

REAL-TIME ELASTOGRAPHY SYSTEMS

by

Nishikant P. Deshmukh

A dissertation submitted to Johns Hopkins University in conformity with
the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2016

© 2016 Nishikant Deshmukh

All Rights Reserved

Abstract

Ultrasound elastography is a technique that is often used to detect cancerous tumors and monitor ablation therapy by detecting changes in the stiffness of the underlying tissue. This technique is a computationally expensive due to the extensive searching between two raw ultrasound images, which are called radio frequency images. This thesis explores various methods to accelerate the computation required for the elastography technique to allow use during surgery.

This thesis is divided into three parts. We begin by exploring acceleration techniques, including multithreading techniques, asynchronous computing, and acceleration of the graphics processing unit (GPU). Elastography algorithms are often affected by out-of-plane motion caused by several external factors, such as hand tremors and incorrect palpation motion, amongst others. In this thesis, we implemented an end-to-end system that integrates an external tracker system to detect the in-plane motion of two radio frequency (RF) data slices. This in-plane detection helps to reduce de-correlated RF slices and produces a consistent elastography output. We also explore the integration of a da Vinci Surgical Robot to provide stable palpation motion during surgery.

The external tracker system suffers from interference caused by ferromagnetic materials present in the operation theater in the case of an electromagnetic tracker, while optical and camera-based tracking systems are restricted due to human, object and patient interference in the path of sight and complete or partial occlusion of the tracking sensors. Additionally, these systems must be calibrated to give the position of the tracked objects with respect to

the trackers. Although calibration and trackers are helpful for inter-modality registration, we focus on a tracker-less method to determine the in-plane motion of two RF slices. Our technique divides the two input RF images into regions of interest and performs elastography on RF lines that encapsulate those regions of interest.

Finally, we implemented the world's first known five-dimensional ultrasound system. We built the five-dimensional ultrasound system by combining a 3D B-mode volume and a 3D elastography volume visualized over time. A user controlled multi-dimensional transfer function is used to differentiate between the 3D B-mode and the 3D elastography volume.

Advisors

Dr. Emad M. Boctor

Dr. Gregory D. Hager

Dr. Russell H. Taylor

Assistant Professor

Mandell Bellmore Professor

John C. Malone Professor

Department of Radiology
Division of Medical Imaging
Physics
The Johns Hopkins Medical
Institutions

Department of Computer Science
The Johns Hopkins University

Department of Computer Science
The Johns Hopkins University

Acknowledgements

I am forever indebted to my advisors Dr. Emad Boctor, Dr. Gregory Hager, and Dr. Russell Taylor for valuable input during my studies.

Thanks to Dr. Emad Boctor for introducing me to the field of ultrasound research and supporting my research. The interesting out-of-the-box problems and solutions in most of your projects helped me to pursue something similar. Thank you for the equipment and the animal data from your laboratory that helped me to complete my research.

Thanks to Dr. Gregory Hager for your insightful comments and enthusiastic encouragement to pursue innovative work. Thanks for pushing me to get the basics of computer vision right during the Computer Vision course and graduate board oral (GBO) preparations. Thank you for all of your hard work during your tenure as the Chair of the Department of Computer Science.

I would like to thank Dr. Russell Taylor for his mentorship during the Computer Integrated Surgery (CIS) course when I was a student and teaching assistant in all versions of the class offerings. Your curiosity and your interest in almost all forms of research in computer science always amazed me. Thank you for introducing me to the eye surgery project and numerous projects via CIS courses.

Special thanks to Dr. Scott Smith for being a great listener and mentor. I will always remember all of your help. As the Chair of Computer Science, your exceptional idea to reduce stress for international students with a tuition waiver for the second year of master's

study helped numerous students. In addition, thank you for your support for the ACM and the graduate lounge.

My graduate studies spanned three Computer Science Chairs, and I would like to thank Drs. Scott Smith, Gregory Hager, and Yair Amir for their valuable input and support as the Chairs of the Department of Computer Science. The support from the Department of Computer Science in the form of Teaching Assistant and Instructor positions is one of the reasons that this thesis was possible. Special thanks to Drs. Joanne Selinski, Peter Fröhlich, and Jason Eisner for invaluable input concerning how to be an effective educator.

Special thanks to Dr. Rao Kosaraju for being my first advisor when I started at The Johns Hopkins University as a Master's student. Thank you for the Randomized Algorithms class and your insights into the development of algorithms. Thanks for the opportunity to be the teaching assistant for the Automata Theory class. Thanks to Dr. Randal Burns for the Parallel Programming class and for intellectual and hardware help in my initial years. Thanks for giving me the opportunity to complete independent study with you on cloud computing. In addition, thanks for the opportunity to be the teaching assistant for the Parallel Programming class in my first year of the Ph.D. program. Thanks to Dr. Matthew Bolitho, who was an excellent teaching assistant for the Parallel Programming class. Thanks to Dr. Marcus Chen for the opportunity to be the teaching assistant for his Wireless Sensor Networks class. Thank you, Dr. Vladimir Braverman, for the opportunity to be the teaching assistant for the Randomized Algorithms class. Finally, thank you Dr. Peter Fröhlich for the opportunity to be the teaching assistant for Computer Systems Fundamentals and the co-instructor for the Intermediate Programming-C/C++ module.

Thanks to Dr. Jerry Prince for his amazing and in-depth courses Medical Imaging Systems and Medical Image Analysis.

I would like to acknowledge Drs. Hassan Rivaz, Ioana Fleming and Pezhman Foroughi for their valuable input in the field of elastography and for being good mentors. Thanks to Dr. Daniel Carnegie for help with the animal experiments. I would like to thank Dr. Hyun Jae Kang, Alexis Cheng, Seth D. Billings and Xiayuo Gao for being excellent peers and collaborators. Thank you, Dr. Muyinatu A. Lediju Bell for being an excellent mentor and for your help in writing my thesis proposal during my GBO. Thanks to Dr. Uday Kamath and Dr. Balaji Krishnamachary for providing feedback on this thesis.

I would like to thank Dr. Jesus Caban, Dr. Philipp Stolka, Dr. Terry S. Yoo, Dr. Marcin Balicki, Dr. Omar Ahmad, Dr. Eric Perlman, Dr. Paul Stanton, Balázs P. Vágvölgyi, Joyce Choi, Tina Vashistha, Joseph Wang and David Mann for their valuable collaborations. Thanks to all of the teaching assistants and the faculty with whom I was a teaching assistant for their valuable insights.

I would like to thank Drs. Arvind Pathak, Eric Frey, Peter Kazanzides, Michael Choti, and Benjamin Tsui for serving on my GBO committee.

I would like to thank my colleagues and friends in the student group's Association for India's Development at The Johns Hopkins University and the Graduate Representative Organization for being a source of inspiration and providing an amazing social service. Special thanks to Catherine Thornton, Debbie DeFord, Laura Graham, Steve Rifkins, Steve DeBlasio, Alison Morrow, Christine Kavanagh, and Zachary Burwell.

Finally, greatest thanks to my family members, my father Mr. Pramod Deshmukh, my mother Mrs. Jayashri Deshmukh, my brother Anand Deshmukh and his wife Bhagyashri Deshmukh, for being a pillar of strength during these long years of studies.

Thanks to NIH SBIR Grant (PI: Everett Burdette) CA134169, NIH Grant CA112852, Intuitive Surgical, Inc., and The Johns Hopkins University for internal funds that supported this work.

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Statement	3
1.3	Hypothesis	4
1.4	Contributions	5
1.5	Outline	7
2	Physics of Ultrasound.....	11
2.1	Introduction	11
2.2	Wave Propagation	13
2.2.1	Acoustic Impedance.....	13
2.2.2	Acoustic Energy and Intensity	14
2.2.3	Reflection and Refraction at Plane Interfaces.....	15
2.3	Ultrasound Instrumentation	21
2.3.1	Ultrasound Transducer.....	21
2.3.2	Piezoelectric Materials.....	21
2.3.3	Resonance Transducers.....	22
2.3.4	Damping Block.....	23
2.3.5	Matching Layer.....	24
2.3.6	Non-resonance (Broad-Bandwidth) “Multi-frequency” Transducers.....	24
2.4	Transducer Arrays	25
2.4.1	Linear Arrays	26
2.4.2	Phased Arrays	26

2.5	Ultrasound Probes.....	26
2.6	Pulse-Echo Imaging.....	27
2.6.1	The Pulse-Echo Equation.....	27
2.6.2	Transducer Motion.....	28
2.7	Ultrasound Imaging Modes.....	31
2.7.1	A-mode ultrasound scan	31
2.7.2	B-mode ultrasound scan.....	31
2.7.3	M-mode ultrasound scan.....	32
2.8	Steering and Focusing.....	32
2.8.1	Transmit Steering and Focusing	33
2.8.2	Beamforming and Dynamic Focusing	34
2.9	Image Quality	35
2.9.1	Resolution	35
2.9.2	Noise, Artifacts, and Speckle.....	36
2.10	Elastography	39
3	Motion estimation algorithms for displacement and strain estimation in elastography.....	45
3.1	Normalized cross-correlation (NCC), sum of squared difference (SSD), and sum of absolute difference (SAD)	45
3.1.1	Spatial Domain.....	45
3.1.2	Spectral Domain.....	46
3.1.3	3D cross-correlation method.....	48
3.2	Optical Flow.....	49
3.3	Phase-shift correlation method	51
3.4	Efficient phase zero search algorithm	52

3.5	Dynamic Programming	53
3.6	Analytical Minimization	55
4	Elastography using Multi-Stream GPU.....	58
4.1	Introduction	62
4.2	Background	65
4.2.1	General Purpose Graphic Processing Units (GPGPU)	65
4.2.2	Normalized Cross-Correlation (NCC) based Elastography	66
4.2.3	Tracked Ultrasound Elastography (TRuE)	67
4.3	Methods	69
4.3.1	Ethical Statement	69
4.3.2	System Overview	70
4.3.3	Multi-Stream GPU-based Elastography	73
4.3.4	Online Tracked Ultrasound Elastography.....	74
4.3.5	O-TRuE Image Fusion.....	75
4.4	Experiments	85
4.4.1	Phantom Experiments	85
4.4.2	In-Vivo Animal Experiments.....	87
4.4.3	da Vinci Surgical System Experiments.....	88
4.5	Results	92
4.5.1	Speed Analysis.....	92
4.5.2	Validation of O-TRuE Frame Selection	94
4.5.3	O-TRuE Image Fusion Evaluation.....	95
4.5.4	Elastography Image Stream Analysis	96
4.5.5	In-Vivo Animal Experiments.....	101
4.5.6	da Vinci Surgical Robot Palpation Analysis.....	104

4.6	Discussion	105
5	Elastography image pair selection via elastography computation on limited regions of interest	110
5.1	Introduction	111
5.1.1	Motivation	111
5.1.2	Statements of Significance.....	112
5.1.3	Contributions.....	113
5.2	Methods	114
5.3	Results and Discussion	118
5.3.1	Experiments	118
5.3.2	Results	120
5.3.3	Discussion	136
6	Five-Dimensional Ultrasound System for Soft Tissue Visualization.....	143
6.1	Introduction	148
6.1.1	General-purpose graphics processing unit (GPGPU)	151
6.1.2	Normalized cross-correlation (NCC)-based elastography on GPU	152
6.2	Methods	153
6.2.1	Five-dimensional ultrasound system.....	153
6.2.2	Multi-threaded 3D ultrasound elastography	154
6.2.3	3D Scan conversion	156
6.2.4	Five-dimensional ultrasound visualizer transfer function.....	157
6.3	Experiments	164
6.4	Results	166
6.5	Discussion	170
7	Conclusions and Future Work.....	177

7.1	Conclusions	177
7.2	Future Work.....	180
8	Bibliography	182
9	Appendix A.....	200
9.1	3D scan conversion on GPU	200
10	Appendix B.....	203
10.1	Data for multi-stream GPU elastography	203
10.2	Data for five-dimensional ultrasound system	203
10.3	Data for elastography image pair selection system by computing elastography on limited regions of interest.....	204

List of Figures

Figure 2.1 Elastography concept: Panel (A) shows the comparison of two RF-lines pre- and post-compression, that gives a strain profile to indicate the stiffness inside the tissue. Panel (B) shows a side-by-side comparison of elastography and B-mode data [2]. The ablated region is clearly visible in the elastography image, and the B-mode image shows the boundary between two layers [2].42

