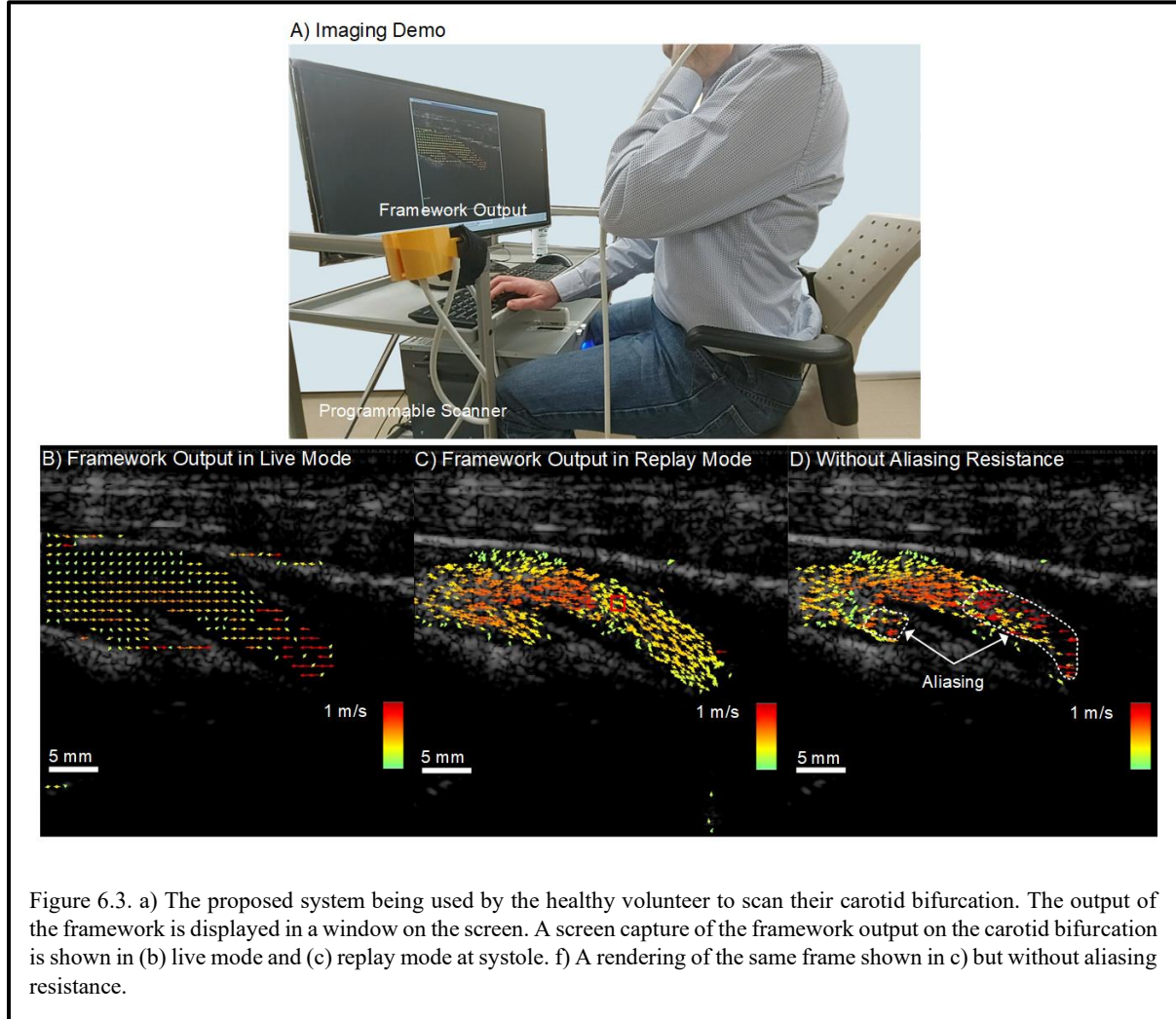
	Clutter filtering, u_c	Block matching window size, B	Regularization factor, λ
Carotid bifurcation	0.16	20	0.5
Femoral bifurcation	0.06	20	0.5
Tilted common carotid	0.06	20	0.5
Carotid stenosis phantom	0.05	10	0.1

6.3.3 Framework Output and Analysis

In this paper, we focus on the demonstration of imaging performance and processing throughput analysis. Quantitative analysis on segmentation performance was reported in the previous Chapter 5. For this study, the imaging sessions by the volunteer were recorded using a camera and screen capture software (OBStudio) installed on the scanner to record the screen at 30 fps. The processing throughput of the framework and key stages in the GPU pipeline were analyzed using the Nvidia Visual Profiler application and CPU timers. This provides information on all GPU kernel calls and memory operations. These operations were divided among the key stages described in 6.2.3, in addition to the overall memory copy operations used throughout the pipeline. After each acquisition, the pre-beamformed RF data was stored and used to generate reference vector flow maps using conventional least squares vector Doppler to showcase the original aliasing that was removed in the framework. The segmentation maps from the aliasing segmentation module and mean Doppler frequency were also extracted and rendered for comparison.

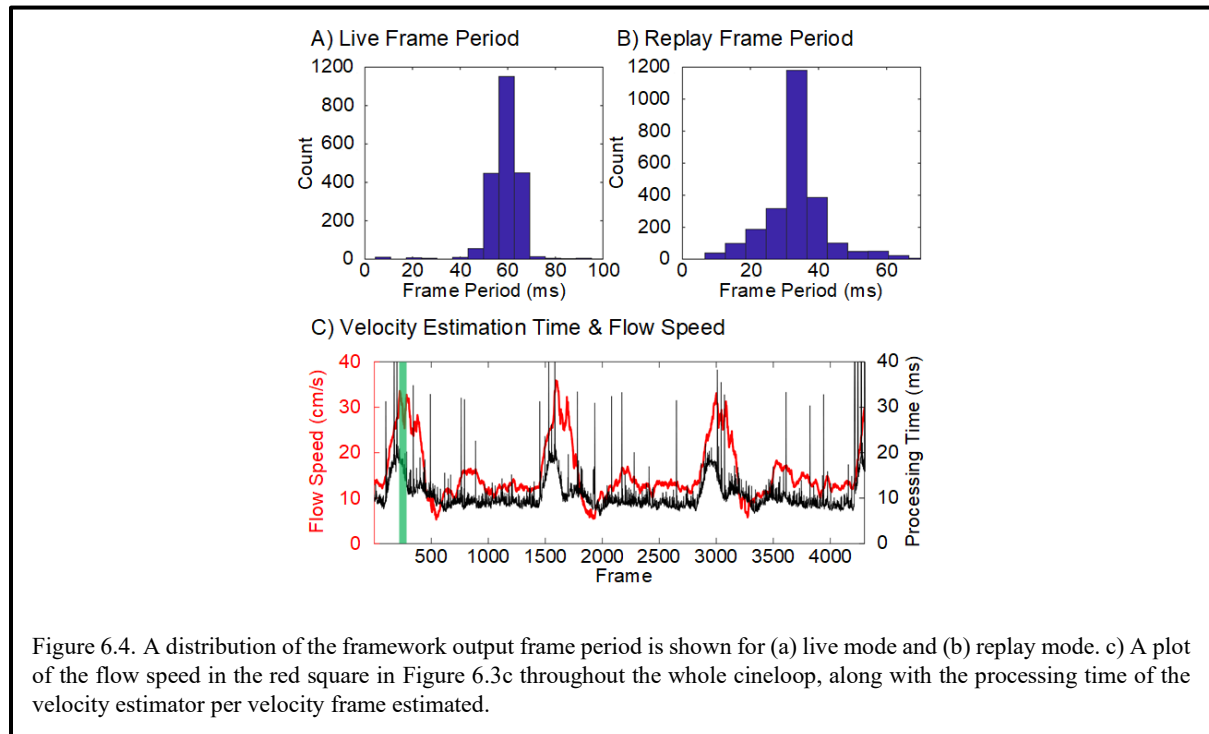
6.4 Results

6.4.1 Real-time Performance Achieved at the Bedside



The proposed framework succeeded in providing guided imaging for vector flow phenomena and real-time aliasing resistant visualization. A demonstration of the framework in action on the carotid bifurcation by the healthy volunteer is shown in Movie 6.1a, where a recording of the imaging setup is presented along with a synchronized capture of the console window in Movie 6.1b. Live vector flow visualization + B-mode was used to align the transducer and identify multi-directional flow. Once the appropriate imaging view was identified, aliasing-resistant playback was triggered to provide a more comprehensive visualization of the flow dynamics in slow-motion. In the movie, we present a synchronized frame-by-frame comparison with vector flow visualization obtained from the

conventional least squares vector Doppler estimator (Yiu & Yu, 2016) without aliasing resistance (Movie 6.1c). A snapshot of this movie is shown in Figure 6.3 with the imaging setup shown in Figure 6.3a and screen captures of the framework output in live mode, replay mode with aliasing resistance, replay mode without aliasing resistance shown in Figure 6.3b-d respectively for the systolic phase. This demonstration showcases the utility of the framework in visualization complex flow dynamics in the carotid bifurcation, which would be degraded without aliasing resistance. Specifically, aliasing can be seen in the systolic phase in Movie 6.1c, Figure 6.3d in both branches of the carotid, while the aliasing free playback (Movie 6.1b, Figure 6.3c) more clearly indicates the change of flow direction throughout the bifurcation. By removing aliasing, the framework allows a clearer identification of flow layer separation and the recirculation zone at the bulb.



The processing throughput of the framework was sufficient to support live vector flow imaging and real-time playback aliasing resistant vector flow imaging. The computational times of key stages in the GPU pipeline described in Sections 6.2.4 and 6.2.5 are shown in Table 6.4. In live mode, throughput analysis was performed on 1972 batches of frames that yielded 1972 output frames of vector flow + B-mode + speckle image. In replay mode, throughput analysis was performed on the processing of 150 batches, 4300 vector flow maps and 2446 visualization frame updates. Figure 6.4a and Figure 6.4b plot the distribution of output frame periods in live and replay mode respectively.