Figure 4.1 Overall System Diagram: The figure shows overall system and data flow diagram of elastography image server which runs on a machine equipped with a GPU. The system is modular with each module configurable to run on different machines or on a same machine (exception is hardware dependent da Vinci surgical system, RF Server and EM Tracker Server). The elastography image server is based on multi-stream elastography algorithm and with little change can handle both tracked and untracked RF data. The MUSiC Sync synchronizes tracking and RF data based on timestamp to be processed by elastography image server. The system is flexible to be connected with da Vinci Surgical console to allow overlay of elastography and b-mode image stream.....70

Figure 4.2 Algorithm of multi-stream GPU elastography and O-TRuE: The multi-stream GPU elastography algorithm is described on the left and the corresponding O-TRuE, which reuses several components of the multi-stream GPU elastography is on the right.....72

Figure 4.3 Elastography stream pipeline: Figure shows contents of the elastography image stream. These are collection of kernel calls in CUDA necessary to generate elastography images [89]. Since these streams support data integrity, they can be plugged into distinct threads.74

Figure 4.4 Elastography Server: This figure shows real-time pipeline where data is acquired through a radio-frequency (RF) server which runs on a US machine. As can be seen, a combination of queue and threading mechanism is implemented to connect all the components efficiently. Queuing mechanism allows the receiver and processing threads to work independently. The processing threads sleep if there is no data available to process and are triggered by data receiving component whenever data is ready. Elastography threads are the multiple threads that are spawned per consecutive or selected pair of RF data received. Every thread can send out the data over the network using IGTLMessages. The n^{th} thread

can collect data from all the other $n-1$ threads to perform aggregate operations as averaging or weighted averaging of selective elastography images.....75

Figure 4.5 Real-time Online tracked Ultrasound Elastography (O-TRuE): Figure shows the real-time online tracked US elastography (O-TRuE) where the cost function is calculated from combinations of the tracked RF data. Then the elastography images are computed for the top m RF data pairs according to the Crr values. The elastography images can then be fused together by simply averaging the images or by weighted averaging based on average correlation values of each elastography image.....76

Figure 4.6 Integration with da Vinci surgical systems: Untracked elastography has been integrated with da Vinci surgical systems using a laparoscopic probe controlled by an arm of the da Vinci surgical robot. (C) Shows the overall setup. (D) Shows the view from surgeon’s console of how B-mode (B) and Elastography image (A) appear when overlaid in the console display.89

Figure 4.7 Timing graph to show speed comparison of multi-stream elastography (threaded) and non-stream elastography (normal): The graphs indicates run times and standard deviation of run time for window size 12, displacement 2 mm, overlap 98% (A, B) and Window size 16, displacement 4 mm, overlap 99% (C, D). The results are per 100 frames. The standard deviation is max 0.13 for Fig. (A), 0.122 for Fig. (B), 0.136 for Fig. (C), 0.167 for Fig. (D), which is very small for 100 frames. This graph also shows that the increased window size reduces the performance of the algorithm due to higher serial search within the large windows.....91

Figure 4.8 Selection map of O-TRuE images: The row above each image sequence indicates the RF data pair index. For e.g. the pair identifier ($n1, m1$) indicates comparison of radio frequency (RF) data frame acquired at time t_{n1} with that of the frame acquired at time t_{m1} . The pair (image rank, Crr value) below the image sequence indicates the rank and Crr value generated by O-TRuE. The pair (CNR, SNR) indicates contrast-to-noise ratio and signal-to-noise ratio values for each image. O-TRuE selected 90% good elastography images in top 20 ranked images with good CNR and SNR above 0.51 and 2.37 respectively. The Crr above 0.457 is observed to provide with good elastography images.....93

Figure 4.9 Elastography image fusion: The images displayed in (a) is elastography image with single image (best O-TRuE) selection, (b) is elastography image for average of top 3 O-TRuE image selections, and

(c) is elastography image for average of top 5 O-TRuE image selections. The results indicates that the fusion by averaging the top 5 elastography images from O-TRuE gives good quality indicated by the average CNR and SNR values of 1.327 and 2.210 respectively.94

Figure 4.10 Elastography image stream analysis of consecutive frames in O-TRuE and Untracked elastography: An analysis of consecutive frames is done to understand the quality of strain images generated by O-TRuE and untracked elastography. (A) Shows a template region selected in the leftmost image and a target region selected in the rightmost image. We apply normalized cross-correlation in these regions as shown in eq. 13 to find max correlation value. A max correlation graph for 100 elastography image pairs is shown in (B), where the red dashed line is for O-TRuE and a blue dotted line is for untracked elastography. O-TRuE has a more consistent high correlation value across consecutive images. As indicated in Table 4.2, O-TRuE (β values) performs better than untracked elastography. (C) Shows the dataset for frames in range [51, 60]; here O-TRuE has its lowest cross-correlation value from 53 to 54; as can be seen, the image quality drastically changes in this range..98

Figure 4.11 Animal Experiment setup: An in-vivo animal experiment was performed on a pig liver; an ablation was induced in the liver using RITA ablator as shown in (C). Elasticity image can be seen in (A), corresponding B-mode image in (B). The ablation region was approximately 2 cm in diameter as validated by gross pathology of the liver in (D).99

Figure 4.12 Trend of untracked elastography for in-vivo pig data: NCC window size vs. CNR and SNR: The graph shows variation of CNR and SNR of individual sample points for different NCC window sizes with untracked elastography. The data was obtained from in-vivo experiments on 350 samples and 199 samples were selected after ignoring invalid strain values. (A) Shows snapshot of CNR values and (B) shows snapshot of SNR values varying for a small subset of the 199 samples. The average/min/max values of the CNR and SNR are listed in Table 4.3. The CNR and SNR across different window sizes are closely related per sample but the global variation in CNR and SNR is high due to wide range of values.100

Figure 4.13 Max cross-correlation graph of consecutive images from robot assisted palpation: Max cross-correlation graph performed on consecutive frames for da Vinci surgical system. (A) Shows a template region selected in the leftmost image and a target region selection in the rightmost image. We apply

normalized cross-correlation in these regions as shown in eq. 13 to find max correlation value. (B) Shows a mean correlation graph of initial 30 elastography image pairs out of 1200 elastography image pairs. Palpation parameters is expressed as frequency (f) in Hz and amplitude λ in mm. High correlation indicates a good match in consecutive frames; clearly, $f = 5$, $\lambda = 3$ indicates a very stable and consistent result with $\beta = 96.58$, where β is defined in eq. 14. Each β value is calculated for 1200 elastography image pairs. 107

Figure 5.1 System diagram of an image-based elastography quality detection system: The figure shows the flow of our method, in which the RF server sends the RF data to the selector, where the elastography is computed for a selected region of interest. If the cumulative weighted average correlation and the cumulative weighted SNR values pass a certain threshold, then the elastography is computed for the full images for the RF image pairs. 116

Figure 5.2 Elastography on selective regions of interest: The RF images are divided into regions of interest, and elastography is computed for the entire RF lines such that the selected regions of interest are calculated. We then calculate the average correlation and SNR values in the regions of interest. Greater weight is given to values in the regions of interest that are near the central areas of the image. 117

Figure 5.3 Comparison of filtered vs. unfiltered elastography based on CNR histograms: For a low average CNR value of 0.1804, nearly ~43% of the unfiltered elastography images fall at this average CNR value and only ~22% of the filtered elastography images correspond to this average value. For values above the good quality CNR value of 2.6982, a total of ~34% of the filtered elastography image fall in that region, and a total of only 7.54% of the freehand elastography data correspond to that region. This finding indicates that the filtered elastography algorithm performs better than unfiltered elastography. 120

Figure 5.4 Filtered vs. unfiltered elastography SNR histogram comparison: For SNR values in the range 0 to 1, a total of ~66% of the unfiltered elastography images and 22.64% of the filtered elastography fall under these value range. Lower SNR values indicate that more low-quality images are acquired with unfiltered elastography. The values change for the range 1 to 1.5, with a total of 75.47% of the filtered elastography images falling under the threshold, and the remaining 33.96% of the freehand

elastography images falling under the threshold. This finding indicates that filtered elastography performs better than unfiltered elastography. 123

Figure 5.5 CNR histogram across different width and height values: We represent width (in pixel) with w and height (in pixel) with h for the regions of interest. For the values between region 0 and 1.22, a total of 53.77% of images with $w = 10$ and $h = 10$ fall in this region, which is higher than the percentage observed for other widths and heights. For the CNR values between 1.5 and 3, the total percentage of samples for $w = 10$ and $h = 10$ is 41.51%. This value is slightly less than observed for other widths and heights. For CNR values above 3, $w = 25$ and $h = 15$ and $w = 20$ and $h = 15$ fare better, with 11.32% and 13.2% of their images, respectively. From the CNR data, we can conclude that $w = 25$ and $h = 15$ has a slight advantage than other values. 126

Figure 5.6 SNR histogram across different width and height values: We again represent width (in pixels) with w and height (in pixels) with h for the regions of interest. For SNR values between 0 and 1, $w = 10$ and $h = 10$ has a total frequency of 33.96% of the elastography images falling under the threshold. This is higher than the values obtained for the other w and h values. For CNR values between 1 and 2, the total frequency values for $w = 25$ $h = 15$ and $w = 20$ $h = 15$ are 76.41% and 77.35% respectively. This finding shows that $w = 25$ $h = 15$ and $w = 20$ $h = 15$ yield better results than the other width and height values. Thus, greater width and height values generate better quality data. 130

Figure 5.7 CNR histogram comparison for different weights: For CNR values in the range of 0 to 1.5, a total of 51.25% of the elastography images corresponds to the weights $\{2\ 4\ 2\}$, which is higher than the values obtained for the other weights. For the range 1.6 to 3, the weights $\{1\ 8\ 1\}$ have a total of 51.25% of the dataset fall within the range, which is higher than the values obtained for the other weights. For CNR values above 3.1, no significant difference in performance was observed. However, overall, the weights corresponding to $\{1\ 8\ 1\}$ perform better than the other weights. 136

Figure 5.8 SNR histogram comparison for different weights: For the SNR range up to 0.95, the weights $\{2,4,2\}$ and $\{1,12,1\}$ have a total of ~50% of their elastography images in this range. The weight $\{1,8,1\}$ has a lower total percentage (36.25%) of its values in this range. For SNR values, above 1, the weights $\{2,4,2\}$ and $\{1,12,1\}$ have totals of 46.25% and 50% of the images, respectively. In contrast,

the weight $\{1,8,1\}$ has a total of 63.75% of its dataset in this range. This result indicates that the weight $\{1,8,1\}$ performs better than the other weights. 137

Figure 5.9 ROI image pair selection output EI images: Figure shows output EI images with varying CNR and SNR values. (A) and (B) are good quality elastography images with both CNR and SNR values above 1. (C) and (D) are poor quality elastography images with CNR values below 0.25 and SNR values below 0.6. 139

Figure 5.10 Visual classification of the filtered elastography data: In this experiment, we visually marked a good quality elastogram with 1 (\times symbol) and a bad quality elastogram with 0 (o symbol). The plot shows CNR on the x-axis and SNR on the y-axis, with each point indicating the classification of the corresponding elastography image as 0 or 1 depending on the visual quality. For filtered elastography a total of 63.2% of the images were found to be good (value 1). That result is 2 times better than the result obtained from freehand elastography. Most of the images with CNR and SNR values above 1 are found to be of good quality. The maximum CNR value is 5.4, and the minimum value is 0.0006. In contrast, the maximum SNR value is 3.4, and the minimum value is 0.5. There are few outliers, but those outliers are near the horizontal and vertical lines at value 1. This figure indicates that the filtered elastography algorithm works better than the freehand algorithm, as shown in Fig. 5.11..... 141

Figure 5.11 Visual classification of the unfiltered (freehand) elastography data: In this experiment, good quality elastogram is marked with 1 (\times symbol) and a bad quality elastogram is marked with 0 (o symbol). The plot shows CNR on the x-axis and SNR on the y-axis. Each elastography image is plotted at its corresponding CNR and SNR values, with 0 or 1 to indicate a good or bad quality image. For freehand elastography, a total of 32% of the samples were found to be good quality (\times symbol). Most of the images with CNR and SNR values above 1 are found to be of good quality. The maximum CNR value is 6.29, and the minimum CNR value is 0.0006. In contrast, the maximum SNR value is 1.35, and the minimum SNR value is 0.37. Similar to Fig. 5.10, there are few outliers, but the outliers are close to the horizontal and vertical lines at value 1. The sample size is $N=106$. The overall results indicate that filtered elastography performs better than freehand elastography..... 142

Figure 6.1 Data flow diagram of 5D US: the RF server collects the 3D data using a wobbler probe that performs a sector scan using a 2D probe in a particular field of view. The 3D RF data are passed to the

elastography image (EI) server, which calculates the 3D USE data and passes them to the scan conversion module in the 5D US system. The RF server also sends the 3D B-mode data directly to the scan conversion module of the 5D US system. The 3D B-mode and the 3D USE scan-converted data are then passed to the visualization system. The users select the transfer function values to highlight different areas of the volume with different colors. 156

Figure 6.2 Data flow diagram of 5D US: a block diagram of the 3D USE system that collects the data. The process is distributed among the elastography image (EI) thread to calculate each slice independently. These slices are then collected in the accumulator thread that waits for the remaining threads to finish their task and then sends these USE data as one volume to the 5D US system. 164

Figure 6.3 Experimental setup: this figure shows the experimental setup for our experiments. (A) Experimental phantom setup where a 4D probe is held by a passive arm on top of CIRS Elasticity 049A QA phantom. (B), (C) Experimental setup to validate scan conversion of 2.2-cm sphere under water. 168

Figure 6.4 Performance of non-threaded versus threaded 3D EI: this figure shows the comparison of time taken to compute a volume of elastography for threaded versus non-threaded version to determine whether our algorithm led to any improvement in performance. As indicated in this graph, a threaded version led to a $4.45 \times$ improvement in speed. This proves that our algorithm has managed to keep the EI volume generation to just below 1 s for 4 cm imaging depth (approximately 0.75 s/volume) and 1.46 s/volume for 8 cm imaging depth. 171

Figure 6.5 Performance of non-threaded 3D EI for different window sizes of NCC: this figure shows comparison between different window sizes of 3D NCC. The forward image search is restricted to maximum 2 mm, and window overlap is kept constant at 98 %. It has been observed that the runtime in general decreases with decreasing window size. The average standard deviation (0.13 s/volume) is lowest for window size 8 and 12 and highest (0.16 s/volume) for window size 10. This indicates that the speed for window size 8 is stable and faster than other window sizes. A difference in standard deviation of 0.03 s/volume is notable because the fastest runtime is 1.45 s/volume. The average sample size is 235 volumes. The maximum average time obtained is 3.337 s/volume for window size 8 and depth of 8 cm indicating slowest speed, and the maximum runtime is 4.45 s/volume. The minimum

average runtime obtained is 1.45 s/volume for window size 8 and depth of 4 cm, while the minimum runtime is 1.347 s/volume.....	173
Figure 6.6 Validation of size after scan conversion: validation of size of an object after scan conversion is performed by imaging a 2.2-cm-diameter sphere inside a water tank. The scan-converted output is shown in the images with an approximate diameter of 2.2 cm in all the three views.	174
Figure 6.7 Impact of changing opacity values: <i>ellipse A</i> indicates the region of high strain value where the lesion is found; <i>ellipse B</i> indicates the region around <i>ellipse A</i> . The opacity value for (A) is set at 3; in (B), it is set at 50; in (C), it is set at 100. The <i>arrow</i> in the output section indicates the corresponding regions highlighted by each <i>ellipse</i> . As shown in (A), the low opacity value reduces unneeded noise and B-mode values while the lesion indicated by <i>arrow A</i> remains visible.....	174
Figure 6.8 B-mode and strain volume fused together: region <i>A</i> on the transfer function map emphasizes the hyper-echoic region as a spherical region in the output. The rest of the B-mode values are in the surrounding region of the lesion.	175
Figure 6.9 Differentiating hard and soft lesion: the soft lesion (elasticity 39 kPa) is highlighted with pink color (<i>ellipse B</i>) and hard lesion (elasticity 58 kPa) is highlighted with blue color (<i>ellipse A</i>). We can differentiate the soft lesion (label <i>B</i>) from the hard lesion (label <i>A</i>) with a subtle difference. There is an overlap of colors where the soft lesion is partially green due to the small difference in elasticity of the lesions.	175
Figure 6.10 Differentiating B-mode, elastography and fused volume: this figure shows volume rendering with different input data types. (A) shows only B-mode volume to display contour of the lesion, (B) shows only elastography volume to show the ablated region, and (C) shows both B-mode and elastography fused volume. In (C) the lesion region is displayed as green region surrounded by background in blue and pink.....	176
Figure 10.1 Raw dataset for visual classification of filtered elastography dataset (1-35): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....	205

Figure 10.2 Raw dataset for visual classification of filtered elastography dataset (36-70): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....205

Figure 10.3 Raw dataset for visual classification of filtered elastography dataset (71-106): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....205

Figure 10.4 Raw dataset for visual classification of freehand elastography dataset (1-35): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....205

Figure 10.5 Raw dataset for visual classification of freehand elastography dataset (36-70): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....205

Figure 10.6 Raw dataset for visual classification of freehand elastography dataset (71-106): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.....205

List of Tables

Table 4.1 Test results for comparing frame rate performance of multi-stream GPU elastography (threaded) with single-stream (streamed) and non-stream (normal) GPU elastography. This table reports average frames per second (with standard deviation in brackets) of images generated by various versions of the elastography program. The term *normal-N* indicates the basic GPU implementation of NCC elastography, *streamed-N* indicates the streamed GPU implementation, and *threaded-N* indicates the multi-streamed GPU implementation, where *N* indicates the number of RF lines in each RF image. Four test cases were performed at different NCC window sizes, NCC maximum search distances (displacements), and NCC search step sizes (specified as percentage of window overlap). The computational load increases with larger window size, displacement, and percent overlap. As seen in the results, the highest speed obtained is 78 frames per second (fps) running the multi-streamed GPU implementation.90

Table 4.3 Percentage of consecutive frame pairs above a certain threshold of max correlation for varying σ values as described in eq. 2, 7, 10 and eq. 14. As an example $\sigma = 1$ (β_1) and buffer size 10 indicates percent of correlation values above the range 0.6 for the graph in Figure 4.10. As can be seen in this table, in most cases, the quality of the output system improves with the increasing buffer size. 102

Table 4.4 Untracked elastography: NCC window size vs. speed and image quality: The table shows the change in frame rate and in CNR and SNR according to NCC window size of the multi-stream elastography. We varied the window size while fixing the maximum NCC search distance at 2 mm and overlap of 98%. The CNR and SNR were averaged for 198 images. The speed was calculated by calculating elastography images for the first 100 RF pairs 20 times. It is found that window size of 10 is optimal with high mean CNR and a good mean SNR value; although the highest mean SNR value corresponds to window size 14. This table indicates that as the window size increases the mean CNR and SNR increase along with a reduction in frame rate. Intermediate frame rates corresponding to window sizes 8 and 10 give satisfactory mean CNR and SNR and a high frame rate of 52.07 and 48.16 respectively. 104

Table 5.1 Filtered vs. unfiltered elastography CNR data: Statistical significance results. These data correspond to the data displayed in Fig. 5.3, with sample size $N = 106$. The data were tested for using both a t-test and ANOVA to determine the statistical significance of the experiment. With a p-value < 0.0001 , the experiment is extremely statistically significant. These data indicate that filtered elastography performs better than unfiltered (freehand) elastography. 121

Table 5.2 Filtered vs. unfiltered elastography SNR data: Statistical significance results. The data presented in this table correspond to the data displayed in Fig. 5.4. The sample size is $N = 106$. The purpose of this t-test and ANOVA is to determine the statistical significance of this experiment to indicate whether the results are repeatable and did not happen by a chance. With p-values for both the t-test and ANOVA < 0.0001 , the experiment is extremely statistically significant. This finding indicates better performance for the filtered elastography algorithm than the unfiltered (freehand) elastography version. 124

Table 5.3 CNR data for different widths and heights of the region of interest: Statistical significance results. The data in this table correspond to the results presented in the Fig. 5.5. The sample size is $N=106$. We performed pairwise t-test comparison between Group 1 (minimum mean value) and the other groups 2, 3, and 4. We found statistical significance for the comparison between Groups 1 and 3 and Groups 1 and 4, with the values of Groups 3 and 4 higher than the mean values. This finding supports our claim that the width and height values that correspond to Group 3 and 4 give better results for the CNR values. The ANOVA result of p-value 0.11483 indicates that the overall experiment is not statistically significant. This outcome can be attributed to the low sample size for a four-group scenario. We expect that the results will improve with more data validation. 128

Table 5.4 SNR data for different widths and heights of the region of interest: Statistical significance results. The table results presented in the table correspond to the result presented in Fig. 5.6. The t-test is statistically significant only for groups 1 and 3. The p-value for groups 1 and 4 is also close to 0.05, which is the conventional threshold used to establish statistical significance. Like the histogram, this test also indicates that $w=20, h=15$ fares better than the other groups and yields performance results close to $w=25, h=15$. The ANOVA test fails to establish statistical significance. This failure can be corrected by including more samples in the experiments. 129

Table 5.5 CNR data with different weights for the region of interest: Statistical significance results. The results presented in this table correspond to the data presented in Fig. 5.7. The t-test and ANOVA (p-value 0.699) fail to establish any statistical significance for the overall results. However, the low p-values for group-pairs 2,3 and 1,2 indicate that the common group 2 performs better than the rest of the groups in terms of the SNR values. The results can be further improved by increasing the sample size, which is N=80 for this experiment. 132

Table 5.6 SNR data with different weights for the region of interest: Statistical significance results. This table indicates the statistical significance test for the results presented in Fig. 5.8. The t-test and ANOVA (p-value 0.4531) again fail to establish any statistical significance for the overall results. These results may improve with more samples, than the N=80 for this experiment. 133

Table 6.1 Throughput of elastography algorithm: this table lists the average throughput of the input RF data volumes for real-time elastography. The values are multiplied by 2 because generating an elastography image volume needs a pair of RF data. As can be seen, the 8 cm imaging depth for threaded 3D EI gives maximum average throughput of 84,643 kB/s. The max throughput obtained is 87273 kB/s. The lowest average throughput is 18103 kB/s for 4 cm depth with the minimum throughput value of 15519 kB/s. The sample size is on an average of approximately 145 volume computations. 172

Table 6.2 Speed of 3D scan conversion for a B-mode volume: the table lists the speed of real-time 3D scan conversion in *volumes per second* (vps). The columns are arranged per number of frames in each volume or *frames per volume* (fpv). The highest average speed is observed for 8 cm imaging depth at 79.40 vps for 31 fpv with max of 81.42 vps. The slowest 13.81 vps is observed for 120 fpv at 4 cm imaging depth with minimum of 13.62 vps. The result is computed on sample of 15 volume set to get time to compute per volume scan conversion in milliseconds, from this we calculated volumes per second since the standard deviation is low. 172

Table 10.1 Unfiltered vs Filtered elastography dataset (Sr. no. 1 – 75) for CNR values and used to generate the results in the section 5.3.2.1.....204

Table 10.2 Unfiltered vs Filtered elastography dataset (Sr. no. 76 – 106) for CNR values and used to generate the results in the section 5.3.2.1.....205

Table 10.3 Unfiltered vs Filtered elastography dataset (Sr. no. 1 – 75) for SNR values and used to generate the results in the section 5.3.2.1.....	205
Table 10.4 Unfiltered vs Filtered elastography dataset (Sr. no. 76 – 106) for SNR values and used to generate the results in the section 5.3.2.1.....	205
Table 10.5 Varying size of region of interest to calculate elastography dataset (Sr. no. 1 – 25) for CNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.6 Varying size of region of interest to calculate elastography dataset (Sr. no. 26 – 50) for CNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.7 Varying size of region of interest to calculate elastography dataset (Sr. no. 51 – 75) for CNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.8 Varying size of region of interest to calculate elastography dataset (Sr. no. 76 – 106) for CNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.9 Varying size of region of interest to calculate elastography dataset (Sr. no. 1 – 25) for SNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.10 Varying size of region of interest to calculate elastography dataset (Sr. no. 26 – 50) for SNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.11 Varying size of region of interest to calculate elastography dataset (Sr. no. 51 – 75) for SNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.12 Varying size of region of interest to calculate elastography dataset (Sr. no. 76 – 106) for SNR values and used to generate the results in the section 5.3.2.2.	205
Table 10.13 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 1 – 50) for CNR values and used to generate the results in the section 5.3.2.3.	205
Table 10.14 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 51 – 80) for CNR values and used to generate the results in the section 5.3.2.3.	205
Table 10.15 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 1 – 50) for CNR values and used to generate the results in the section 5.3.2.3.	205
Table 10.16 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 51 – 80) for CNR values and used to generate the results in the section 5.3.2.3.	205