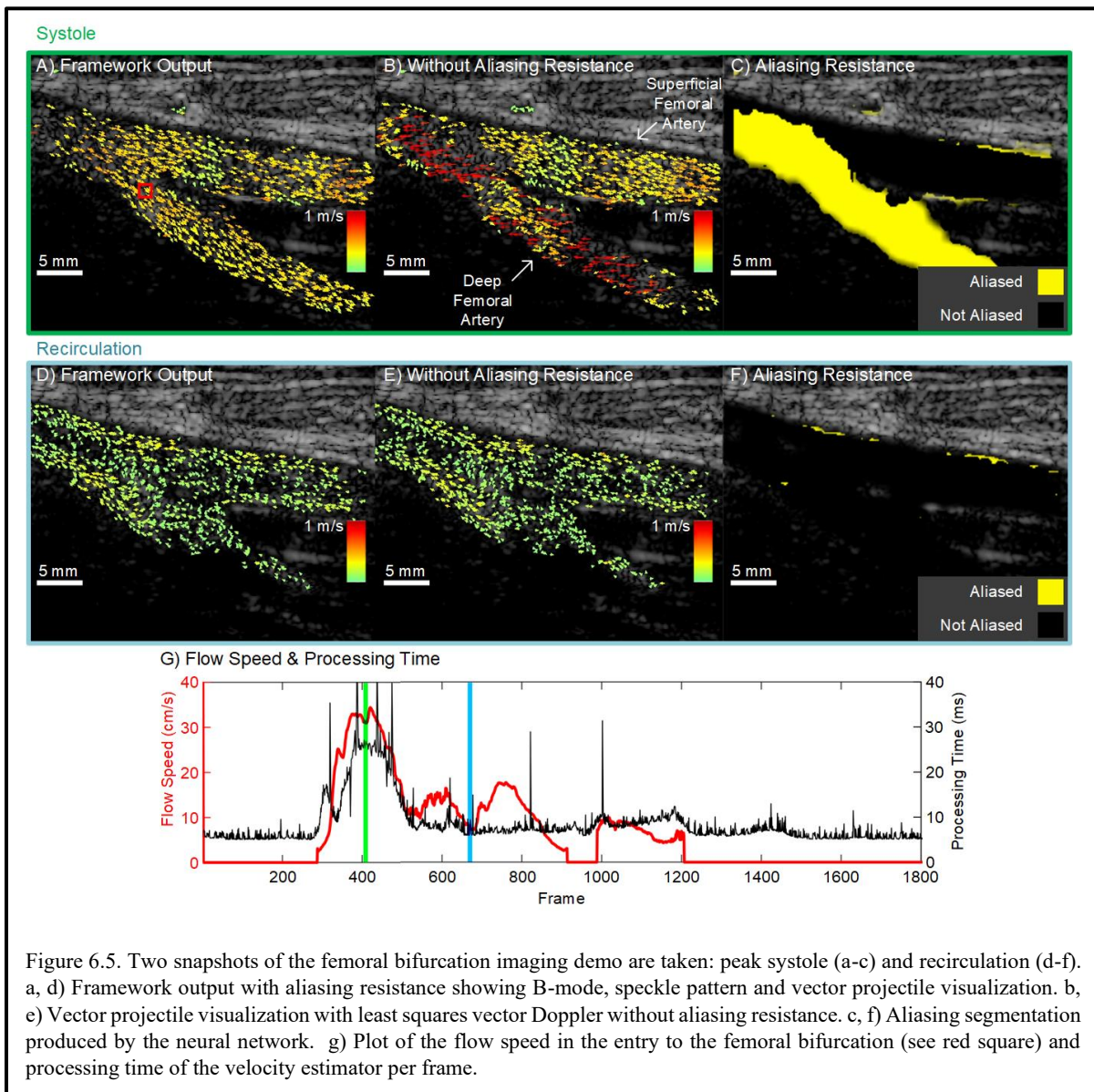
The visualization frame period was measured at 58.9 ± 8.68 ms in live mode (Figure 6.4a) and 33.3 ± 10 ms in replay mode (Figure 6.4b). In Figure 6.4c, we plot the estimated flow speed profile in the red box rendered in Figure 6.3c for the entire stored cineloop consisting of three cardiac cycles, along with the processing time of the velocity estimation phase per frame. While most stages were consistent in processing time between the frames, the vector flow estimator in replay mode was more variable, which is consistent with the strategy described in Section 6.2.5.3. As the systolic phase approaches, the number of aliased pixels in the scene detected by the CNN increases, leading to more application of the aliasing resistant velocity estimator (Figure 6.4c). Some sporadic spikes in the processing time of the velocity estimator were observed due to GPU kernel overlap.

Table 6.4
Processing Times of Key Stages in the Framework for Live and Replay Modes.

		Live	Replay
Data transfer batch to GPU		10.77 $\pm$ 0.35 ms	8.45 $\pm$ 0.35 ms
Image formation of batch		13.71 $\pm$ 0.40 ms	21.96 $\pm$ 0.69 ms
Doppler processing of batch		4.01 $\pm$ 0.22 ms	35.33 $\pm$ 0.91 ms
Vector flow estimation of frame	Least squares vector Doppler estimator	11 $\pm$ 0.8 μ s	27 $\pm$ 2 μ s
	Aliasing segmentation	N/A	4.54 $\pm$ 2.48 ms
	Aliasing resistant estimator	N/A	6.05 $\pm$ 3.34 ms
	Sub-total	11 $\pm$ 0.8 μ s	10.62 $\pm$ 4.45 ms
Image preprocessing of frame		455 $\pm$ 48 μ s	224 $\pm$ 6 μ s

6.4.2 Aliasing Resistant Vector Flow Improves Visualization of Multidirectional Flow

With its aliasing resistant playback, the framework was effective in visualizing multidirectional flow dynamics in the femoral bifurcation. Movie 6.2a shows a screen capture of the scanner during the imaging of the femoral bifurcation of the healthy volunteer, including both operational modes. During replay mode, a synchronized side-by-side comparison is made with the framework output without aliasing resistance (Movie 6.2b) and the aliasing segmentation achieved (Movie 6.2c). The imaging view comprises the common femoral artery as it branches into the deep and superficial femoral arteries. A plot of the processing time of the vector flow estimator and mean flow speed in the red square (Movie 6.1a) throughout the cardiac cycle is also included to capture the dynamic processing time of the aliasing resistant vector flow estimator in Movie 6.2d. A snapshot of Movie 6.2 is shown

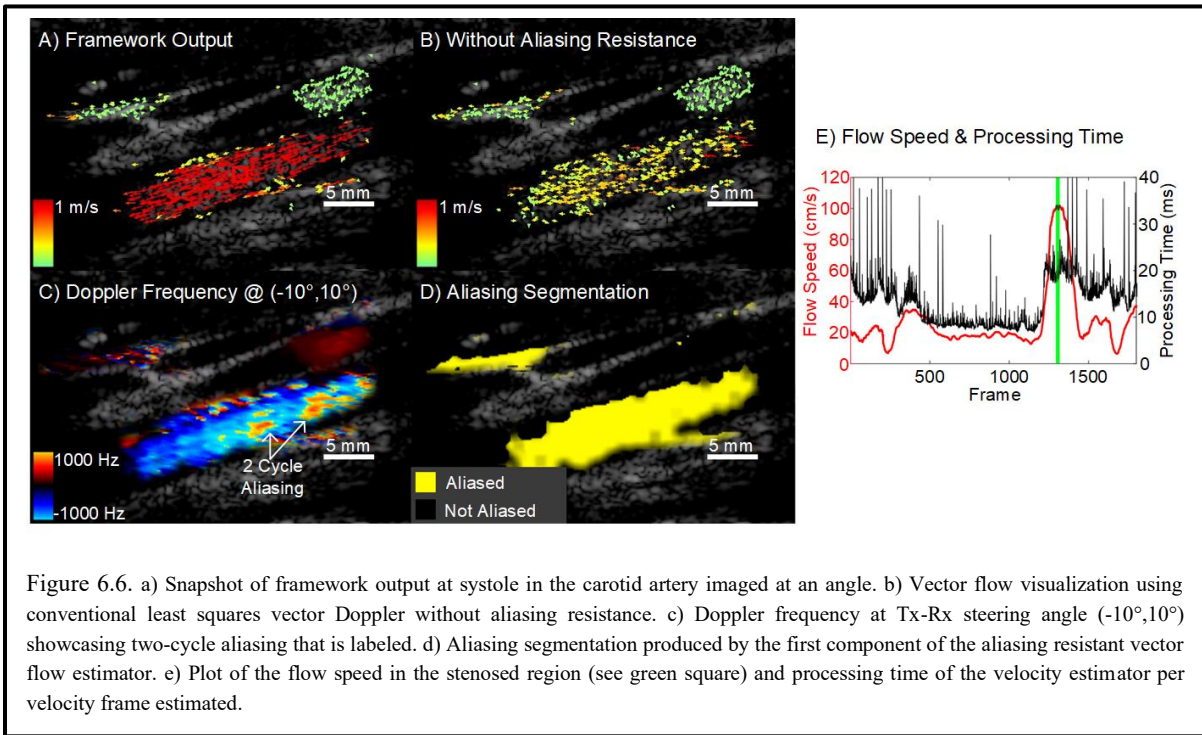


in Figure 6.5, where peak systole and recirculation phases are depicted in the top (a-c) and bottom rows respectively (d-f). The plot shown in Movie 6.2d is presented in Figure 6.5g with the two phases marked. As evident in Movie 6.2 and Figure 6.5, aliasing resistant vector flow estimation is necessary to visualize the flow dynamics as they change direction and flow into the deeper branch. Significant aliasing is observed as the systolic phase approaches due to the lower beam-to-flow angle in the deeper branch, leading to an incorrect flow speed and direction rendering (Figure 6.5b). The framework output (Movie 6.2a) illustrates clearly the complex dynamics of flow as blood splits in the branches with some recirculation occurring in the deeper branch. This is followed by flow backing up

from both branches and swirling at the junction (Figure 6.5d), then a further smaller forward phase into the deeper branch. The aliasing segmentation (Movie 6.2c) was able to identify the aliased pixels in systole and was not triggered in non-aliased but complex flow (Figure 6.5f). Throughout the cardiac cycle, the processing time of the estimator was generally under 10 ms, reaching a peak of just under 30 ms at systole when more aliased pixels were identified.

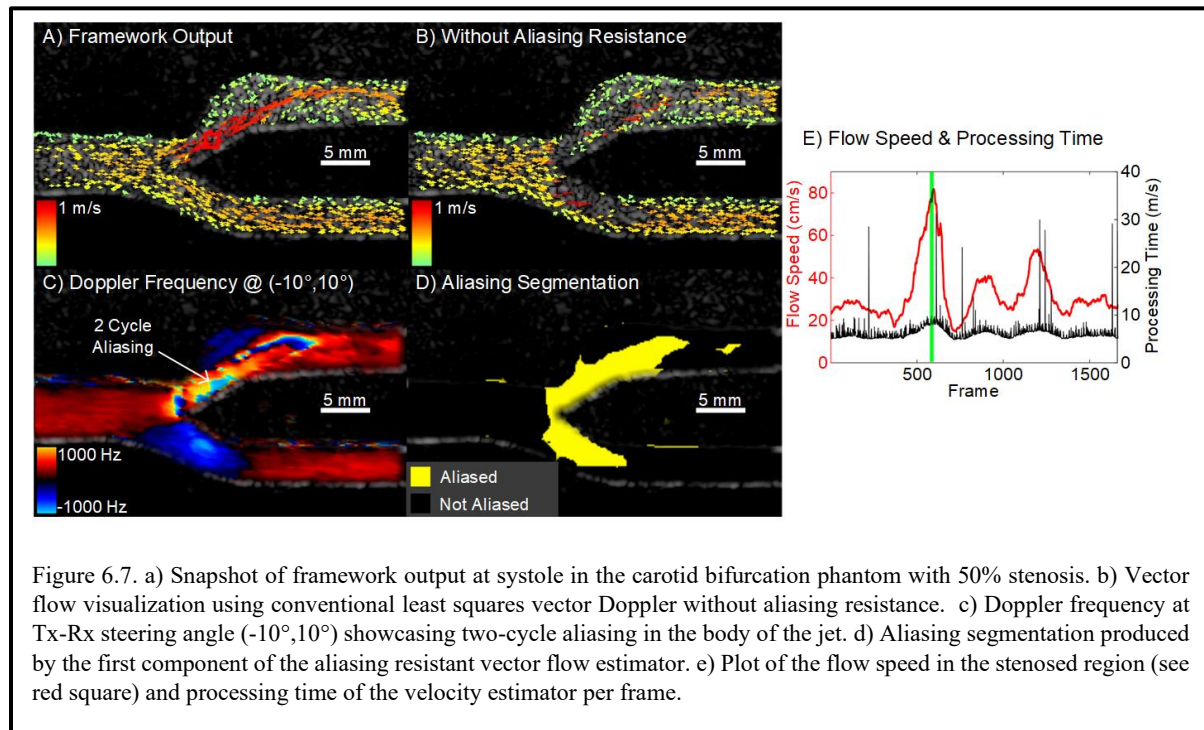
6.4.3 Aliasing Resistant Vector Flow is Possible in Excessive Aliasing Scenarios

The framework was successful in scenarios where two-cycle Doppler aliasing was seen. This was demonstrated in a tilted scan of the carotid bifurcation and in the carotid bifurcation phantom with 50% stenosis. The output cineloops are shown in Movies 6.3 and 6.4 and for the *in vivo* and phantom carotids respectively. The screen recordings of the scanner in replay mode are shown in Movies 6.3a and 6.4a where aliasing resistant visualization is realized. As before, the movies include a side-by-side comparison with the vector flow visualization achieved using conventional least squares vector Doppler estimation (Movies 6.3b and 6.4b) and a rendering of the aliasing segmentation (Movies 6.3d and 6.4d). To emphasize the presence of two-cycle aliasing, we also included the color Doppler map obtained from one of the 7 Tx-Rx combinations (see Table 6.2b) where two-cycle aliasing can be



observed in color flow rendering (Movies 6.3c and 6.4c). Finally, a plot of the processing time per frame for the velocity estimator and flow speed throughout the cardiac cycle is included (Movie 6.3e and 6.4e). Snapshots of the cineloops are shown in Figure 6.6 and Figure 6.7.

In the tilted *in vivo* carotid, framework output succeeded in visualizing flow dynamics in peak systole even when the entire lumen was aliased (Figure 6.6b, c). Moreover, aliasing artifacts persisted beyond the peak of the systolic phase where flow was distinctly non-parabolic (Movie 6.3a, b). Despite this significant aliasing, the framework was still able to achieve a smooth aliasing-free flow profile throughout the cardiac cycle (Movie 6.3a). Meanwhile, the processing time of the aliasing resistant velocity estimator was under 10 ms per frame until systole where it jumped to 25 ms (Figure 6.6e). This represents a worst-case scenario for the framework where the entire flow region is aliased, and no benefit is conferred by the segmentation. Nonetheless, the segmentation allows the overall processing time for the cardiac cycle to be lower by preventing unnecessary applications in other phases of the cardiac cycle where no aliasing is observed.



The framework was able to successfully address aliasing artifacts in the flow jet induced by stenosis in the carotid phantom. As can be seen from Movie 6.4c and Figure 6.7c, the systolic phase features two-cycle aliasing in the superficial branch of the phantom in at least one of the transmit-

receive steering angle combinations, and this was successfully segmented by the CNN (Movie 6.4d and Figure 6.7d) and corrected by the framework (Movie 6.4a and Figure 6.7a). Without this aliasing correction, the vector flow profile in the jet cannot be visualized as can be seen in Movie 6.4b and Figure 6.7b. Some aliasing was also seen in the deeper branch, emerging from aliasing in a different transmit-receive steering angle than shown in Figure 6.7c. The framework was able to visualize flow dynamics throughout the cardiac cycle, tracking the formation of the jet due to stenosis and identifying the adjacent vortices formed. Single-cycle aliasing was seen in the following two forward phases (Figure 6.7e) and these were also addressed by the framework. Throughout the cardiac cycle, the processing time of the velocity estimator fluctuated but was generally lower than 10 ms, with its peaks coinciding with the flow speed profile.

6.5 Discussion

6.5.1 Summary of Contributions

The on-site realization of aliasing-free high frame rate vector Doppler imaging is important for the clinical adoption of this modality. This is because ultrasound imaging is widely expected to be a point-of-care modality (Moore & Copel, 2011), and it is important that vector flow visualization can be performed in real-time. Such a realization cannot afford the prevalence of aliasing artifacts in vector Doppler, and these errors must be removed without compromising the processing throughput of the framework so that point-of-care applications are possible. A number of solutions for Doppler aliasing exist in the literature (Ekroll et al., 2016; Lai et al., 1997; Posada et al., 2016), but a reliable on-site demonstration of two-cycle aliasing resistant HiVFI using vector Doppler has yet to be made.

In this work, we presented an on-site vector Doppler implementation with live imaging and aliasing-resistant slow-motion playback in real-time (Figure 6.1). The pillars of the proposed framework are 1) GPU processing to leverage data-independency inherent throughout the imaging pipeline; 2) Deep learning to realize rapid performance using custom trained networks; 3) A software programmable ultrasound scanner with a fast data-transfer connection and GPU devices. Using these principles, we were able to realize aliasing-resistant vector flow imaging at the bedside at a framerate of 60 fps (Figure 6.4). We showcased this framework on acquisitions from a number of arteries, including the carotid and femoral arteries (Figure 6.3,5-6, Movies 6.1-3). We also highlighted the utility of this framework in conditions of stenosis using an anthropomorphic model (Figure 6.7. Movie 6.4).

A number of vector flow imaging solutions have emerged in the past, utilizing a range of strategies including transverse oscillations, color-doppler based vector flow mapping and vector Doppler (Jensen, J. A. et al., 2016a; Jensen, J. A. et al., 2016b). Additionally, real-time aliasing resistant vector Doppler was demonstrated using staggered transmissions and programmable hardware where velocity estimation exceeding two times the Nyquist limit was reported on a grid of 8×512 pixels (Ricci et al., 2017). Compared to these efforts, the work presented offers the unique advantage of providing aliasing resistant vector Doppler imaging at the bedside on a 256×256 grid with high framerate of 2000 frames per second. By expanding velocity estimation to more than 3 times the Nyquist limit (two-cycle aliasing), the framework can image a wide range of velocities while simultaneously transmitting 30 angles for B-mode imaging, thereby maintaining a useful display of tissue. The presented framework does not utilize beamforming hardware and instead uses GPU processing to achieve real-time performance. This framework also showcases the realization of deep learning at the bedside for the purpose of signal processing towards ultrasound flow imaging. The work demonstrates how such a deep learning-based framework can be developed, including the software and hardware considerations made, and is relevant considering the recent proliferation of deep learning-based solutions in ultrasound imaging (Luchies & Byram, 2018).

6.5.2 Insights on Real-time Performance at the Bedside

The presented work highlights the importance of parallel computing and optimized GPU utilization in realizing high frame rate vector flow imaging at the point-of-care. Each component of the processing pipeline was achieved using dedicated GPU kernels, each optimized for the specific task through minimizing thread divergence and local memory requirements. On the other hand, the availability of dedicated deep learning APIs such as TensorRT allowed the realization of on-site inference with high processing throughput. In addition to GPU parallelization, CPU multi-threading was another instrumental tool in achieving consistent performance on the scanner. Some aspects of the framework could also only be performed using CPU commands, such as the vector projectile visualization management. Decoupling the visualization and velocity estimation threads allowed a more consistent visualization and accommodated the variable processing time of the velocity estimator.

One important consideration in GPU systems is the appropriate scheduling of GPU kernels and utilization of the PCIe link. In this work, kernel overlap was the primary source of most variability in the processing throughput and can be seen in Figure 6.4c, 5g, 6e, 7e. Data transfer is another source of variability as the PCIe stream can only support one stream for data transfer from GPU to CPU and

one other for CPU to GPU. When CPU operations are required in the pipeline, such in the vector flow visualization rendering, there is likely to be stalling hence the variability in the reported visualization frame rates in Section 6.4.

6.5.3 *In Vivo* Applicability of the Proposed Framework

Live vector flow imaging and real-time aliasing resistant playback at the bedside should expand the application of these techniques *in vivo* by overcoming the prohibitive offsite processing. Live vector flow imaging is crucial for identifying and navigating towards complex flow dynamics that may be present in the vasculature (Goddi et al., 2017; Hansen et al., 2016; Hansen et al., 2017), as demonstrated in the bifurcation (Figure 6.3). Once a flow feature is identified by the sonographer, the aliasing-resistant slow-motion playback can then provide a detailed visualization of the flow dynamics and their evolution. This visualization in turn can identify transient phenomena more clearly such as recirculation zones and jets (Movie 6.1,2, 4 and Figure 6.7a) as these features are multidirectional and only a ~ 100 milliseconds long (Au et al., 2020; Goddi et al., 2018). Moreover, aliasing resistance is necessary to capture all velocities in the imaging view at the supported PRF in the scanner. Aliasing in vector Doppler is typically encountered in three kinds of scenarios: 1) Multidirectional flow where some regions have a low beam-to-flow angle with respect to one or more of the steering angles leading to high Doppler frequencies in those angles. 2) High-velocity regions due to alterations in the vasculature geometry, such as in the presence of stenosis (Au et al., 2019; von Reutern et al., 2012). 3) Imaging deeper vasculature where the maximum PRF is limited by the required imaging depth (Hansen et al., 2013; Ishii et al., 2017). We demonstrated the aliasing resistance of this framework in multi-directional flow in the bifurcations (Movies 6.1 and 6.2, Figure 6.3,5) and in stenosis-like conditions (Movie 6.4). Fast-flow conditions are particularly problematic as these may lead to two-cycle aliasing in the Doppler frequency from one or more steering angles. We demonstrated successful aliasing resistance in the presence of two-cycle aliasing in Figure 6.6-7. and Movies 6.3-4. We previously also demonstrated the efficacy of the proposed velocity estimator (Section 6.2.5.3) in an *in vivo* carotid stenosis dataset (Chapter 5).

The proposed framework should pave the way for large-scale human studies by facilitating live guidance for interesting complex flow phenomena and providing complex flow visualization directly for immediate observations at the bedside. High frame rate vector flow cineloops from the scanner can be readily extracted for subsequent analysis, including the vector flow data or the raw pre-beamformed RF data. Indeed, the raw RF data can be reprocessed by the scanner after the imaging

session with different parameters. With these features, the developed system may be used to conduct comprehensive studies on the hemodynamics of flow in healthy and diseased arteries. Flow in the presence of stenosis is dramatically altered (Ha et al., 2015; Hansen et al., 2019; Thiriet et al., 2015), and non-invasively mapping flow in these conditions should provide statistically significant insights on this interaction. Even in a healthy population, flow dynamics can vary widely from person-to-person depending on vasculature geometry (Goddi et al., 2018). An aliasing-resistant vector flow imaging tool at the bedside can more rigorously investigate this relationship in a larger study without suffering from aliasing errors.

6.5.4 Perspectives for Further Validation and Development

With its reliance on deep learning, it will be important to validate the framework on scenarios it was not directly trained on. Examples of important scenarios to investigate would be flow in stenoses with different morphology (Brinjikji et al., 2016), ventricles and in deeper vasculature such as the aorta. The network was already trained on a variety of healthy and diseased arteries, with demonstrated generalizability both in this and previous work (Chapter 5). Moreover, the modularity of the presented algorithm supports alternative networks with different weights or architecture. Indeed, it is possible to implement an on-site training mode for the network model as it encounters new imaging scenarios.

Currently, the scanner can only employ a linear probe for imaging superficial arteries. Meanwhile, there are interesting flow dynamics that can be found in deeper vasculature, such as the aorta (Bogren et al., 1997; Hansen et al., 2017) and the ventricles (Hansen et al., 2013). It should be possible to expand the functionality of the scanner to these imaging scenarios through interfacing the ultrasound module with other probes (convex/phased) and employing alternative transmission schemes, such as diverging waves. It is important to note, however, that the underlying aliasing resistant estimation principles (Ekroll et al., 2016) have been devised for plane wave imaging and further algorithmic work will be necessary for different transmission types.