List of Pseudocodes

Pseudocode 4.1 RFDataReceiver: This module receives the RF data from the RF server or from a file and transmits the data to the EI processing engine via a queuing mechanism. This module also receives the used RF data images to be freed in the memory.	77
Pseudocode 4.2 EIProcessor: This module receives the RF data from the RFDataReceiver and processes the data to compute elastography images and transmit the images over the network. This module is the basic version without the CPU threading model.....	78
Pseudocode 4.3 EIProcessorAdvanced: This module receives the RF data from the RFDataReceiver and processes the data to compute elastography images and transmit the images over the network. This module is the advanced threaded version with the CPU threading model.	79
Pseudocode 4.4 EIProcessorOTruE: This module receives the RF data from RFDataReceiver that has a tracking information embedded from a tracking device and combined by the MusiiCSync program. This module calculates a permutation of all images for a buffer size M and selects the top X images to be sent on the network.	80
Pseudocode 4.5 EIProcessorOTruEAveraging: This module receives the RF data from RFDataReceiver that has tracking information embedded from a tracking device and combined by the MusiiCSync program. This module calculates a permutation of all images for a buffer size M and selects the top X images to calculate a weighted average based on the average correlation of each image and send the data on the network.	81
Pseudocode 4.6 EIComputationAveraging: This module computes the average of the top X ranked elastography images. This module computes the average based on the weighted average of the average correlation values. The last Xth thread computes the average. Other threads simply deposit their results in a common location.....	82
Pseudocode 4.7 CalculateTopXTruE: This module returns the Top X RF data image pairs that are in close proximity to obtain a good quality elastography output.	83

Pseudocode 4.8 EIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography. 84

Pseudocode 6.1 3DRFDataReceiver: This module receives the 3D RF data from the RF server or from a file and transmits it to the EI processing engine via a queueing mechanism. It also receives back the used RF data images to be freed in the memory. 159

Pseudocode 6.2 3DEIProcessor: This module receives the 3DRF data from the 3DRFDataReceiver and processes them to compute elastography images and transmit them over the network. This is the basic version without the CPU threading model. 160

Pseudocode 6.3 3DEIProcessorAdvanced: This module receives the 3D RF data from the 3DRFDataReceiver and processes them to compute elastography images and transmit them over the network. This is the advanced threaded version with the CPU threading model. 161

Pseudocode 6.4 threaded3DEIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography. Note, the threaded version calculates the elastography for individual RF image pairs but the last Mth thread waits and collects rest of the threads to construct a volume and send it over the network. 162

Pseudocode 6.5 nonThreaded3DEIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography. Note, the non-threaded version simply computes elastography for only one pair and the calling function creates the volume. 163

1 Introduction

1.1 Motivation

Ultrasound elastography (USE) is a technique that is used to find a cancerous tumor via tissue displacement estimation [1]. In quasi-static elastography, the displacement motion of the tissue is achieved by externally inducing palpation of the ultrasound probe [1]. Two radio frequency images (one image obtained before compression, and another image obtained post-compression) are usually used to generate a displacement map [1]. The outliers are removed by subjecting the displacement map to averaging and median filters[1]. Finally, the strain map is calculated by linear regression[1].

This thesis addresses the challenge of using elastography successfully in the operation theater to allow surgeons to obtain real-time feedback. Ultrasound elastography is computationally expensive, and conventional CPU-based methods are slow. The CPU utilization that occurs during elastography leaves the ultrasound system with fewer resources for the remainder of the tasks. We have devised a solution that offloads this elastography computation task by using a graphics processing unit (GPU).

Free-hand elastography is challenging, as a trembling hand motion induces out-of-plane motion of the US probe. We select in-plane RF frames with the correct displacement between two images by using external trackers as an optical tracking system, an

electromagnetic tracking system, and in limited scenarios, a robot-actuated stable palpation motion.

The external tracker, as an optical tracker, is obstructed by various objects and staff members several times during surgery, while the EM tracker is affected by ferromagnetic materials present in the operating room. An image-based search method for calculating elastography in a limited ROI is developed to address these challenges. The weighted average is calculated for the SNR values and correlation coefficients of these ROIs. The system uses these averages to test with a user-defined threshold and select images based on the threshold values [2].

The 3D elastography technique provides tracking information for the tumor in the elevation direction of the US probe, while the 3D B-mode provides contour information for the palpated organ. To obtain a better view of both modalities, we combine 3D B-mode and 3D elastography to visualize the organ over time. We call this new modality the 5D US system [3].

The use of ultrasound elastography in stiff areas detected inside the tissue is well established [1]. A stiff area often corresponds to the cancerous region inside the human body [1]. As a result, ultrasound elastography has become a potential non-invasive tool for cancerous tumor detection, with the ability to replace biopsy if used in conjunction with the B-mode ultrasound method [4].

USE has been used to detect cancer in several organs, such as the liver [5][6], breast [1][4][7], prostate [8][9][10], kidney [11][12], and spleen [13]. Because USE can distinguish soft tissue from hard tissues, this technique has also been applied in the

assessment of liver fibrosis caused by Hepatitis infection, which is one of many causes of liver tissue hardening [14]–[18]. Recent advances in acoustic radiation force impulse elastography have been used to detect liver fibrosis [19][20].

Thermal ablation therapy is a popular minimally invasive technique that is used to burn the cancerous tissue in an organ [21]. Thermal ablation methods have become adjunct to the surgical removal procedure for tumors that are more invasive and require a longer patient recovery time [22]. This ablation therapy technique uses a highly targeted system to identify the tumor location during preoperative planning and deliver heat using target needles [21][22]. Several technologies, such as radio frequency ablation, high-energy focused ultrasound (HIFU), microwave technology, and lasers, are used to deliver the ablation [21][22]. Such a system requires continuous monitoring of the ablation region to ensure that healthy tissues are not accidentally ablated and subsequently killed [21][22]. Partial burning of cancerous tumors may result in the recurrence of cancer at a later time[21][22]. As many as 34-55% of cases have recurred after RF ablation treatment [21]. Real-time thermal monitoring of the ablation therapy is possible with this infrastructure of high-speed feedback using elastography on a GPU [21][22].

1.2 Thesis Statement

Real-time ultrasound elastography systems can enable the detection of cancerous tumors during a biopsy or an intraoperative procedure, providing instantaneous feedback to the surgeon. Real-time elastography systems can be achieved by using the many cores of the GPU to compute 2D and 3D elastography, developing robust methods to filter bad

correlation images, and developing a volumetric visualization system. Fused real-time 3D B-mode and elastography data that are updated over time enable a five-dimensional (5D) ultrasound system. The 5D ultrasound system will enable the tracking of depth information for the tumor in both B-mode and elastography.

1.3 Hypothesis

I evaluated the hypothesis that a fast, reliable, and real-time elastography system can accelerate the delivery of feedback concerning suspected cancerous tissues or ablated regions to the surgeon and enable various technologies that are difficult to achieve with traditional CPU-based architectures. This hypothesis is realized by achieving the following specific aims in this thesis:

Aim 1. Develop a graphics processing unit (GPU)-based real-time elastography system to speed-up the existing elastography algorithm and evaluate clinical feasibility with in-vivo animal experiments.

Aim 2. Integrate the GPU-based elastography system with tracked ultrasound elastography using an external tracking system to enable online tracked ultrasound elastography (O-TRUE).

Aim 3. Develop an algorithm that performs elastography on limited regions of interest to determine whether the future elastography generated for the entire image will result in a good elastogram.

Aim 4. Fuse 3D B-mode and 3D elastography and visualize over time to realize the first five-dimensional ultrasound system, that will provide a combined view of both B-mode and elastography modalities that are easily distinguishable via the use of transfer functions.

1.4 Contributions

1. Real-time elastography system pipeline:

We developed an end-to-end pipeline from RF data acquisition to the elastography image visualization system. This system essentially retrieves data from the ultrasound machine and sends it to the network using the OpenIGTLinkMUSiiC library [23], that has a customized data format for the exchange of ultrasound data [2]. The system is modular and allows the elastography component to run on a system equipped with a GPU. An external GPU box can be connected to the existing ultrasound system via a PCI Express slot [2]. The ability to add extra GPUs using an external PCI Express slot helps in the deployment of this system for existing ultrasound systems worldwide [2].

2. Integration with a tracking system to maximize the chances of finding correlated image pairs:

The problem with the freehand ultrasound system is that out-of-plane motion and decorrelation caused by untrained surgeon hands or trembling hands lead to noisy elastography images [2]. Tracked ultrasound elastography [24] helps in selecting in-plane elastography images. We developed a system that collects data from the US machine, collects tracking information from an external tracker connected to the US

machine, combines the US data and the tracking data using MUSiiC Sync, selects in-plane images from a pool of tracked RF images and then calculates elastography in real-time [2]. We developed this innovative pipeline, that we call online tracked ultrasound elastography (O-TRUE) [2].

3. Integration with the da Vinci surgical systems:

Advanced surgical robotic systems, such as da Vinci surgical systems, perform minimally invasive surgery in which the arms of the robots enter the human body without the need to cut open the body [2]. Several advanced robotic surgery systems lack the haptic feedback required to understand the stiffness distribution while navigating [2]. USE can give the surgeon feedback about the underlying tissue stiffness that is hard to see via camera [2]. In this phantom-based study, we integrated USE with a da Vinci surgical robot to generate a stable palpation motion [2]. We evaluated different palpation frequencies and amplitudes to obtain a stable elastography stream quality [2].

4. Image-based selection of correlated image pairs by performing elastography on a limited area of interest in RF images:

In this research project, we perform elastography on a limited area of interest. We then evaluate SNR and correlation values in those areas of interest to determine whether the entire elastography image for the given RF image pair will be of good quality (i.e., with high CNR and SNR values). This method eliminates the need for an external tracking system to select high-quality RF image pairs. We have developed this system as a real-time system in which the GPU is used to perform the computation in real-time for these

areas of interest. The image-based search is a cost-effective method and requires an inexpensive GPU for rapid computation.

5. Five-dimensional ultrasound system:

We have combined 3D B-mode with 3D strain images to obtain a 4D ultrasound and visualized over time to obtain the fifth dimension [3]. The five-dimensional ultrasound system is an end-to-end pipeline for acquiring, processing and visualizing [3]. The prime problem with this system is distinguishing between the strain and B-mode data [3]. For this reason, we have integrated and built the system using a visualizer [25], that helps to distinguish the two modalities with a color code and opacity functions [3].

1.5 Outline

The thesis is divided into six primary chapters in addition to the conclusions, the bibliography, and the appendix chapters.

Chapter 1 is the introduction to the thesis and lays out the foundation of the thesis. We explain the motivation, thesis statements, and contributions.

Chapter 2 details the physics of ultrasound to provide readers with the necessary background to understand various terminologies in the field of ultrasound research. This chapter explains the concept of elastography that is the basis of this thesis.

Chapter 3 lists various 2D and 3D motion estimation algorithms that provide information about the state of the art methods available for motion estimation in ultrasound. This

chapter also details the normalized cross correlation method, which is a type of motion estimation methods that is used extensively in this thesis.

In Chapter 4, we detail the multi-stream GPU-based elastography algorithm that is used to accelerate the GPU based elastography algorithm. The algorithm involves connecting different stages of the elastography computing stages on a GPU using CUDA streams. These streams are then executed on different threads that lead to the asynchronous multi-threaded execution of an elastography computation pipeline. We developed a modular framework that allows us to distribute the processes on a single and multiple compute node. This approach allows us to easily integrate new modules into the system. We integrated our system with an external tracking system, such as an electromagnetic tracker and compared the results with free-hand elastography. We also present a study of integration with da Vinci surgical systems, in which palpation motion is induced externally by the robotic arm. In this chapter, I contributed to the development (design and implementation) of a multi-stream GPU-based elastography algorithm and online tracked ultrasound elastography (O-TRuE). I devised a method to measure the performance of a continuous stream of elastography for evaluating the quality of an online algorithm. Further, I worked in collaboration with Seth Billings and Hyun Jae Kang on the development of integration with da Vinci surgical systems. I am the lead analyzer of the data and the writer of the journal article that was published as [2].

In Chapter 5, we present a tracker-less elastography system to detect in-plane motion for two RF data frames by performing elastography on the limited regions of interest between the two RF image pairs. This system aims to eliminate the use of an external tracking

system to correct hand motion. We define fixed regions of interest (ROIs) and compute elastography only on the RF lines that encapsulate those ROIs. We then give different weights to each ROI and use those weights to compute the weighted average of the signal-to-noise (SNR) ratio and the cross-correlation value. We compare these weighted averages to a threshold and compute elastography only for the RF image pairs that meet these threshold values. I developed this method (design and implementation), analyzed the data, and wrote the chapter.

In Chapter 6, which is the final chapter, and we present a five-dimensional (5D) ultrasound (US) system that combines 3D strain and 3D B-mode data and visualizes over time to create a 5D US system. Surgeons may find it difficult to visualize and correlate two different views consisting of 3D strain and 3D B-mode elastography. Additionally, there is an inherent research problem with the generation and visualization of elastography at the speed of 3D B-mode. We solved this problem by generating the 3D strain images in real-time using a multi-stream GPU. We scan-converted both volumes so that the final volume approximately matches the sector shape of the volume caused by the motion of the 3D wobbler probe. A specially developed visualizer, which was published previously in [25], distinguishes the two volumes using the transfer functions and was improvised for integration with the five-dimensional ultrasound system. The system is an end-to-end system from data acquisition to visualization. In this chapter, I implemented (design and development) the 5D ultrasound system and the 3D scan conversion code and 3D elastography algorithm. I integrated and made substantial changes to a visualizer from a previously published work [25] to generate the visualizer for the 5D ultrasound system. I

am the lead analyzer of the data and also the lead writer for the published version of this chapter [3].

2 Physics of Ultrasound

2.1 Introduction

Sound waves are a mechanical or pressure waves that travel in a medium due to its elastic property [26]. Waves are of two types: longitudinal waves and transverse waves [26]. Longitudinal waves are the waves that cause particles in the medium to move (oscillate) parallel to the direction of propagation [26]. Sound is in the longitudinal category [26]. When a sound wave hits a tissue, pressure is created at the boundary of the tissue [26]. This pressure causes the tissue molecules to oscillate about their original positions and transfer this energy to the neighboring molecules [26]. This transfer propagates inside the tissue in the direction perpendicular to the source of sound generation [26]. This oscillating motion of the molecules can be measured in terms of recurring cycles [26]. When the medium molecules vibrate at a repeated interval, we call the number of repetitions per second or the number of cycles per second as the frequency, that is denoted in Hz (Hertz) [26]. A human ear can typically hear sound waves between 20 and 20,000 Hz (or cycles/sec) [26]. This sound wave is called an audible sound wave or simply sound [26]. Ultrasound is a sound wave with a frequency greater than 20 kHz, and infrasound is a sound wave with a frequency less than 20 Hz [26]. For clarity sound and ultrasound are used interchangeably in this thesis [26].

The wavelength is the length of one complete cycle [26]. One cycle is a representation of the amplitude change that occurs during regular repetitions [26]. In the case of ultrasound,

the amplitude represents a plot of density versus time and represents the variation in density [26]. The wavelength can be viewed as the variation in two density zones defined by two amplitude peaks [26]. The amplitude can be considered to be a representation of the voltage applied to piezoelectric transducer elements [26]. The wavelength can be specified in meters (m), centimeters (cm), or millimeters (mm) [26]. When the amplitude is plotted on a time scale, the period is the difference in time between two successive rarefactions or compressions [26]. The period τ is specified in second (s) [26]. Acoustic velocity (unit: m/s), that is different from particle velocity, is the speed at which a sound wave propagates in the medium and is denoted by c [26].

Density is defined as the tissue mass per unit volume and directly affects the velocity of ultrasound [26]. An increase in density produces more resistance to the propagation of sound waves; thus, the speed of sound waves is inversely proportional to the density ρ , and the relationship [26] is given by:

$$c \propto \frac{1}{\sqrt{\rho}} \quad (1)$$

Compressibility (K) is the property of the tissue that compresses or reduces the volume of the tissue upon the application of external force [26]. Compressibility is inversely proportional to the speed of sound [26]. The bulk modulus β that determines the elasticity of the tissue is also inversely proportional to the compressibility [26]. The relationship between the speed of sound, compressibility, and the bulk modulus [26] is given by:

$$c \propto \sqrt{\beta} \propto \frac{1}{\sqrt{K}} \quad (2)$$

Acoustic velocity can be derived by combining the relationships of the speed of sound with compressibility and density [26]. The revised acoustic velocity is given by

$$c = \frac{1}{\sqrt{K\rho}} = \sqrt{\frac{\beta}{\rho}} \quad (3)$$

Dense materials have lower compressibility, that in turn increases the speed of sound in the denser medium [26].

The relationship between velocity (c), frequency f and wavelength λ [26] is given by:

$$c = f\lambda \quad (4)$$

2.2 Wave Propagation

2.2.1 Acoustic Impedance

The acoustic impedance of the material helps in identifying the difference between the tissues along the path of wave propagation [27]. The acoustic impedance is given by

$$Z = \rho c \quad (5)$$

where ρ is the density of the tissue in kg/m^3 , and c is the speed of sound in the tissue in m/sec [27]. The unit for Z is $\text{k}/(\text{m}^2\text{sec})$ or rayls [27]. The acoustic impedance indicates the stiffness and flexibility of the medium, such as a tissue or a phantom in our case [27]. A large difference in stiffness, as in a bone surrounded by tissue, leads to nearly complete reflection of the acoustic waves originating from the tissue [27]. In contrast, when the

impedance is small, as in the case of two adjacent organs, the reflection is small [27]. Small reflections reduce the difference at a boundary, but a subtle difference may be observed. Similarly, reflection and refraction are dependent on acoustic impedance [27].

2.2.2 Acoustic Energy and Intensity

Acoustic energy contains two parts, primarily kinetic energy when it is in motion and potential energy when it is about to be emitted from the transducer elements [28]. The kinetic energy density w_k of the acoustic wave's kinetic energy [28] is given by

$$w_k = \frac{1}{2} \rho v^2 \quad (6)$$

where v is the velocity of the oscillating particles in the medium [28]. The potential energy density w_p of the acoustic wave's potential energy [28] is given by

$$w_p = \frac{1}{2} \kappa p^2 \quad (7)$$

where κ is the compressibility of the underlying tissue and p is the acoustic pressure [28]. The total acoustic energy density that covers the energy change at a given point [28] is given by

$$w = w_k + w_p \quad (8)$$

We define the acoustic energy flux I [28] by

$$I = pv \quad (9)$$

The acoustic energy flux provides a representation of the energy associated with the wave [28]. The acoustic pressure p [28] is given by

$$p = Zv \quad (10)$$

Substituting the above value for acoustic pressure p into the acoustic energy flux equation [28] gives

$$I = \frac{p^2}{Z} \quad (11)$$

The following equation gives the relationship between acoustic energy flux and acoustic energy density, that describes the acoustic wave power and energy [28]:

$$\frac{\partial I}{\partial x} + \frac{\partial w}{\partial t} = 0 \quad (12)$$

2.2.3 Reflection and Refraction at Plane Interfaces

2.2.3.1 Reflection

An acoustic wave traveling in the tissue gets reflected due to acoustic impedance mismatch, and this reflected wave is detected by the ultrasound transducer [27]. This reflected wave helps to locate the boundary of the organ, and a complete B-mode picture reveals the contour of the organ [27]. The reflection pressure amplitude coefficient is given by:

$$R_p = \frac{P_r}{P_i} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (13)$$

where P_r is the reflected pressure and P_i is the incident pressure [27]. The subscripts 1 and 2 helps to distinguish between media. Medium 2 is proximal, and medium 1 is distal to the transducer [27]. The intensity reflection coefficient is given by

$$R_I = \frac{I_r}{I_i} = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \quad (14)$$

where I_r is the reflected intensity and I_i is the incident intensity [27]. The intensity transmission coefficient, that is the fraction of the incident intensity that is transmitted, is given by the law of conservation of energy as:

$$T_I = 1 - R_I \quad (15)$$

An acoustic coupling gel helps to fill the gap between the surface of the tissue and the transducer [27]. The reason for using this gel is that the acoustic waves are reflected almost 100% back to the transducer due to the high acoustic impedance mismatch between air and the transducer element surface [27]. It is very difficult to image organs that contain air cavities, such as lungs, due to this impedance mismatch [27]. An acoustic window is generated when interconnected tissues inside the lungs transmit the ultrasound [27].

For a perpendicular incidence beam and a smooth surface, the beam is reflected back to the transducer with a 180-degree shift [27]. This is true with the assumption that the wavelength of the ultrasound beam is greater than the surface variation of the boundary [27]. This total reflection is possible due to the high impedance mismatch between the two tissues, with the beam transmitting from the tissue with low impedance to the tissue with high impedance and the impedance mismatch very high [27]. For an ultrasound beam with

a high frequency, the inversely proportional wavelength is small and may be smaller than the surface variation [27]. In this case, the incident ultrasound beam will diffuse in different directions, and not all of the beam is reflected back to the transducer [27]. An ultrasound beam that is not perpendicular to the tissue boundary is incident with an angle θ_i , that is equivalent to the reflected angle θ_r [27]. This nonperpendicular incidence also leads to a loss of signal returning to the transducer [27].

2.2.3.2 Refraction

When the ultrasound beam is not perpendicular to the boundary between two tissues, the transmitted beam changes direction [27]. This attribute is related to the fact that while crossing the tissue boundary the frequency remains constant and the speed of sound in the two media changes [27]. This difference in the speed of sound in two media coupled with the incidence angle helps to calculate the transmitted angle using Snell's law [27], as follows:

$$\frac{\sin \theta_t}{\sin \theta_i} = \frac{c_2}{c_1} \quad (16)$$

Where θ_t is the angle of transmission in medium 2 with speed of sound c_2 and θ_i is the angle of incidence in medium 1 with speed of sound c_1 [27].

For a small incidence angle, the transmission angle can be approximated [27] by:

$$\frac{\theta_t}{\theta_i} \cong \frac{c_2}{c_1} \quad (17)$$

As can be seen from the above equation, when the speed of sound in the transmission medium c_2 is greater than the speed of sound in the incidence medium c_1 then $\theta_t > \theta_i$, and vice-versa for $c_2 < c_1$ [27].

No refraction occurs when the speed of sound in the two media is identical, on a perpendicular incidence where a straight trajectory occurs [27]. An ultrasound machine usually assumes straight line propagation but in the presence of refraction, artifacts occur [27]. A total reflection occurs when $c_2 > c_1$ and the angle of incidence is above a certain angle, that is called the critical angle [27] and is given by:

$$\sin \theta_c = \frac{c_1}{c_2} \quad (18)$$

At this critical angle, the reflected wave travels along the boundary and does not penetrate the second medium [27]. Thus, the critical angle is calculated by setting $\theta_t = 90^\circ$ in Snell's law [27].

2.2.3.3 Attenuation

As the ultrasound wave travels through the tissue, the loss of acoustic energy is termed the ultrasound attenuation and is caused by scattering, reflection, refraction and tissue absorption [27]. The absorbed energy is converted to heat [27]. This loss leads to a loss in the intensity measurement of the ultrasound image [27]. As the depth increases, more acoustic energy gets absorbed, resulting in a noisy image at greater depths [27]. The relative intensity loss per centimeter of travel is called the attenuation coefficient and is denoted by μ [27]. A higher frequency means a lower amplitude and smaller penetration

depths, so the attenuation is higher for higher transmission frequencies [27]. For a 2X MHz transmission frequency, the attenuation coefficient is approximately twice that of an acoustic beam with a 1X MHz transmit frequency [27]. An attenuation coefficient expressed in dB increases logarithmically, so the attenuation increases exponentially with distance [27]. The half value thickness (HVT) is defined as the distance traveled by the sound wave until it is 50% attenuated. The HVT is equivalent to a 3dB drop in intensity and a 6 dB drop in pressure amplitude [27]. Naturally, as the frequency increases, the beam becomes attenuated at a shorter distance and the HVT decreases [27]. Due to the attenuation factor, an ultrasound beam with a lower frequency is used to achieve a greater penetration depth [27].