6.6 Conclusion

Aliasing poses a significant limitation in vector Doppler imaging and has prevented a reliable realization of the modality in the clinic. We demonstrated how high frame rate aliasing resistant vector flow imaging can be realized at the bedside on a programmable ultrasound scanner using GPU processing and Deep learning principles. By resolving aliasing errors, the framework was able to

improve the visualization of hemodynamics in the arteries without compromising the modality's role as a point-of-care tool. This work is important for the clinical adoption of vector Doppler techniques.

Chapter 7

Discussion and Future Directions

7.1 Summary of Contributions

Complex flow dynamics are important biomarkers for cardiovascular health and their mapping is instrumental in clinical monitoring (Grant et al., 2003; Steinman & Taylor, 2005). Clinicians look for disturbed flow in diagnosing conditions including vascular stenosis, aneurysms, and ventricular defects, and use this information in their clinical decision-making (Grant et al., 2003; Warboys, Amini, de Luca, & Evans, 2011). Existing solutions in the clinic for flow imaging are limited in their ability to capture and quantify complex flow conditions. Meanwhile, high frame rate vector Doppler imaging is a next-generation ultrasound-based modality that excels in non-invasively visualizing complex blood flow dynamics *in vivo* in terms of flow speed and direction at superior framerates (Goddi et al., 2018). Despite these favorable characteristics, this modality has faced limited adoption in the clinic, with only one known commercial solution existing for high frame rate (Du et al., 2018). To this end, Doppler aliasing is a fundamental shortcoming that must be addressed reliably without compromising the point-of-care application of ultrasound.

In this dissertation, I have presented a framework for high frame rate vector Doppler imaging suited for bedside applications. The framework succeeded in visualizing blood flow with velocities exceeding 3 times the Nyquist limit (Section 6.4.3) and capture flow dynamics at 2000 fps. The vector flow information was successfully processed and visualized in aliasing resistant playback at an average of 60 fps. This achievement was made on a programmable ultrasound scanner and using GPU-accelerated processing to achieve live and real-time imaging. The framework also showcases the bedside application of deep learning in ultrasound flow imaging, where a high-speed neural network was used to detect flow estimation errors and expand the range of velocities that can be imaged using the framework. With these contributions, this thesis presents an important milestone towards the clinical adoption of vector Doppler imaging. This framework can then drive larger-scale human studies on blood flow dynamics and the role they play in cardiovascular health.

In Chapters 3 and 5, I showcased how aliasing removal in Doppler ultrasound significantly improves flow visualization *in vivo* in both color flow imaging and vector Doppler imaging. This is because aliasing creates regions of flow measurements that are nonsensical and difficult to interpret, especially in scenarios with excessive aliasing (2 cycle aliasing). Despite this difficulty, Chapters 3

and 5 showcase how convolutional neural networks may be leveraged to identify aliased pixels in complex scenarios (see Sections 3.4.2 and 5.6.1). The results also demonstrate how aliasing identification can improve the efficacy of aliasing removal techniques for color flow imaging (Section 3.4.3) and processing throughput for aliasing resistant vector Doppler imaging (Section 5.6.3). The networks were specifically trained on femoral and carotid bifurcations with arterial flow as these vasculatures often contain complex flow dynamics even in healthy conditions (Pedersen et al., 2014) and are prone to plaque-build up due to these flow dynamics (Warboys et al., 2011).

This dissertation presents important contributions in system development and the utilization of GPU computing for ultrasound flow imaging, which is of interest to ultrasound developers. Chapter 5 demonstrates how a real-time aliasing resistance can be achieved using GPU computing principles, and Chapter 6 presents a vector Doppler imaging system on a GPU-powered scanner where aliasing resistant velocity estimation is achieved at 30 fps (Section 6.4.1), including on-site convolutional neural network inference. This work complements other efforts in the ultrasound community that utilize GPU processing to accelerate the innovative but computationally demanding imaging techniques that are emerging in the literature to achieve real-time performance (Goddi et al., 2018; Idzenga et al., 2014; So et al., 2011). This is important for ultrasound imaging as the modality is widely expected to be a point-of-care solution and this accessibility is a key strength that should be preserved (Moore & Copel, 2011).

The contributions of this dissertation are also relevant to ultrasound users who are interested in imaging complex blood flow comprehensively. The devised system may be potentially applied to visualize blood flow in conditions of vascular stenosis and in tortuous vasculature such as bifurcations. Indeed, the carotid bifurcation phantom with stenosis was the testbed of choice in Chapters 4-6 included evaluation on carotid arteries with plaque. The results in Sections 5.6.2 and 6.4.2 reveal how aliasing resistant vector Doppler imaging significantly improves flow visualization in these conditions and more clearly reveals complex dynamics such as jets and recirculation zones. From the perspective of Doppler imaging, vascular stenosis is a challenging scenario to image reliably and past attempts have reported Doppler aliasing as a recurring challenge (Hansen et al., 2017; Hansen et al., 2019; Kronick et al., 2020). Increasing the PRF is usually not feasible due to hardware limitations (Yiu et al., 2019) or undesirable as these flow scenarios may include slow flow in recirculation zones that may be lost at high PRFs (Hwang, 2017).

7.1.1 Aliasing Removal in Color Flow Visualization Using Deep Learning

In Chapter 3, I presented a deep learning-based framework for aliasing removal framework in CFI. This was achieved by: 1) aliasing segmentation by CNN, 2) adaptive phase unwrapping to achieve a continuous flow profile. The CNN was trained on *in vivo* data from the femoral bifurcation with the labels generated using a manual protocol (Section 3.3.3). The results in Chapter 3 showcase the removal of one and two-cycle aliasing in CFI in the femoral bifurcation by using an adaptive phase unwrapping method.

By removing aliasing, the work of Chapter 3 improves the visualization of flow in CFI. CFI is the go-to-modality for point-of-care visualization of flow in the clinic today and has many applications including for stroke, peripheral artery disease and cardiac disease (Oglat et al., 2018). Doppler aliasing is a serious limitation that degrades the quality of visualization, particularly in tortuous vasculature such as bifurcations and in diseased conditions (Gododi et al., 2018). In such scenarios, the combination of flow complexity and aliasing often leads to a ‘mosaic’ of colors that is very challenging to discern (Nishimura et al., 2011). By addressing such conditions in the femoral bifurcation, the work presented in Chapter 3 improves the quality of flow visualization in this commonplace modality and expands the range of velocities that can be imaged simultaneously. This work is relevant for applications where the PRF is limited by depth or the need for visualizing slower flow dynamics or in scenarios with high velocities such as in vascular stenosis (Thiriet et al., 2015).

Compared to existing dealiasing techniques for CFI (Muth et al., 2011), the work done in Chapter 3 makes no assumptions with regards to the shape of the aliasing artifacts. Even when aliasing corrupts most of the flow region, the trained network is able to segment the one and two-cycle aliasing errors (Sections 3.4.2). Additionally, the presented framework does not require multiple different PRFs which can strain the data transfer bandwidth and limit the ensemble length of the system (Posada et al., 2016). This reduces the barrier for clinical adoption in existing scanners in the clinic. Finally, the framework integrates well with high frame rate imaging and is thus suitable for next generation applications (Yiu & Yu, 2013).

7.1.2 Real-time Aliasing Resistant Vector Doppler Using GPU Computing

This dissertation presents the real-time realization of aliasing resistant vector Doppler imaging using GPU processing, the details of which are reported in Chapter 4. This is achieved using single-instruction-multiple-threads (SIMT) principles to accelerate ELS-VD (Ekroll et al., 2016) and bring

down compute times from a reported 5 seconds per frame for two-cycle dealiasing case down to real-time range (25 fps). The results also demonstrated processing throughput under different configurations of the framework and identified the bottlenecks in the pipeline. Aliasing resistance velocity estimation is also successfully demonstrated in a carotid bifurcation phantom with stenosis where a processing time under 16 ms per frame was achieved.

The framework in Chapter 4 was realized in the CUDA platform, a popular choice for general-purpose GPU applications, and on a modest GPU (RTX-2080) that is commercially available. The presented work is therefore accessible to a significant body of researchers who should be able to utilize this methodology and achieve real-time aliasing resistant velocity estimation in their own work. There is growing interest in ultrasound systems supporting GPU processing (So et al., 2011) and thus the presented work can potentially fit in an end-to-end GPU-based pipeline to realize real-time imaging. This work may be potentially integrated with other ultrasound modalities such as wall shear rate imaging and pulse wave imaging to provide more comprehensive analysis of the vasculature without prohibitive processing times (Perrot et al., 2021).

7.1.3 Robust Velocity Estimation with High Processing Throughput

In Chapter 5, a robust vector Doppler estimator is realized with high processing throughput by utilizing the findings of Chapters 3-4. Specifically, a compact neural network is trained to identify pixels in the imaging view that are affected by Doppler aliasing with minimal compute time on a GPU (~7 ms per frame). The GPU-accelerated aliasing resistant estimator from Chapter 4 is then applied exclusively on the aliased pixels, thereby reducing the number of pixels that need to be processed by that estimator and overall compute time per frame. This strategy achieved compute times under 20 ms for most cases and was significantly faster than the GPU accelerated framework alone. This achievement is important as excessive processing times prohibit clinical adoption, prevent point-of-care application and hinder large-scale human studies.

This module showcases the potential of deep learning techniques in accelerating the flow imaging pipeline by approximating complex processes and realizing them at a high inference rate. This approach benefits from the natural relationship between neural networks and GPU computing with many available toolkits for deep learning supporting GPU inference (Shi, Wang, Xu, & Chu, 2016). As GPU efficacy continues to improve and considering efforts to natively support deep learning at the hardware level (Sze, Chen, Yang, & Emer, 2017), it is likely that deep learning strategies to approximate and accelerate processing, as presented in Chapter 4, will become more effective. One

potential extension of this work in the vector Doppler pipeline is using deep learning for clutter filtering (Tabassian, Hu, Chakraborty, & D'hooge, 2019) and flow detection.

7.1.4 Aliasing Resistant Vector Doppler Imaging at the Bedside

Chapter 6 presents a reliable vector Doppler imaging system on a programmable ultrasound scanner, housing the estimator described in Chapter 5. The system has two operational modes: live and replay. In live mode, the scanner renders a live visualization of the vector flow data and can be used for guiding the imaging session. Once a satisfactory imaging window is located, slow-motion replay mode can be triggered where the vector flow cineloop is rendered at 60 fps with aliasing errors removed. This achievement is an important contribution in the translation of vector Doppler techniques to the clinic, where aliasing and excessive processing times would hinder clinical use.

A number of bedside vector flow imaging solutions exist using transverse oscillations, color-doppler based vector flow mapping and vector Doppler (Jensen, J. A. et al., 2016a; Jensen, J. A. et al., 2016b). Real-time aliasing resistant vector Doppler was also demonstrated using staggered transmissions and programmable hardware where velocity estimation exceeding two times the Nyquist limit was reported (Ricci et al., 2017). Compared to these efforts, the work presented offers the unique advantage of providing aliasing resistant vector Doppler imaging at the bedside on a higher resolution grid with a higher framerate and more transmit-receive steering angle combinations, allowing more robust velocity estimation (Yiu & Yu, 2016). The proposed framework was also able to image velocities more than 3 times the Nyquist limit (two-cycle aliasing), while simultaneously transmitting 30 angles for B-mode imaging. The presented framework also achieves its goal using a software-based strategy that leverages GPU processing as opposed to dedicated circuitry for parallel beamforming and Doppler processing.

Another key contribution in this work is demonstrating the feasibility of deep learning directly at the bedside for the purpose of signal processing and flow imaging. The significance of this contribution is amplified by the advent of multiple deep learning techniques in ultrasound imaging, from beamforming (Luchies & Byram, 2018) to super-resolution imaging (Brown et al., 2020), indicating the growing need for deep learning modules in the ultrasound scanner. The programmable system used in this work is commercially available to ultrasound researchers and the system reported here may be readily replicated for a growing body of ultrasound researchers.

7.2 Limitations and Future Directions

7.2.1 Deep Learning Generalizability

The data used in training the deep learning models used in this thesis were acquired from femoral and carotid bifurcations, including healthy and diseased conditions, but it will be important to consider other scenarios not included in training for the practical use of this framework. This includes other vessels such as the brachial artery or the aorta and other vasculature such as the ventricles, where aliasing is encountered (Hansen et al., 2016; Mehregan et al., 2014). Vascular stenosis is also highly variable in cause and morphology (Brinjikji et al., 2016; Thiriet et al., 2015), so it is important to evaluate the generalizability of the trained models on a more extensive set of vascular stenosis data.

These considerations were taken into account in the development of the system reported in this dissertation by ensuring modularity in the processing pipeline. The neural network component may be readily updated with new weights or architectures as improved models are realized. Indeed, there is also room for implementing an on-site training mode for the network model as it encounters new imaging scenarios. This is because the system may be configured to generate its own labeled data using the automatic protocol reported in Section 5.2.3.3.

7.2.2 Conditions of Extreme Aliasing

Two-cycle Doppler aliasing is a considerably more challenging problem to tackle, compared to one cycle aliasing. In vector Doppler, two-cycle aliasing exponentially expands the space of aliasing patterns to be considered in ELS-VD which significantly increases the processing time per frame (Section 4.5.2). On the other hand, two-cycle aliasing may also be accompanied by bandwidth aliasing, where the spectrum of the Doppler signal at least partially overlaps with itself (Brands & Hoeks, 1992). This has the effect of artificially increasing the bandwidth of the signal, effectively ‘whitening’ it so that it resembles noise. The spectrum of the Doppler signal may also intercept the stopband of the clutter filter more than once, altering the mean frequency of the signal (Bjaerum, Torp, & Kristoffersen, 2002). This is especially problematic in stenosis scenarios (Sections 5.6.2 and 6.4.3) where the increased flow complexity leads to significant extrinsic spectral broadening (von Reutern et al., 2012). In such conditions, mean Doppler frequency estimation has high variance and considerable bias (Brands & Hoeks, 1992) which leads to failure in the ELS-VD framework (Ekroll et al., 2016). In the discrete least squares error optimization process of ELS-VD, erroneous Doppler estimates that cannot be related to the true estimates by an integer of the PRF lead to the incorrect

selection of the aliasing pattern set. It is best to filter those estimates out for vector flow visualization (Sections 5.5.2 and 6.2.5.3). Despite these challenges, the presented framework in Chapters 3 and 5 successfully segmented pixels affected by two-cycle aliasing (Sections 3.4.2, 5.6.1 and 6.4.3) and aliasing-resistant vector estimation was achieved for two-cycle aliased *in vivo* flow (Section 5.6.2 and 6.4.3).

To address issues pertaining to bandwidth aliasing, strategies that limit spectral bandwidth may be employed (Yu, Steinman, & Cobbold, 2006). For example, a high F-number and long pulse cycle can reduce intrinsic broadening (Cobbold, 2007). Alternatively, spectral broadening can be reduced by minimizing the minimum beam-to-flow angle among all transmit-receive steering angle combinations (Park, Jung, Byun, Kim, & Joo, 2012). Larger transmit steering angles can do this for lateral flow but consideration has to be given to grating lobes. Finally, alternative transmission schemes may be explored to maintain a higher effective PRF/angle so that the bandwidth of the signal may be sufficiently captured.

7.2.3 Expanding the Range of Application

The system reported in this dissertation was tailored for linear probe imaging which is typically suited for superficial arteries such as the carotid and femoral, but there are also interesting flow dynamics in deeper vasculature, such as the aorta (Whitlock & Hundley, 2015) and the ventricles (Rijnberg et al., 2018) that may also experience significant aliasing. Flow in the urinary tract may also be a fruitful application of vector flow imaging that requires deeper imaging (Ishii et al., 2017). An important further direction is therefore to translate the framework to other imaging probes and expand the range of applications. This may be done by interfacing the ultrasound module (Section 6.2.1) with other probes (convex/phased) and employing alternative transmission schemes, such as diverging waves (Jensen, J. A. et al., 2016b). The underlying aliasing resistant velocity estimation (Chapter 5) will also have to be extended to different transmission schemes and validated accordingly.

Bibliography

- Au, J. S., Yiu, B. Y. S., So, H., Chee, A. J. Y., Greaves, D. K., Hughson, R. L., & Yu, A. C. H. (2020). Ultrasound vector projectile imaging for detection of altered carotid bifurcation hemodynamics during reductions in cardiac output. *Medical Physics*, 47(2), 431-440.
- Au, J. S., Yiu, B. Y. S., & Yu, A. C. H. (2019). Case studies in physiology: Visualization of blood recirculation in a femoral artery “trifurcation” using ultrasound vector flow imaging. *Journal of Applied Physiology*, 127(6), 1809-1813.
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481-2495.
- Bae, S., Kang, J., Yoo, J., Yoo, Y., Chang, J. H., & Song, T. (2012). Real-time realization of adaptive dynamic quadrature demodulation on a gpu-based ultrasound imaging system. *In: Proceedings of IEEE International Ultrasonics Symposium, 2012*, 1651-1654.
- Baker, S., & Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3), 221-255.
- Batschelet, E. (1981). *Circular statistics in biology*. New York: Academic Press.
- Bierig, S. M., & Jones, A. (2009). Accuracy and cost comparison of ultrasound versus alternative imaging modalities, including CT, MR, PET, and angiography. *Journal of Diagnostic Medical Sonography*, 25(3), 138-144.

- Bjaerum, S., Torp, H., & Kristoffersen, K. (2002). Clutter filter design for ultrasound color flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 49(2), 204-216.
- Bogren, H. G., Mohiaddin, R. H., Kilner, P. J., Jimenez-Borreguero, L. J., Yang, G. Z., & Firmin, D. N. (1997). Blood flow patterns in the thoracic aorta studied with three-directional MR velocity mapping: The effects of age and coronary artery disease. *Journal of Magnetic Resonance Imaging*, 7(5), 784-793.
- Bohs, L. N., Geiman, B. J., Anderson, M. E., Gebhart, S. C., & Trahey, G. E. (2000). Speckle tracking for multi-dimensional flow estimation. *Ultrasonics*, 38(1), 369-375.
- Boni, E., Yu, A. C. H., Freear, S., Jensen, J. A., & Tortoli, P. (2018). Ultrasound open platforms for next-generation imaging technique development. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 65(7), 1078-1092.
- Brands, P. J., & Hoeks, A. P. G. (1992). A comparison method for mean frequency estimators for doppler ultrasound. *Ultrasonic Imaging*, 14(4), 367-386.
- Brinjikji, W., Huston, J., Rabinstein, A. A., Gyeong-Moon, K., Lerman, A., & Lanzino, G. (2016). Contemporary carotid imaging: From degree of stenosis to plaque vulnerability. *Journal of Neurosurgery*, 124(1), 27-42.
- Brown, K. G., Ghosh, D., & Hoyt, K. (2020). Deep learning of spatiotemporal filtering for fast super-resolution ultrasound imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(9), 1820-1829.

- Buklioska-Ilievska, D., Minov, J., Kochovska-Kamchevska, N., Gigovska, I., Doneva, A., & Baloski, M. (2019). Carotid artery disease and lower extremities artery disease in patients with chronic obstructive pulmonary disease. *Open Access Macedonian Journal of Medical Sciences*, 7(13), 2102-2107.
- Céspedes, I., Huang, Y., Ophir, J., & Spratt, S. (1995). Methods for estimation of subsample time delays of digitized echo signals. *Ultrasonic Imaging*, 17(2), 142-171.
- Chang, L. W., Hsu, K. H., & Li, P. C. (2009). Graphics processing unit-based high-frame-rate color doppler ultrasound processing. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 56(9), 1856-1860.
- Chávez-Cartaya, R. E., Ramirez-Romero, P., Calne, S. R. Y., & Jamieson, N. V. (1994). Laser-doppler flowmetry in the study of in vivo liver ischemia and reperfusion in the rat. *Journal of Surgical Research*, 56(5), 473-477.
- Chee, A. J. Y., Ho, C. K., Yiu, B. Y. S., & Yu, A. C. H. (2016). Walled carotid bifurcation phantoms for imaging investigations of vessel wall motion and blood flow dynamics. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(11), 1852-1864.
- Chee, A. J. Y., Yiu, B. Y. S., & Yu, A. C. H. (2017). A GPU-parallelized eigen-based clutter filter framework for ultrasound color flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(1), 150-163.
- Cheung, C. C. P., Yu, A. C. H., Salimi, N., Yiu, B. Y. S., Tsang, I. K. H., Kerby, B., . . . Dickie, K. (2012). Multi-channel pre-beamformed data acquisition system for research on advanced

- ultrasound imaging methods. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 59(2), 243-253.
- Cikes, M., Tong, L., Sutherland, G. R., & D'hooge, J. (2014). Ultrafast cardiac ultrasound imaging: Technical principles, applications, and Clinical Benefits. *JACC: Cardiovascular Imaging*, 7(8), 812-823.
- Cobbold, R. S. C. (2007). *Foundations of biomedical ultrasound*. New York: Oxford University Press.
- Denarie, B., Tangen, T. A., Ekroll, I. K., Benech, N., Torp, H., Bjåstad, T., & Lovstakken, L. (2013). Coherent plane wave compounding for very high frame rate ultrasonography of rapidly moving targets. *IEEE Transactions on Medical Imaging*, 32(7), 1265-1276.
- Dong, H., Yang, G., Liu, F., Mo, Y., & Guo, Y. (2017). Automatic brain tumor detection and segmentation using U-net based fully convolutional networks. *In: Proceedings of Medical Image Understanding and Analysis, 2017*, 506-517.
- Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., & Pal, C. (2016). The importance of skip connections in biomedical image segmentation. *In: Proceedings of 2nd Workshop on Deep Learning in Medical Image Analysis, 2016*, 179-187.
- Du, Y., Shen, Y., Yiu, B. Y. S., Yu, A. C. H., & Zhu, L. (2018). High frame rate vector flow imaging: Development as a new diagnostic mode on a clinical scanner. *In: Proceedings of IEEE International Ultrasonics Symposium, 2018*, 1-4.

- Dunmire, B., Beach, K. W., Labs, K., Plett, M., & Strandness, D. E. (2000). Cross-beam vector doppler ultrasound for angle-independent velocity measurements. *Ultrasound in Medicine & Biology*, 26(8), 1213-1235.
- Ekroll, I. K., Wigen, M., Fadnes, S., & Avdal, J. (2018). Quantitative vascular blood flow imaging: A comparison of vector velocity estimation schemes. *In: Proceedings of IEEE International Ultrasonics Symposium, 2018*, 1-4.
- Ekroll, I. K., Avdal, J., Swillens, A., Torp, H., & Lovstakken, L. (2016). An extended least squares method for aliasing-resistant vector velocity estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(11), 1745-1757.
- European Carotid Surgery Trialists' Collaborative Group. (1998). Randomised trial of endarterectomy for recently symptomatic carotid stenosis: Final results of the MRC european carotid surgery trial (ECST). *The Lancet*, 351(9113), 1379-1387.
- Evain, E., Faraz, K., Grenier, T., Garcia, D., Craene, M. D., & Bernard, O. (2020). A pilot study on convolutional neural networks for motion estimation from ultrasound images. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12), 2565-2573.
- Evans, D. H., Barrie, W. W., Asher, M. J., Bentley, S., & Bell, P. R. F. (1980). The relationship between ultrasonic pulsatility index and proximal arterial stenosis in a canine model. *Circulation Research*, 46(4), 470-475.
- Fadnes, S., Ekroll, I. K., Nyrnes, S. A., Torp, H., & Lovstakken, L. (2014). Robust angle-independent blood velocity estimation based on multi-angle speckle tracking and vector doppler. *In: Proceedings of IEEE International Ultrasonics Symposium, 2014*, 337-340.