2.2.3.4 Scattering

The following factors affect the echo amplitude of the reflected signal: (a) the scatterers per unit volume, (b) the acoustic impedance mismatch at the scatterer boundaries, (c) the size of the scatterers, and (d) the transducer frequency [27]. An area is termed hyperechoic when the scatterer amplitude is higher, in contrast to a hypoechoic area [27]. In a hypoechoic area, the scatterer amplitude is lower [27]. This parameter of hyper- and hypoechoic describes the scatter characteristics for the signals originating from the background area [27]. Specular reflections are from a smooth surface; hence, such reflections are independent of the transmission frequency, whereas non-specular reflection is dependent on the frequency of the transducer [27]. A higher ultrasound frequency is helpful in obtaining a better-scattered echo signal [27].

2.2.3.5 Nonlinear Wave Propagation

The wave equation in ultrasound assumes the linearity of the system, where the ultrasound waves traveling through the medium are reflected back at the same frequencies [28]. However, non-linearity is an inherent property of the underlying tissue, especially due to the presence of different organs, foreign bodies, contrast agents, and several other factors [28]. The wave propagating in the medium may hit a surface or material that has a higher acoustic pressure, that will reflect the wave at a faster rate [28]. Thus, nonlinear or harmonic imaging involves sending acoustic waves at identical or different frequencies and receiving those waves at different frequencies [28].

The speed of sound in a nonlinear environment [28] is given by

$$c = c_0 + \beta \frac{p}{Z_0} \quad (19)$$

where p is the acoustic impedance at a given location and time, β is called the coefficient of nonlinearity and is dependent on the acoustic medium, c_0 is the assumed speed of sound for the medium, and Z_0 is the assumed acoustic impedance for the medium [28].

Nonlinear wave propagation can be useful in microbubble imaging with contrast agents, in ablation procedures to monitor various sources of ablation, and to improve the resolution of the underlying ultrasound image [28].

2.3 Ultrasound Instrumentation

2.3.1 Ultrasound Transducer

Ultrasound is produced by a transducer, that can be a linear, phased, or curvilinear array [27]. The transducer typically contains ceramic elements that have electromagnetic properties [27]. These elements convert electrical energy to mechanical energy and mechanical energy into electric energy [27]. A transducer has several components. The primary components are piezoelectric materials, resonance transducers, a damping block, an acoustic absorber, a matching layer, and insulating covers [27].

2.3.2 Piezoelectric Materials

The functional components of the transducer are the elements that contain piezoelectric materials [27]. The piezoelectric material converts electrical energy into mechanical energy to create a local deformation that generates ultrasound waves [27]. These materials also have the reverse property of converting incident mechanical energy that is transferred by an ultrasound wave into electrical energy [27]. These materials have electrical dipoles on their surfaces, that give a positive and negative charge to the material [27]. These charges are equal in quantity to nullify each other and maintain a zero net charge [27]. When a mechanical pressure is applied to this material, a deformation occurs on the surface of the material [27]. Once there is an imbalance between the positive and negative charges, an electrical charge is created which is directly proportional to the amplitude of the incident

waves [27]. Similarly, a reverse effect occurs when the electric pulse induces a mechanical motion in the transducer element, followed by the generation of ultrasound waves [27].

A natural piezoelectric crystal, such as quartz crystal, is used in the common watch to provide mechanical vibrations at a frequency of 32.768 kHz to measure time [27]. The most common piezoelectric material used in ultrasound is lead-zirconate-titanate (PZT), that is a synthetic piezoelectric ceramic [27]. These materials acquire a permanent dipole orientation after a series of procedures that involve molecular synthesis, heating, the application of external voltage, and cooling [27]. The ceramic exhibits piezoelectric properties after heating to a “Curie temperature,” and followed by the application of an external voltage and cooling [27]. After cooling, the ceramic maintains its piezoelectric properties [27].

2.3.3 Resonance Transducers

In a resonance transducer, the piezoelectric material of the transducer vibrates at the natural resonance frequency of the material [27]. This effect can be achieved by applying a small voltage (e.g., 150 V) for a very short duration of time (typically one μsec), so that the piezoelectric ceramic starts vibrating at the natural resonance frequency after an initial contraction [27]. The thickness of the piezoelectric material should be half of the desired wavelength of the ultrasound wave [27]. We can achieve higher frequencies by using a thinner transducer element and lower frequencies by using a thicker transducer element [27]. This effect can be derived from the following equation to calculate frequency [27]:

$$f = \frac{c}{\lambda} \quad (20)$$

where f is the resonant frequency in MHz, c is the speed of sound in the piezoelectric material in m/sec, and λ is the wavelength in meters [27]. These transducers transmit and receive at the center frequency [27].

2.3.4 Damping Block

The back of the transducer must include a wave absorbing material to prevent the occurrence of an echo between the back surface and the transducer, that may create interference with the receiver system [27]. This damping block is layered at the back of the piezoelectric material to attenuate any stray ultrasound waves [27]. This dampening material also impacts the resonance frequency of the transducer and the frequency spectrum [27]. A broadband frequency spectrum or range of frequencies is observed around the center frequency [27]. The bandwidth of the sound emanating from a transducer is described by the Q factor [27], as follows:

$$Q = \frac{f_0}{\text{Bandwidth}} \quad (21)$$

where f_0 is the center frequency of the transducer element, and the bandwidth is the width of the frequency spectrum of the sound wave emanating from the transducer [27].

Because Q is inversely proportional to the bandwidth, a high Q transducer corresponds to a narrow bandwidth and vice versa [27]. To obtain a high spatial resolution along the axial direction, the transducer needs a low Q level or a higher bandwidth. In applications such

as Doppler imaging, the echo frequency must be preserved about the incident frequency due to the motion of the subject and requires a relatively narrow bandwidth transducer [27]. A high Q factor is needed for continuous-wave ultrasound transducers [27]. Please note that the term Q factor is derived from Quality factor, and a lower value does not indicate a reduced image quality [27].

2.3.5 Matching Layer

To minimize the occurrence of sudden acoustic impedance differences between the transducer element and the patient's skin, a matching layer is attached to the transducer element. This matching layer has approximately half of the impedance difference between the two elements [27]. The thickness of this layer is calculated using the center frequency and the speed of sound in the matching layer, that gives the wavelength in the matching layer [27]. The width of the matching layer is one-fourth of the wavelength of sound waves in the matching layer [27]. Additionally, to reduce air pocket between the matching layer and the patient's skin, an acoustic gel is used [27].

2.3.6 Non-resonance (Broad-Bandwidth) “Multi-frequency” Transducers

A multi-frequency or multi-Hertz transmitting and receiving transducer can be generated by changing the design of the transducer elements [27]. The design is changed to allow the PZT elements to be machined into a large quantity of small rods on a surface, and the surface is made smooth by filling with epoxy resin [27]. This material allows for an

acoustic impedance that is closer to the tissue impedance and thus helps in avoiding or reducing the need for matching layers [27]. These multi-frequency transducers have larger bandwidths than resonance transducers [27].

Unlike the resonance transducer, a multi-frequency transducer is excited with a square wave of approximately 150 V with 1-3 cycles [27]. The use of this larger square wave instead of a spike helps in selecting a range of frequencies as a center frequency within the bandwidth of the transducer frequencies [27]. This higher response time also enables the reception of multiple frequencies or a different frequency than the transmitted frequency [27]. The ability to send the wave pulse with one frequency and receive on another (higher or multiple) frequency enables the use of harmonic imaging [27]. Harmonic imaging provides the ability to “listen” to a different frequency when the patient is injected with contrast agents [27]. The interaction of contrast agents and tissues gives rise to higher frequencies [27]. Detecting these frequencies has helped in the use of contrast agents to improve the contrast of the ultrasound images [27]. Harmonic imaging allows a better SNR ratio for increased depth, noise removal and artifact (clutter) removal in the images [27].

2.4 Transducer Arrays

Ultrasound transducers typically have 128 to 512 piezoelectric elements that are rectangular in shape [27]. The width of these elements is less than half of the wavelength of the transducer sound waves, and the length of these elements is several millimeters [27]. Two main activation modes can be used: the linear and phased modes [27]. The arrays are called linear and phased arrays [27].

2.4.1 Linear Arrays

In linear arrays, a group of approximately 20 elements is fired simultaneously, generating a synthetic aperture equal to the number of elements used at a given time [27]. The echoes are detected by these 20 elements and after a short duration of time, the next batch of elements are fired with a 1- to 2-element shift [27]. This firing-sensing-shifting cycle continues until the end of the array [27]. A linear array typically contains 256 to 512 elements [27]. Such a rectangular array assembly creates a rectangular field of view, while a curvilinear array creates a trapezoidal field of view [27].

2.4.2 Phased Arrays

In a phased array, the elements are fired with a fixed interval between them, generating a steered or focused wave-front [27]. This wavefront is detected by the same elements that are activated at a similar fixed interval [27]. Several algorithms then reconstruct the images that originate from the returned echoes [27].

2.5 Ultrasound Probes

Several different types of ultrasound probes are available for different applications [28]. A **single-element probe** has one lens or a curved crystal to help focus on a specific target [28]. Beam steering must be performed to create a sector scan of the underlying body [28]. Real-time imaging is hard to achieve with these systems [28]. A **mechanical scanner** contains a rotating or moving transducer that effectively ensures that the elements on the

transducer cover different areas of the body [28]. This system effectively ensures a planar scan with a little error other than mechanical wheel misalignment [28]. An **electronic scanner** contains an array of elements that electronically swipe through element by element, sending and receiving pulses in the medium [28].

2.6 Pulse-Echo Imaging

2.6.1 The Pulse-Echo Equation

The received signal waveform can be represented by the following simplified equation [28]:

$$r(t) = K \frac{e^{-\mu_a ct}}{(ct)^2} \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty R(x, y, z) n(t - 2c^{-1}z) \tilde{q}^2(x, y, z) dx dy dz \quad (22)$$

where $R(x, y, z)$ is the energy of the spherical wave scatterer, n is the input narrowband pulse, μ_a is the attenuation of the ultrasound signal in the underlying tissue, and K is the arbitrary gain factor [28]. In addition,

$$\tilde{q}(x, y, z) = zq(x, y, z) \quad (23)$$

where the transducer field pattern is given by $q(x, y, z)$ [28].

To cancel or compensate the default gain outside the integral, we multiply with the gain from the time-gain compensation system with which many ultrasound system circuits are equipped [28]. One example of such a gain [28] is given by

$$g(t) = \frac{(ct)^2 e^{\mu_a ct}}{K} \quad (24)$$

The gain compensated input signal [28] is given by

$$r_c(t) = g(t)r(t) = \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty R(x, y, z)n(t - 2c^{-1}z)\tilde{q}^2(x, y, z)dx dy dz \quad (25)$$

The simplified A-mode signal obtained by a simple AM modulation scheme, which takes the absolute value followed by a low pass filter, of $r_c(t)$ is represented by the following equation [28]:

$$e_c(t) = \left| \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty R(x, y, z)n_e(t - 2c^{-1}z)e^{j2kz}\tilde{q}^2(x, y, z)dx dy dz \right| \quad (26)$$

where n_e is the envelope of the input signal obtained by the complex modulus of the narrowband input pulse [28]. In the given system n_e is derived from the following equation [28]:

$$n(t) = n_e(t)e^{j\phi}e^{-j2\pi f_0 t} \quad (27)$$

where f_0 is the central frequency of the transducer, and ϕ is the phase angle [28].

2.6.2 Transducer Motion

This section gives imaging equations that are needed when the transducer is allowed to move in the x-y plane [28]. An A-mode signal for a transducer moving in the $z = 0$ plane (i.e., at position $(x_0, y_0, 0)$) [28] is given by:

$$e_c(t; x_0, y_0) = \left| \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty R(x, y, z) n_e(t - 2c^{-1}z) e^{j2kz} \tilde{q}^2(x - x_0, y - y_0, z) dx dy dz \right| \quad (28)$$

A signal originating from an echo location at time t must originate from the following z_0 position [28]:

$$z_0 = ct/2 \quad (29)$$

We express e_c as a reflective function \hat{R} to express in x_0 , y_0 , and z_0 instead of x_0 , y_0 , and t as [28]:

$$\hat{R}(x, y, z) = e_c(2z_0 / c; x_0, y_0) \quad (30)$$

This substitution yields the following equation [28]:

$$\hat{R}(x_0, y_0, z_0) = \left| \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty R(x, y, z) e^{j2kz} n_e(2(z_0 - z) / c) \tilde{q}^2(x - x_0, y - y_0, z) dx dy dz \right| \quad (31)$$

2.6.2.1 The Geometric Assumption

If we assume that the energy propagated in the medium travels down a cylinder with the same shape as the transducer then we make the following assumption [28]:

$$\tilde{q}(x, y, z) = s(x, y) \quad (32)$$

This assumption simplifies equation (32) to give:

$$\hat{R}(x, y, z) = K \left| R(x, y, z) e^{2kz} * \tilde{s}(x, y) n_e\left(\frac{z}{c/2}\right) \right| \quad (33)$$

where $\bar{s}(x, y) = s(-x, -y)$ is the reflective term of $s(x, y)$ and $*$ is the convolution reflective coefficient and blurring function [28].

We obtain a resolution cell from the equation after ignoring the term e^{2kz} [28]. This term e^{2kz} causes a speckle in the resolution cell [28]. The phase term e^{2kz} , that has multiple periodic cycles within the resolution cell multiplies the reflective distribution [28]. This phase then generates a Rayleigh random variable after taking the modulus of integration over an entire resolution, generating a large number of complex numbers $R(x, y, z)e^{2kz}$ with a complete random phase for each resolution cell [28]. This Rayleigh random variable is dependent on the underlying reflective distribution hence it produces a spatially matched random pattern [28]. This pattern is called a speckle pattern [28].

2.6.2.2 Fresnel and Fraunhofer Approximation

Applying the Fresnel and Fraunhofer approximation to \tilde{q} simplifies \hat{R} and gives the following expression [28]:

$$\hat{R}(x, y, z) = \left| R(x, y, z) e^{j2kz} * S\left(\frac{x}{\lambda z}, \frac{y}{\lambda z}\right)^2 n_e\left(\frac{z}{c/2}\right) \right| \quad (34)$$

where S is the approximation of \tilde{q} which is similar to the sinc function [28]. The form of $S(\cdot, \cdot)$ gives rise to sidelobes that cause artifacts in the resulting image [28]. The speckle term e^{j2kz} is still present, which indicates that we cannot remove the speckles with this method [28]. The resolution cell size increases with depth and usually blurs the local object at shallower depths [28].

2.7 Ultrasound Imaging Modes

2.7.1 A-mode ultrasound scan

Various interfaces in human body reflects the ultrasound energy burst that is sent by the ultrasound transducer at regular short intervals [26]. This reflected energy is detected by ultrasound transducer crystal to “induce” radio frequency (RF) signal [26]. This signal is amplified and plotted on the display as a short spike [26]. The position of the spike is determined by the time delay taken by the ultrasound wave to reflect back [26]. The A-mode or Amplitude-mode is simply the amplitude of the reflected ultrasound energy that represents the reflectivity of the different interfaces from which the wave is reflected [26]. The A-mode ultrasound imaging mode helps in scanning an organ along the axial direction orthogonal to the transducer plane. A-mode ultrasound has been helpful in diagnosing maxillary sinusitis [29], measuring the distribution of cartilage thickness in the knee joint in elderly individuals [30], and performing biometric studies of the canine eye to measure the axial length, tissue velocity, and differences related to gender [31].

2.7.2 B-mode ultrasound scan

B-mode ultrasound scanning is probably the most widely used imaging mode in ultrasound. A B-mode scan is simply a combination of A-mode signals in space such that amplitude is reflected by different brightness, with each A-line pointing to the location of transducers at a different location. A change in the thickness of arteries observed from a B-mode scan

indicates the progression of atherosclerotic lesions in humans [32]. B-mode ultrasound can also be used to detect, deep-vein thrombosis [33], measure the skin thickness for progressive systemic sclerosis [34], and detect cancerous breast tumors [35].

2.7.3 M-mode ultrasound scan

The M-mode ultrasound is simply the plot of motion of B-mode dots displayed over time [26]. The data is collected on a single line of sign with the different points getting constantly updated at a frequency of 1 kHz [26]. This makes M-mode ideal for tracking the motion of beating heart and its aorta valve movement to detect any irregularities [26]. The M-mode ultrasound imaging mode has been used to measure the thickness of the transversus abdominal in different body positions [36], to perform fetal echocardiography [37], [38] to measure the diameters of the right and left ventricular regions, and to measure the arterial stiffness to detect the early development of major vascular disease [39]. M-mode signals are frequently displayed alongside and compared with electromyography (EMG) [40] and echocardiography (ECG) [37][41] to determine the presence of any diseases.

2.8 Steering and Focusing

A sector scan is generated by the phased array transducers with the center of the transducer array appearing as the source of the origin of the waves [42]. Similar to other scanning methods, each line indicates a transmit-receive path of the sound waves [42].

The physical structure of the phased transducer array appears to be similar to that of a linear transducer array [42]. The phased array contains approximately 128 rectangular transducer elements [42]. The transducer is smaller than the linear transducer array, with a total aperture of approximately 30λ for a square side. λ is the wavelength of the sound wave in the PZT material [42]. The size of the individual elements is typically $\lambda/2$ width [42]. Each element of the phased array system is used as a transmitter and receiver for a scan line [42]. However, in the linear array, there is a different set of active elements that receives the ultrasound beam echoes [42]. Due to the identical thickness of the element in linear and phased arrays, the elevation resolution remains constant for both types of arrays [42].

To achieve a good signal to noise ratio, the phased array transducer should be angled on the surface of the tissue in such a way that the region of interest is in the center of the field of view [42]. This technique also ensures the best lateral resolution, the highest sensitivity, and adequate contrast resolution [42]. With an increasing steering angle, the width of the beam increases, reducing the lateral resolution at the sides of the sector [42]. The sensitivity decreases with an increasing steering angle, as the individual elements are most efficient close to the direction perpendicular to the transducer array [42].

2.8.1 Transmit Steering and Focusing

To change the FOV of the sweep angle across the medium, a time delay is added sequentially to each transducer element [27]. It is preferable to obtain a wave with a very large far field for divergence and a short near field [42]. This effect can be achieved with the use of very narrow elements [42].

2.8.2 Beamforming and Dynamic Focusing

Beamforming and dynamic focusing are the reverse effects of transmit steering and focusing where the time delay system in receive mode is delayed to accept acoustic waves from a certain direction [27][28][42].

2.8.2.1 Depth of Penetration

The depth of penetration of the given ultrasound B-mode pulse is dependent on the attenuation that the ultrasound wave undergoes in the underlying tissue [28]. The depth of penetration is given by

$$d_p = \frac{L}{2af} \quad (35)$$

where L is the loss in dB, f is the transmit frequency, and a is the absorption coefficient [28].

2.8.2.2 Pulse Repetition Rate

The pulse repetition interval [28] is given by

$$T_R \geq \frac{2d_p}{c} \quad (36)$$

The pulse repetition rate [28] is given by

$$f_R = \frac{1}{T_R} \quad (37)$$

2.8.2.3 Image frame rate

If N pulses are needed to generate N A-lines of an image, then the B-mode image frame rate [28] is given by

$$F = \frac{1}{T_R N} \quad (38)$$

2.9 Image Quality

2.9.1 Resolution

Spatial resolution can be defined as the lowest possible distance between two objects/boundaries in an ultrasound image such that the two objects/boundaries can be distinguished [43]. In ultrasound, there are three types of resolution: axial, elevation, and lateral [43]. The axial resolution is dependent on the transducer frequency and the amplitude of the wave [43]. A higher transducer frequency implies a shorter pulse length [43]. The assumption here is that the number of cycles in each pulse is constant for a given system [43]. The lateral resolution is the resolution in the ultrasound image plane [43]. The lateral resolution depends upon the width of the ultrasound beam and pitch of the transducer elements [43]. The elevation resolution is dependent on the beam width in the direction perpendicular to the beam [43]. The elevation resolution is dependent on the transducer element, and there little a user can do to change this parameter [43].

Contrast resolution is generated primarily by the acoustic impedance difference that reflects ultrasound waves from the organ boundaries [27]. The attenuation difference can

be adjusted in the gray scale dynamic range to generate gray scale images [27]. Help from contrast agents can also improve the contrast by introducing microbubbles that reflect the acoustic waves from their surface and help in visualizing tissue perfusion [27].

2.9.2 Noise, Artifacts, and Speckle

2.9.2.1 Noise

In general, noise can be defined as any random artifact or a pseudo signal that is mistaken for the original signal [44]. Several factors contribute to noise, including power fluctuations, small noises amplified by the electronic amplifiers, dead transducer elements, and equipment malfunction [27]. For a good contrast-to-noise ratio, low noise and a high gain amplifier are important, as these parameters often contribute to noise amplification at greater depths [27]. A time gain compensation (TGC) technique is used to amplify and exponentially attenuate an ultrasound beam [27].

2.9.2.2 Artifacts

Artifacts are generated due variations in the seemingly fixed average wave speed of 1540 m/s in tissues throughout the body [27]. This variation leads to incorrect application to the range equation and the generation of an incorrect map [27]. The generated image map has visible artifacts that can sometimes be used for diagnostic purposes, such as speckle [27]. When the beam is not perpendicular to the tissue boundary, it can become refracted and, deflected to the receiver from neighboring areas of the receiver beam [27]. A nonuniform texture is generated based on the improper estimation of TGC [27]. Some of the common artifacts are refraction, shadowing and enhancement, reverberation, speed displacement,

side lobes and grating lobes, multipath reflection and mirror images, ambiguity, and slice thickness [27].

Anatomic displacement, which changes with different transducer angle and position is generated when a transducer pulse hits the boundary of a tissue at a perpendicular angle, and the tissues have different speeds of sound [27]. This type of artifact generated by refraction [27].

Highly attenuating objects, such as bones, reflect or highly attenuate incident beams [27]. These objects create a shadow that is a hypointense signal [27]. This shadow is created away from the center of the object [27]. Streaks are caused when the incident beam hits a curved surface [27]. Whereas, enhancement occurs when the incident wave hits a fluid-like low attenuation object, generating hyperintense signals [27]. A hyperintense signal has increased transmission [27].

A reverberation artifact is caused by multiple echoes between highly reflective surfaces or a highly reflective surface and the transducer [27]. These artifacts appear as multiple spaced boundaries of equal length with a width or amplitude that decreases in the direction perpendicular to the transducer [27].

Due to an assumed constant speed of 1450 m/sec, a speed artifact occurs, which causes inaccuracy in the spatial measurements made with ultrasound [27]. A subtle difference between displaced and non-displaced tissue is observed, where the displaced tissue is mapped outwards about its original position [27].

Sometimes, an organ, such as the liver, produces multiple reflections and refraction waves towards the diaphragm [27]. These waves are exchanged continuously between the liver and

the diaphragm [27]. After a certain period, the waves are reflected from the diaphragm to the transducer [27]. This effect creates a mirror-like image of the organ on the screen [27].

The expansion of the piezoelectric element in the orthogonal direction of the beam leads to side lobes in the ultrasound image [27]. Echoes from side lobes are imaged in the same ultrasound beam, and reflection of such a beam from a highly reflective surface gives the appearance of a diffuse echo in normally hypoechoic organs [27]. On the other hand, the grating lobes are generated from the many elements on the surface of the transducer creating out-of-direction beams that give ghost artifacts that appear as high-contrast objects that are not along the same axis [27].

2.9.2.3 Speckle

Speckles are generated from the interference of scattered waves deflected from scatterers inside a given resolution cell. Fully developed speckle develops when the scatterer density inside a given resolution cell is at its maximum [45]. For a fully developed speckle, the detected envelope signal possesses a Rayleigh distribution with an SNR of 1.91 and a B-mode signal with an SNR of 1 [45]. For tissue characterization, speckle statistics below the SNR of a Rayleigh distribution are useful [45]. The characterization of the speckle depends on the scatterer distribution and the resolution cell volume [45]. Of the scatterer distribution and the resolution cell volume, only the resolution cell volume can be controlled by the imaging system [45].