- Ferguson, G. G., Michael, E., Barr, H. W., Patrick, C. G., Barnes, R. W., Christopher, W. M., . . . Barnett, H. J. (1999). The north american symptomatic carotid endarterectomy trial. *Stroke*, 30(9), 1751-1758.
- Flynn, J., Daigle, R., Pflugrath, L., Linkhart, K., & Kaczkowski, P. (2011). Estimation and display for vector doppler imaging using planewave transmissions. *In: Proceedings of IEEE International Ultrasonics Symposium, 2011*, 413-418.
- Foley, W. D., & Erickson, S. J. (1991). Color doppler flow imaging. *American Journal of Roentgenology*, 156(1), 3-13.
- Franz, R. W., Jump, M. A., Spalding, M. C., & Jenkins, J. J. (2013). Accuracy of duplex ultrasonography in estimation of severity of peripheral vascular disease. *The International Journal of Angiology*, 22(3), 155-158.
- Gao, H., Bijmens, N., Coisne, D., Lugiez, M., Rutten, M., & D'hooge, J. (2015). 2-D left ventricular flow estimation by combining speckle tracking with Navier–Stokes-based regularization: An in silico, in vitro and in vivo study. *Ultrasound in Medicine & Biology*, 41(1), 99-113.
- Garcia, J., Barker, A. J., & Markl, M. (2019). The role of imaging of flow patterns by 4D flow MRI in aortic stenosis. *JACC: Cardiovascular Imaging*, 12(2), 252-266.
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. *Eprint arXiv:1704.06857*.

- Garland, M., Le Grand S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., . . . Volkov, V. (2008). Parallel computing experiences with CUDA. *IEEE Micro*, 28(4), 13-27.
- Gasse, M., Millioz, F., Roux, E., Garcia, D., Liebgott, H., & Friboulet, D. (2017). High-quality plane wave compounding using convolutional neural networks. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(10), 1637-1639.
- Gill, R. W. (1985). Measurement of blood flow by ultrasound: Accuracy and sources of error. *Ultrasound in Medicine & Biology*, 11(4), 625-641.
- Goddi, A., Bortolotto, C., Raciti, M. V., Fiorina, I., Aiani, L., Magistretti, G., . . . Calliada, F. (2018). High-frame rate vector flow imaging of the carotid bifurcation in healthy adults: Comparison with color doppler imaging. *Journal of Ultrasound in Medicine*, 37(9), 2263-2275.
- Goddi, A., Bortolotto, C., Fiorina, I., Raciti, M. V., Fanizza, M., Turpini, E., . . . Calliada, F. (2017). High-frame rate vector flow imaging of the carotid bifurcation. *Insights into Imaging*, 8(3), 319-328.
- Goldstein, L. B. (2003). Extracranial carotid artery stenosis. *Stroke*, 34(11), 2767-2773.
- Granata, A., Fiorini, F., Andrulli, S., Logias, F., Gallieni, M., Romano, G., . . . Fiore, C. E. (2009). Doppler ultrasound and renal artery stenosis: An overview. *Journal of Ultrasound*, 12(4), 133-143.
- Grant, E. G., Benson, C. B., Moneta, G. L., Alexandrov, A. V., Baker, J. D., Bluth, E. I., . . . Zierler, R. E. (2003). Carotid artery stenosis: Gray-scale and doppler US Diagnosis—Society of radiologists in ultrasound consensus conference. *Radiology*, 229(2), 340-346.

Ha, H., Choi, W., Park, H., & Lee, S. J. (2015). Effect of swirling blood flow on vortex formation at post-stenosis. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 229(2), 175-183.

Hansen, K. L., Hansen, P. M., Ewertsen, C., Lönn, L., Jensen, J. A. ,., & Nielsen, M. B. (2019). Vector flow imaging compared with digital subtraction angiography for stenosis assessment in the superficial femoral artery - A study of vector concentration, velocity ratio and stenosis degree percentage . *Ultrasound International Open*, 5(2), E53-E59.

Hansen, K. L., Moller-Sorensen, H., Kjaergaard, J., Jensen, M. B., Lund, J. T., Pedersen, M. M., . . . Nielsen, M. B. (2016). Intra-operative vector flow imaging using ultrasound of the ascending aorta among 40 patients with normal, stenotic and replaced aortic valves. *Ultrasound in Medicine & Biology*, 42(10), 2414-2422.

Hansen, K. L., Moller-Sorensen, H., Kjaergaard, J., Jensen, M. B., Jensen, J. A., & Nielsen, M. B. (2017). Aortic valve stenosis increases helical flow and flow complexity: A study of intra-operative cardiac vector flow imaging. *Ultrasound in Medicine & Biology*, 43(8), 1607-1617.

Hansen, K. L., Pedersen, M. M., Møller-Sørensen, H., Kjaergaard, J., Nilsson, J. C., Lund, J. T., . . . Nielsen, M. B. (2013). Intraoperative cardiac ultrasound examination using vector flow imaging. *Ultrasonic Imaging*, 35(4), 318-332.

Hansen, K. L., Udesen, J., Gran, F., Jensen, J. A., & Nielsen, M. B. (2009). In-vivo examples of flow patterns with the fast vector velocity ultrasound method. *Ultraschall in Der Medizin*, 30(5), 471-477.

- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *E-Print arXiv:1512.03385*.
- Hein, I. A., & O'Brien, W. D. (1993). Current time-domain methods for assessing tissue motion by analysis from reflected ultrasound echoes-a review. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 40(2), 84-102.
- Ho, C. K., Chee, A. J. Y., Yiu, B. Y. S., Tsang, A. C. O., Chow, K. W., & Yu, A. C. H. (2017). Wall-less flow phantoms with tortuous vascular geometries: Design principles and a patient-specific model fabrication example. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(1), 25-38.
- Hood, D. B., Mattos, M. A., Mansour, A., Ramsey, D. E., Hodgson, K. J., Barkmeier, L. D., & Sumner, D. S. (1996). Prospective evaluation of new duplex criteria to identify 70% internal carotid artery stenosis. *Journal of Vascular Surgery*, 23(2), 254-262.
- Hoskins, P. R. (2011). Estimation of blood velocity, volumetric flow and wall shear rate using doppler ultrasound. *Ultrasound*, 19(3), 120-129.
- Hu, C., Wu, G., Chang, C., & Li, M. (2021). Acoustic-field beamforming for low-power portable ultrasound. *Ultrasonic Imaging*.
- Hwang, J. Y. (2017). Doppler ultrasonography of the lower extremity arteries: Anatomy and scanning guidelines. *Ultrasonography*, 36(2), 111-119.

- Idzenga, T., Gaburov, E., Vermin, W., Menssen, J., & De Korte, C. L. (2014). Fast 2-D ultrasound strain imaging: The benefits of using a GPU. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(1), 207-213.
- Ishii, T., Yiu, B. Y. S., & Yu, A. C. H. (2017). Vector flow visualization of urinary flow dynamics in a Bladder outlet obstruction model. *Ultrasound in Medicine and Biology*, 43(11), 2601-2610.
- Jensen, J., Villagómez Hoyos, C. A., Stuart, M. B., Ewertsen, C., Nielsen, M. B., & Jensen, J. A. (2017). Fast plane wave 2-D vector flow imaging using transverse oscillation and directional beamforming. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(7), 1050-1062.
- Jensen, J. A. (1993). Range/velocity limitations for time-domain blood velocity estimation. *Ultrasound in Medicine & Biology*, 19(9), 741-749.
- Jensen, J. A. (2017). Directional transverse oscillation vector flow estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(8), 1194-1204.
- Jensen, J. A. (2019a). Estimation of high velocities in synthetic-aperture Imaging—Part I: Theory. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 66(6), 1024-1031.
- Jensen, J. A. (2019b). Estimation of high velocities in synthetic-aperture Imaging—Part II: Experimental investigation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 66(6), 1032-1038.
- Jensen, J. A., Nikolov, S. I., Gammelmark, K. L., & Pedersen, M. H. (2006). Synthetic aperture ultrasound imaging. *Ultrasonics*, 44, e5-e15.

- Jensen, J. A., Nikolov, S. I., Yu, A. C. H., & Garcia, D. (2016a). Ultrasound vector flow imaging-part I: Sequential systems. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(11), 1704-1721.
- Jensen, J. A., Nikolov, S. I., Yu, A. C. H., & Garcia, D. (2016b). Ultrasound vector flow imaging-part II: Parallel systems. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(11), 1722-1732.
- Jiang, J., & Hall, T. J. (2011). A fast hybrid algorithm combining regularized motion tracking and predictive search for reducing the occurrence of large displacement errors. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 58(4), 730-736.
- Johnston, K. W., Baker, W. H., Burnham, S. J., Hayes, A. C., Kupper, C. A., & Poole, M. A. (1986). Quantitative analysis of continuous-wave doppler spectral broadening for the diagnosis of carotid disease: Results of a multicenter study. *Journal of Vascular Surgery*, 4(5), 493-504.
- Joos, P., Porée, J., Liebgott, H., Vray, D., Baudet, M., Faurie, J., . . . Garcia, D. (2018). High-frame-rate speckle-tracking echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 65(5), 720-728.
- Kasai, C., Namekawa, K., Koyano, A., & Omoto, R. (1985). Real-time two-dimensional blood flow imaging using an autocorrelation technique. *IEEE Transactions on Sonics and Ultrasonics*, 32(3), 458-464.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *E-Prints arXiv:1412.6980*.

- Krishnamurthy, R., Cheong, B., & Muthupillai, R. (2014). Tools for cardiovascular magnetic resonance imaging. *Cardiovascular Diagnosis and Therapy*, 4(2), 104-125.
- Kronick, M. D., Chopra, A., Swamy, S., Brar, V., Jung, E., Abraham, C. Z., . . . Moneta, G. L. (2020). Peak systolic velocity and color aliasing are important in the development of duplex ultrasound criteria for external carotid artery stenosis. *Journal of Vascular Surgery*, 72(3), P951-957.
- Ku, D. N., Giddens, D. P., Phillips, D. J., & Strandness, D. E., J. (1985). Hemodynamics of the normal human carotid bifurcation: *in vitro* and *in vivo* studies. *Ultrasound in Medicine and Biology*, 11(1), 13-26.
- Kumamaru, K. K., Hoppel, B. E., Mather, R. T., & Rybicki, F. J. (2010). CT angiography: Current technology and clinical use. *Radiologic Clinics of North America*, 48(2), 213-235.
- Lai, X., Torp, H., & Kristoffersen, K. (1997). An extended autocorrelation method for estimation of blood velocity. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(6), 1332-1342.
- Lantz, J., Gupta, V., Henriksson, L., Karlsson, M., Persson, A., Carlhäll, C., & Ebbers, T. (2018). Intracardiac flow at 4D CT: Comparison with 4D flow MRI. *Radiology*, 289(1), 51-58.
- Lenge, M., Ramalli, A., Tortoli, P., Cachard, C., & Liebgott, H. (2015). Plane-wave transverse oscillation for high-frame-rate 2-D vector flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 62(12), 2126-2137.

- Loizou, C. P., Pattichis, C. S., & D'hooge, J. (2017). *Handbook of speckle filtering and tracking in cardiovascular ultrasound imaging and video*. New Jersey: Institution of Engineering and Technology.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015*, 3431-3440.
- Lu, J., Zou, H., & Greenleaf, J. F. (1994). Biomedical ultrasound beam forming. *Ultrasound in Medicine & Biology*, 20(5), 403-428.
- Lu, L., Shin, Y., Su, Y., & Karniadakis, G. E. (2020). Dying ReLU and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28(5), 1671-1706.
- Luchies, A. C., & Byram, B. C. (2018). Deep neural networks for ultrasound beamforming. *IEEE Transactions on Medical Imaging*, 37(9), 2010-2021.
- Lui, E. Y. L., Steinman, A. H., Cobbald, R. S. C., & Johnston, K. W. (2005). Human factors as a source of error in peak doppler velocity measurement. *Journal of Vascular Surgery*, 42(5), 972.e1-972.e10.
- Madiena, C., Faurie, J., Porée, J., & Garcia, D. (2018). Color and vector flow imaging in parallel ultrasound with sub-nyquist sampling. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 65(5), 795-802.

- Mehregan, F., Tournoux, F., Muth, S., Pibarot, P., Rieu, R., Cloutier, G., & Garcia, D. (2014). Doppler vortography: A color doppler approach to quantification of intraventricular blood flow vortices. *Ultrasound in Medicine & Biology*, 40(1), 210-221.
- Merritt, C. R. B. (1987). Doppler color flow imaging. *Journal of Clinical Ultrasound*, 15(9), 591-597.
- Montaldo, G., Tanter, M., Bercoff, J., Benech, N., & Fink, M. (2009). Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 56(3), 489-506.
- Moon, T. K., & Stirling, W. C. (2000). *Mathematical methods and algorithms for signal processing*. New Jersey: Prentice Hall.
- Moore, C. L., & Copel, J. A. (2011). Point-of-care ultrasonography. *The New England Journal of Medicine*, 364(8), 749-757.
- Muth, S., Dort, S., Sebag, I. A., Blais, M. J., & Garcia, D. (2011). Unsupervised dealiasing and denoising of color-doppler data. *Medical Image Analysis*, 15(4), 577-588.
- Nahas, H., Au, J. S., Ishii, T., Yiu, B. Y. S., Chee, A. J. Y., & Yu, A. C. H. (2020). A deep learning approach to resolve aliasing artifacts in ultrasound color flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12), 2615-2628.
- Nahas, H., Ishii, T., Chee, A. J. Y., Yiu, B. Y. S., & Yu, A. C. H. (2019). Segmentation of aliasing artefacts in ultrasound color flow imaging using convolutional neural networks. In: Karray F., Campilho A., Yu A. (Eds) *Image Analysis and Recognition. ICIAR 2019. Lecture Notes in Computer Science, Vol 11663*. Springer, Cham, 452-461.

Newhouse, V. L., Bendick, P. J., & Varner, W. (1976). Analysis of transit time effects on doppler flow measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence, BME-23*(5), 381-387.

Nikolov, S. I., & Jensen, J. A. (2002). Virtual ultrasound sources in high-resolution ultrasound imaging. In: *Proceedings of Medical Imaging: Ultrasonic Imaging and Signal Processing, 2002*, 4687

Nishimura, K., Okayama, H., Inoue, K., Saito, M., Yoshii, T., Hiasa, G., . . . Higaki, J. (2011). Usefulness of the MOSAIC (measurement of stenosis by aliasing coronary flow) method using transthoracic color doppler echocardiography in unstable angina patients. *International Journal of Cardiology, 151*(2), 170-174.

Nyrnes, S. A., Fadnes, S., Wigen, M. S., Mertens, L., & Lovstakken, L. (2020). Blood speckle-tracking based on High-Frame rate ultrasound imaging in pediatric cardiology. *Journal of the American Society of Echocardiography, 33*(4), 493-503.e5.

Oglat, A. A., Matjafri, M. Z., Suardi, N., Oqlat, M. A., Abdelrahman, M. A., & Oqlat, A. A. (2018). A review of medical doppler ultrasonography of blood flow in general and especially in common carotid artery. *Journal of Medical Ultrasound, 26*(1), 3-13.

Østvik, A., Smistad, E., Espeland, T., Berg, E. A. R., & Lovstakken, L. (2018). Automatic myocardial strain imaging in echocardiography using deep learning. In: *Stoyanov D. Et Al. (Eds) Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA 2018, ML-CDS 2018. Lecture Notes in Computer Science, 11045* 309-316.

- Park, M. Y., Jung, S. E., Byun, J. Y., Kim, J. H., & Joo, G. E. (2012). Effect of beam-flow angle on velocity measurements in modern doppler ultrasound systems. *American Journal of Roentgenology*, 198(5), 1139-1143.
- Pedersen, M. M., Pihl, M. J., Haugaard, P., Hansen, K. L., Lange, T., Lönn, L., . . . Jensen, J. A. (2014). Novel flow quantification of the carotid bulb and the common carotid artery with vector flow ultrasound. *Ultrasound in Medicine & Biology*, 40(11), 2700-2706.
- Pellett, A. A., Tolar, W. G., Merwin, D. G., & Kerut, E. K. (2005). Doppler aliasing. *Echocardiography*, 22(6), 540-543.
- Perrot, V., Ekroll, I. K., Avdal, J., Saxhaug, L. M., Dalen, H., Vray, D., . . . Liebgott, H. (2021). Translation of simultaneous vessel wall motion and vectorial blood flow imaging in healthy and diseased carotids to the clinic: A pilot study. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(3), 558-569.
- Poelma, C. (2016). Ultrasound imaging velocimetry: A review. *Experiments in Fluids*, 58(1), 3.
- Porée, J., Goudot, G., Pedreira, O., Laborie, E., Khider, L., Mirault, T., . . . Pernot, M. (2021). Dealiasing high-frame-rate color doppler using dual-wavelength processing. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(6), 2117-2128.
- Posada, D., Poree, J., Pellissier, A., Chayer, B., Tournoux, F., Cloutier, G., & Garcia, D. (2016). Staggered multiple-PRF ultrafast color doppler. *IEEE Transactions on Medical Imaging*, 35(6), 1510-1521.

- Ricci, S., Bassi, L., Dallai, A., Matera, R., & Tortoli, P. (2017). Real-time staggered PRF for vector doppler blood velocity assessment. *In: Proceedings of IEEE International Ultrasonics Symposium, 2017*, 1-4.
- Ricci, S., Bassi, L., & Tortoli, P. (2014). Real-time vector velocity assessment through multigate doppler and plane waves. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(2), 314-324.
- Rijnberg, F. M., Hazekamp, M. G., Wentzel, J. J., de Koning, P. J. H., Westenberg, J. J. M., Jongbloed, M. R. M., . . . Roest, A. A. W. (2018). Energetics of blood flow in cardiovascular disease: Concept and clinical implications of adverse energetics in patients with a fontan circulation. *Circulation*, 137(22), 2393-2407.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, 9351, 234-241.
- Ross, R. S. (1975). Ischemic heart disease: An overview. *American Journal of Cardiology*, 36(4), 496-505.
- Sanderson, C., & Curtin, R. (2016). Armadillo: A template-based C++ library for linear algebra. *Journal of Open Source Software*, 1(2), 26.
- Sanderson, C., & Curtin, R. (2018). A user-friendly hybrid sparse matrix class in C++. *In: Mathematical Software – ICMS 20*, 422-430.

- Shahin, A., Ménard, M., & Eboueya, M. (2000). Cooperation of fuzzy segmentation operators for correction aliasing phenomenon in 3D color doppler imaging. *Artificial Intelligence in Medicine*, 19(2), 121-154.
- Sheldon, C. D., Murie, J. A., & Quin, R. O. (1983). Ultrasonic doppler spectral broadening in the diagnosis of internal carotid artery stenosis. *Ultrasound in Medicine and Biology*, 9(6), 575-580.
- Shi, S., Wang, Q., Xu, P., & Chu, X. (2016). Benchmarking state-of-the-art deep learning software tools. In: *Proceedings of the 7th International Conference on Cloud Computing and Big Data, 2016*, 99-104.
- Sirmans, D., & Bumgarner, B. (1975). Numerical comparison of five mean frequency estimators. *Journal of Applied Meteorology*, 14(6), 991-1003.
- Smistad, E., Østvik, A., Haugen, B. O., & Lovstakken, L. (2017). 2D left ventricle segmentation using deep learning. In: *Proceedings of IEEE International Ultrasonics Symposium, 2017*, 1-4.
- So, H., Chen, J., Yiu, B. Y. S., & Yu, A. C. H. (2011). Medical ultrasound imaging: To GPU or not to GPU? *IEEE Micro*, 31(5), 54-65.
- Sorensen, M. D., Harper, J. D., Hsi, R. S., Shah, A. R., Dighe, M. K., Carter, S. J., . . . Bailey, M. R. (2013). B-mode ultrasound versus color doppler twinkling artifact in detecting kidney stones. *Journal of Endourology*, 27(2), 149-153.
- Spaide, R. F., Fujimoto, J. G., Waheed, N. K., Sadda, S. R., & Staurengi, G. (2018). Optical coherence tomography angiography. *Progress in Retinal and Eye Research*, 64, 1-55.

Steinman, D. A., & Taylor, C. A. (2005). Flow imaging and computing: Large artery hemodynamics. *Annals of Biomedical Engineering*, 33(12), 1704-1709.

Swillens, A., Segers, P., Torp, H., & Lovstakken, L. (2010). 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*, 57(2), 327-339.

Swillens, A., Segers, P., & Lovstakken, L. (2010). Two-dimensional flow imaging in the carotid bifurcation using a combined speckle tracking and phase-shift estimator: A study based on ultrasound simulations and *in vivo* analysis. *Ultrasound in Medicine and Biology*, 36(10), 1722-1735.

Sze, V., Chen, Y., Yang, T., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295-2329.

Tabassian, M., Hu, X., Chakraborty, B., & D'hooge, J. (2019). Clutter filtering using a 3D deep convolutional neural network. *In: Proceedings of IEEE International Ultrasonics Symposium, 2019*, 2114-2117.

Tanter, M., & Fink, M. (2014). Ultrafast imaging in biomedical ultrasound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(1), 102-119.

Terslev, L., Diamantopoulos, A. P., Dohn, U. M., Schmidt, W. A., & Torp-Pedersen, S. (2017). Settings and artefacts relevant for doppler ultrasound in large vessel vasculitis. *Arthritis Research & Therapy*, 19(1), 167.

- Thiriet, M., Delfour, M., & Garon, A. (2015). Vascular stenosis: An introduction. *In: Lanzer P. (Eds) PanVascular Medicine.*
- Tremblay-Darveau, C., Bar-Zion, A., Williams, R., Sheeran, P. S., Milot, L., Loupas, T., . . . Burns, P. N. (2017). Improved contrast-enhanced power doppler using a coherence-based estimator. *IEEE Transactions on Medical Imaging*, 36(9), 1901-1911.
- Udesen, J., Nikolov, S. I., & Jensen, J. A. (2005). A simple method to reduce aliasing artifacts in color flow mode imaging. *In: Proceedings of IEEE Ultrasonics Symposium, 2005.* 1352-1355.
- Villagómez Hoyos, C. A., Stuart, M. B., Hansen, K. L., Nielsen, M. B., & Jensen, J. A. (2016). Accurate angle estimator for high-frame-rate 2-D vector flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(6), 842-853.
- von Reutern, G. M., Goertler, M. W., Bornstein, N. M., Del Sette, M., Evans, D. H., Hetzel, A., . . . Neurosonology Research Group of the World Federation of Neurology. (2012). Grading carotid stenosis using ultrasonic methods. *Stroke*, 43(3), 916-921.
- Warboys, C. M., Amini, N., de Luca, A., & Evans, P. C. (2011). The role of blood flow in determining the sites of atherosclerotic plaques. *F1000 Medicine Reports*, 3(5), 1-8.
- Whitlock, M. C., & Hundley, W. G. (2015). Noninvasive imaging of flow and vascular function in disease of the aorta. *JACC: Cardiovascular Imaging*, 8(9), 1094-1106.
- Yap, M. H., Pons, G., Martí, J., Ganau, S., Sentís, M., Zwigelaar, R., . . . Martí, R. (2018). Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 22(4), 1218-1226.

- Ye, J. C., & Sung, W. K. (2019). Understanding geometry of encoder-decoder CNNs. *In: Proceedings of the 36th International Conference on Machine Learning*, 97 7064-7073.
- Yeung, F., Levinson, S. F., & Parker, K. J. (1998). Multilevel and motion model-based ultrasonic speckle tracking algorithms. *Ultrasound in Medicine & Biology*, 24(3), 427-441.
- Yiu, B. Y. S., Lai, S. S. M., & Yu, A. C. H. (2014). Vector projectile imaging: Time-resolved dynamic visualization of complex flow patterns. *Ultrasound in Medicine and Biology*, 40(9), 2295-2309.
- Yiu, B. Y. S., Tsang, I. K. H., & Yu, A. C. H. (2011). GPU-based beamformer: Fast realization of plane wave compounding and synthetic aperture imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 58(8), 1698-1705.
- Yiu, B. Y. S., & Yu, A. C. H. (2013). High-frame-rate ultrasound color-encoded speckle imaging of complex flow dynamics. *Ultrasound in Medicine & Biology*, 39(6), 1015-1025.
- Yiu, B. Y. S., & Yu, A. C. H. (2015). GPU-based minimum variance beamformer for synthetic aperture imaging of the eye. *Ultrasound in Medicine and Biology*, 41(3), 871-883.
- Yiu, B. Y. S., & Yu, A. C. H. (2016). Least-squares multi-angle doppler estimators for plane-wave vector flow imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(11), 1733-1744.
- Yiu, B. Y. S., Walczak, M., Lewandowski, M., & Yu, A. C. H. (2019). Live ultrasound color-encoded speckle imaging platform for real-time complex flow visualization in vivo. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 66(4), 656-668.

- Younis, H. F., Kaazempur-Mofrad, M. R., Chan, R. C., Isasi, A. G., Hinton, D. P., Chau, A. H., . . . Kamm, R. D. (2004). Hemodynamics and wall mechanics in human carotid bifurcation and its consequences for atherogenesis: Investigation of inter-individual variation. *Biomechanics and Modeling in Mechanobiology*, 3(1), 17-32.
- Yu, A. C. H., Steinman, A. H., & Cobbold, R. S. C. (2006). Transit-time broadening in pulsed doppler ultrasound: A generalized amplitude modulation model. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 53(3), 530-541.
- Zhang, Y., Ying, M. T. C., Yang, L., Ahuja, A. T., & Chen, D. Z. (2016). Coarse-to-fine stacked fully convolutional nets for lymph node segmentation in ultrasound images. *In: Proceedings of IEEE International Conference on Bioinformatics and Biomedicine, 2016*, 443-448.

Appendix

This appendix lists the video files referenced in this dissertation.

Movie3.1.mp4: A cineloop (playback rate: 30 fps) of dealiased CFI over the systolic phase of the cardiac cycle in the femoral bifurcation. Shown in this movie is a side-by-side comparison between the original CFI (left column), the dealiased CFI (middle column), and the CNN-derived aliasing segmentation map (right column). Flow moves from the common artery in the left and splits into the superficial (top right) and deep (bottom right) femoral arteries, exhibiting some recirculation and aliasing in the deeper artery. The video shows how the proposed segmentation framework can successfully track one and two-cycle aliasing, and how the overall dealiasing framework can resolve excessive aliasing scenarios in the bifurcation. See Section 3.4.3 for more details.

Movie5.1.mp4: Demonstration of aliasing removal in the 50% stenosis carotid bifurcation phantom with two-cycle aliasing, including a render of the vector flow cineloop derived using (a) conventional LS-VD at $\text{PRF} = 750 \text{ Hz/angle}$, (b) ELS-VD at $\text{PRF} = 750 \text{ Hz/angle}$, (c) the proposed framework at $\text{PRF} = 750 \text{ Hz/angle}$ and (d) LS-VD at high $\text{PRF} = 3000 \text{ Hz/angle}$. The movie also shows (e) the aliasing segmentation by the CNN and (f) the mean Doppler frequency map from one of the transmit-receive steering angle combinations to emphasize aliasing contours and order. The phase of the cardiac cycle can be tracked using a (g) plot of the flow speed in the white square included with the movies. Blood flow is shown moving from the common carotid artery in the left to the superficial (top) and deep arteries in the right, with two-cycle aliasing seen at systole due to jet formation near the stenosis. This video demonstrates how the proposed velocity estimator captured vector Doppler aliasing in the diseased carotid bifurcation phantom and produced an aliasing-free profile with reduced compute time per frame. See Section 5.6.2 for more details.

Movie5.2.mp4: Demonstration of aliasing removal in the femoral bifurcation, including a render of the vector flow cineloop at $\text{PRF} = 3000 \text{ Hz/angle}$ derived using (a) conventional LS-VD, (b) ELS-VD, and (c) the proposed framework. The movie also shows (d) the aliasing segmentation by the CNN and (e) the mean Doppler frequency map from one of the transmit-receive steering angle combinations to emphasize aliasing contours and order. The phase of the cardiac cycle can be tracked using a (f) plot of the flow speed in the white square included with the movies. Blood flow is shown moving from the common femoral artery in the left to the superficial (top) and deep arteries in the right, with aliasing seen at systole. This video demonstrates how the proposed velocity estimator

captured vector Doppler aliasing in the femoral bifurcation and produced an aliasing-free profile with reduced compute time per frame. See Section 5.6.2 for more details.

Movie5.3.mp4: Demonstration of aliasing removal in a carotid artery with plaque, including a render of the vector flow cineloop at PRF = 3000 Hz/angle derived using (a) conventional LS-VD, (b) ELS-VD, and (c) the proposed framework. The movie also shows (d) the aliasing segmentation by the CNN and (e) the mean Doppler frequency map from one of the transmit-receive steering angle combinations to emphasize aliasing contours and order. The phase of the cardiac cycle can be tracked using a (f) plot of the flow speed in the white square included with the movies. Blood flow is shown moving in the carotid artery from the left to right, with aliasing seen at systole due to jet formation near the plaque. This video demonstrates how the proposed velocity estimator captured vector Doppler aliasing in the plaque-affected carotid artery and produced an aliasing-free profile with reduced compute time per frame. See Section 5.6.2 for more details.

Movie5.4.mp4: Demonstration of aliasing removal in a tilted carotid artery at a PRF of 2000 Hz/angle with two-cycle aliasing, including a render of the vector flow cineloop derived using (a) conventional LS-VD, (b) ELS-VD, and (c) the proposed framework. The movie also shows (d) the aliasing segmentation by the CNN and (e) the mean Doppler frequency map from one of the transmit-receive steering angle combinations to emphasize aliasing contours and order. The phase of the cardiac cycle can be tracked using a (f) plot of the flow speed in the white square included with the movies. Blood flow is shown moving in the carotid artery from the bottom left to top right, with two-cycle aliasing seen at systole due to jet formation near the plaque. This video demonstrates how the proposed velocity estimator captured vector Doppler aliasing in the tilted carotid artery and produced an aliasing-free profile with reduced compute time per frame. See Section 5.6.2 for more details.

Movie6.1.mp4: A demonstration of the framework in action on the carotid bifurcation by the healthy volunteer is shown in (a), where a recording of the imaging setup is presented along with a synchronized capture of the console window in (b). Live vector flow visualization + B-mode was used to align the transducer and identify multi-directional flow. Once the appropriate imaging view was identified, aliasing-resistant playback was triggered to provide a more comprehensive visualization of the flow dynamics in slow-motion. (c) shows a synchronized frame-by-frame comparison with vector flow visualization obtained from the conventional least squares vector Doppler estimator without aliasing resistance. Blood flow is shown moving from the common carotid artery in the left to superficial and deep branches, with aliasing seen at systole in both branches. This

video demonstrates the proposed framework in a live imaging session with live mode for guidance and replay mode for aliasing-free imaging. See Section 6.4.1 for more details.

Movie6.2.mp4: A screen capture of the scanner during the imaging of the femoral bifurcation of the healthy volunteer, including both operational modes, is shown in (a). During replay mode, a synchronized side-by-side comparison is made with (b) the framework output without aliasing resistance and (c) the aliasing segmentation achieved. The imaging view comprises the common femoral artery as it branches into the deep and superficial femoral arteries. A plot of the processing time of the vector flow estimator and mean flow speed in the red square in (a) throughout the cardiac cycle is also included in (d) to capture the dynamic processing time of the aliasing resistant vector flow estimator. Blood flow is shown moving from the femoral carotid artery in the left to superficial and deep branches, with aliasing seen at systole. This video demonstrates the utility of the proposed framework in providing complex flow visualization at the bedside. See Section 6.4.2 for more details.

Movie6.3.mp4: The output of the framework on the tilted carotid with two-cycle aliasing. The screen recordings of the scanner in replay mode are shown in (a) where aliasing resistant visualization is realized. (b) shows a side-by-side comparison with the vector flow visualization achieved using conventional least squares vector Doppler estimation; (c) the color Doppler map obtained from one of the 7 Tx-Rx combinations where two-cycle aliasing can be; (d) shows a rendering of the aliasing segmentation produced by aliasing detection module. A plot of the processing time per frame for the velocity estimator and flow speed throughout the cardiac cycle is included in (e). Blood flow is shown moving in the carotid artery from the top right to the bottom left, with two-cycle aliasing seen at systole. This video demonstrates the utility of the proposed framework in a scenario with excessive aliasing at the bedside. See Section 6.4.3 for more details.

Movie6.4.mp4: The output of the framework on the carotid bifurcation phantom with 50% stenosis. The screen recordings of the scanner in replay mode are shown in (a) where aliasing resistant visualization is realized. (b) shows a side-by-side comparison with the vector flow visualization achieved using conventional least squares vector Doppler estimation; (c) the color Doppler map obtained from one of the 7 Tx-Rx combinations where two-cycle aliasing can be; (d) shows a rendering of the aliasing segmentation produced by aliasing detection module. A plot of the processing time per frame for the velocity estimator and flow speed throughout the cardiac cycle is included in (e). Blood flow is shown moving from the common carotid artery in the left to the superficial and deep branches, with two-cycle aliasing seen at systole near the stenosed region. This

video demonstrates the utility of the proposed framework in a scenario resembling pathological flow. See Section 6.4.3 for more details.

If you accessed this thesis from a source other than the University of Waterloo, you may not have access to this file. You may access it by searching for this thesis on <https://uwspace.uwaterloo.ca/>