When the same transducer and underlying conditions are utilized, the speckle appears to be the same; however, the speckle changes when the transducer and underlying conditions change [46]. Hence, speckles are not non-random noise like electronic noise [46]. This

property of a speckle can be used to track the movement of tissue inside ultrasound RF or B-mode signals [1].

A common speckle reduction technique is ultrasound compounding [46]. In ultrasound compounding a region under consideration is scanned at a different angle and at different amplitudes [46]. The transducer must be translated by half of its width during speckle reduction [46].

2.10 Elastography

Elastography is a relatively new modality in ultrasound that is used for the detection of cancerous tumors [1]. Elastography provides information about the stiffness distribution in the imaged area [1]. A cancerous region that is less elastic than the surrounding tissue can be distinguished [1]. In quasi-static elastography, an external palpation is applied by using an ultrasound probe to measure the axial strain profile using motion estimation methods [1]. The applied stress distribution can then be used to generate an elastic modulus to generate an image called an elastogram [1]. In practice, the inverse elastic modulus is used instead to display the elastogram [1]. The inverse elastic modulus has a finite range compared to the values generated by the elastic modulus [1]. This finite range is important because the strain range has an upper bound that is equal to the applied input strain [1]. To generate a strain profile, we measure the local variations of the tissue strain in the axial direction as a function of depth [1]. These local axial variations occur due to the application of external axial compression [1]. This profiling can be repeated for all of the adjacent axial lines to obtain a two-dimensional strain profiles [1].

Elastography can be broadly categorized into static and dynamic elastography [42]. The static elastography technique or strain technique involves the palpation of the tissue externally with the transducers and the measurement of elasticity [42]. In contrast, dynamic elastography or shear wave elastography involves generating a shear-wave internal to the tissue and measuring the velocity of the shear-wave inside the tissue using ultrasound [42].

Young's modulus is defined by the ratio of the externally applied stress and the strain induced in the tissue [42], as follows:

$$\frac{stress}{strain} = E \quad (39)$$

The externally applied stress on the tissue can be defined as the ratio of the applied force to a given area of the tissue [42], as follows:

$$stress = \frac{force}{area} \quad (40)$$

where the force is the applied external force in the given area. The strain in the tissue is the ratio of the displacement of the tissue in the direction of the force and the original length of the tissue [42] and is given by:

$$strain = \frac{\Delta d}{d} \quad (41)$$

where d is the length of the tissue with a fixed width and Δd is the increase in the length of the tissue after applying the external stress. The axial compression, the compression in the direction parallel to the transducer, generates a lateral compression [42]. For every axial compression, a lateral compression occurs [42]. Soft lesions expand in the lateral direction

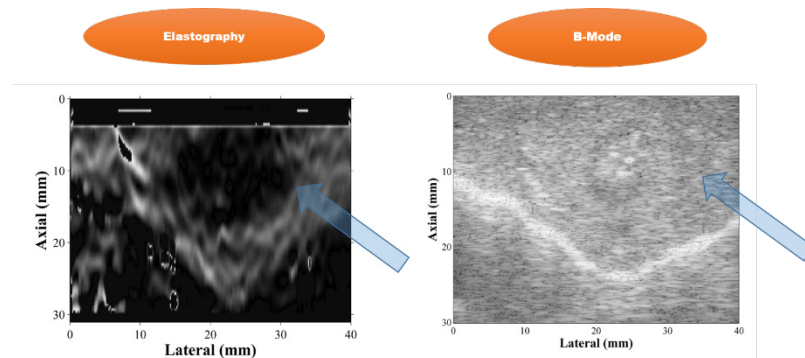
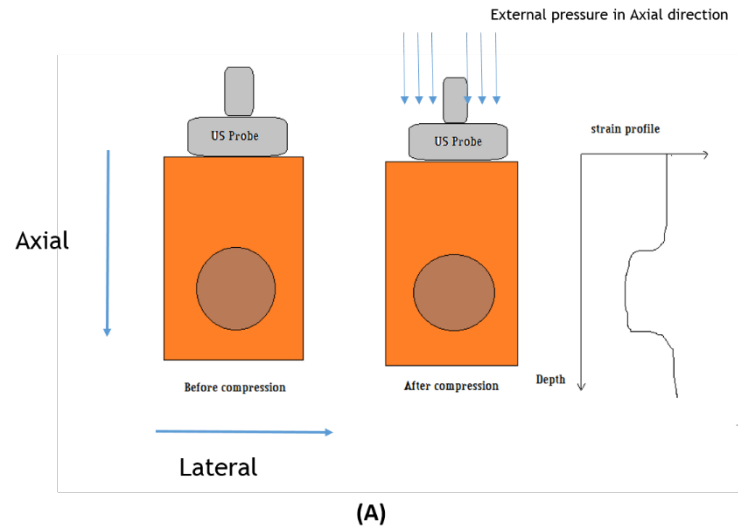
more than the surrounding tissue [42]. In contrast, a hard lesion expands less than the soft lesion and the surrounding tissue [42]. The value of strain decreases as the stiffness of the tissue increases, with zero as the lowest value for strain, which corresponds to very hard lesions [42]. The strain-ratio is given by

$$\text{strain - ratio} = \frac{\text{background - strain}}{\text{lesion - strain}} \quad (42)$$

where the background-strain is the strain calculated in the background region or soft lesion, and the lesion-strain is the strain calculated in the target lesion area [42]. Because the external stress is difficult to measure in order to quantify the strain profile using the Young's modulus, a strain-ratio can be used as a measure to estimate lesion hardness [42]. It may be advantageous to convert the strain profile to the elastic modulus because the absolute strain value is proportional to the initial compression, and any defect in the initial strain value is propagated along the axial direction [1]. Additionally, the elastic modulus is also a reliable tissue property [1]. Strain profiling is a limited measure due to different strain profiles across the tissue, and even if the external compression is constant, this parameter varies while propagating in the axial direction [1]. For this reason, it is beneficial to apply a known stress instead of a known displacement to estimate the elastic modulus that is independent of the initial compression [1].

Ultrasound elastography depends on speckle tracking to find the displacement in the tissue along the axial direction [1]. Two scans are needed to perform elastography. One scan is performed before the compression of the tissue, and the second scan is performed immediately after compressing the tissue [1]. Because the cancerous region is less elastic than the surrounding neighboring tissue, the strain can be calculated based on the relative

difference in the displacement of different regions in the tissue [1]. This method of palpation is similar to hand based palpation observation, in which the presence of a hard lump is detected by pressing the skin at different locations [1].



Comparative images for an in vivo ablation experiment on pig liver.

Figure 2.1 Elastography concept: Panel (A) shows the comparison of two RF-lines pre- and post-compression, that gives a strain profile to indicate the stiffness inside the tissue. Panel (B) shows a side-by-side comparison of elastography and B-mode data [2]. The ablated region is clearly visible in the elastography image, and the B-mode image shows the boundary between two layers [2].

The axial A-line is divided into different segments for the pre- and post-compressed images [1]. These segments are compared to pre- and post-compressed images using displacement

estimation techniques, such as normalized cross-correlation [1]. The segment size defines the accuracy of the search. However, a smaller segment size is preferred to improve the axial resolution of the elastogram [1]. Similarly, a cross-correlation estimation deteriorates with increasing segment size due to error propagation in the axial direction [1]. The resolution is also dependent on the sampling rate and can be improved by performing the scan using interpolation algorithms [1].

A previous axial stress distribution study [1] indicates that the uniform stress distribution assumption works well as long as the compressor sizes are greater than or equal to the axial depth. The axial stress distribution for a semi-infinite elastic medium [47] is given by

$$\sigma(z) = \sigma(0) \left(\frac{1}{\left(1 + \left(\frac{a}{z}\right)^2\right)^{3/2}} - 1 \right) \quad (43)$$

where $\sigma(z)$ is the axial stress, $\sigma(0)$ is the uniform stress distribution applied by the circular compressor, a is the radius of the compressor, and z is the depth of penetration in the axial direction [47]. This equation also implies that the strained quality is better near the edge of the tissue near the transducer, and the quality decreases with increasing depth [42].

The lateral resolution of the elastogram is dependent on the resolution of the underlying RF data, typically proportional to the number of transducer elements [1]. The axial resolution is dependent upon several factors that affect the quality of elastogram, including the desired contrast-to-noise ratio (CNR), the desired signal-to-noise ratio (SNR), the

central transducer frequency, the bandwidth, overlapping of the underlying correlation function window, and the sampling frequency among many parameters [1].

The CNR measure of the ultrasound elastography [48] is given by

$$CNR = \sqrt{\frac{2(\bar{a}_b - \bar{a}_t)^2}{\sigma_b^2 + \sigma_t^2}} \quad (44)$$

where \bar{a}_b and \bar{a}_t are the mean intensity values of the background and the target region and σ_b and σ_t are the standard deviations of the intensity values of the background and the target region [48]. The SNR measure of the ultrasound elastography [48] is given by

$$SNR = \frac{\bar{a}}{\sigma} \quad (45)$$

where \bar{a} and σ are the mean and standard deviation values of the intensity measures in the entire elastogram [48].

3 Motion estimation algorithms for displacement and strain estimation in elastography

3.1 Normalized cross-correlation (NCC), sum of squared difference (SSD), and sum of absolute difference (SAD)

3.1.1 Spatial Domain

The early implementations of elastography were implemented using a normalized cross-correlation (NCC) based matching technique [1]. The NCC [49] score can be given by:

$$NCC_{m,n} = \frac{\sum_{i=1}^l \sum_{j=1}^k (X_{i,j} - \bar{X})(Y_{i+m,j+n} - \bar{Y})}{\sqrt{\sum_{i=1}^l \sum_{j=1}^k (X_{i,j} - \bar{X})^2 \sum_{i=1}^l \sum_{j=1}^k (Y_{i+m,j+n} - \bar{Y})^2}} \quad (1)$$

where X is the pre-compressed image, Y is the post-compressed image, \bar{X} is the mean of the intensity values in the pre-compressed image, \bar{Y} is the mean of the intensity values in the post-compressed image, m and n are the indices of the starting region of the search

region in the post-compressed image, and l and k are the axial and lateral dimensions of the windowed region [49]. We compare the windowed region of the pre- and post-compressed images, and we obtain the displacement in that particular window [1]. A cosine fit interpolation is applied to obtain a sub-integer estimation, followed by a median and moving average reduce the outliers [1]. This refined displacement map generates a strain profile after least squares estimation [1]. The small window size is beneficial for obtaining better axial resolution [1].

Another algorithm that is used to estimate the displacement is the sum of squared differences [50][49], given by:

$$SSD_{m,n} = \sum_{i=1}^l \sum_{j=1}^k (X_{i,j} - Y_{i+m,j+n})^2 \quad (2)$$

Similarly, the sum of absolute differences algorithm [49] is given by:

$$SAD_{m,n} = \sum_{i=1}^l \sum_{j=1}^k |X_{i,j} - Y_{i+m,j+n}| \quad (3)$$

Both SSD and SAD use the windowing techniques highlighted for NCC [50][49].

3.1.2 Spectral Domain

Spectral shift elastography involves measuring the shift in the frequency domain to estimate the underlying strain. The normalized cross-correlation [51] defined below can be used to estimate this shift:

$$R_{NCC}(k) = \frac{\sum_{m=1}^{N/2} S_0(m) \cdot S_1(m+k)}{\sqrt{\sum_{m=1}^{N/2} S_0(m) \sum_{m=1}^{N/2} S_1(m)}} \quad (4)$$

where S_0 is the pre-compressed spectral image, S_1 is the post-compressed spectral image, N is the maximum window length of the spectral density, and the limit of $N/2$ is based on the folding frequency of the power spectra [51]. This equation is evaluated for different integer values of k to determine the strain estimate [51] using the following formula:

$$s = -\frac{k}{k_0} \quad (5)$$

where k_0 is the center frequency that corresponds to s_0 [51].

The sum of squared difference (SSD) [51] method is given by:

$$R_{SSD}(k) = \sum_{m=1}^{N/2} (S_0(m) - S_1(m+k))^2 \quad (6)$$

The sum of absolute difference (SAD) [51] method is given by:

$$R_{SAD}(k) = \sum_{m=1}^{N/2} |S_0(m) - S_1(m+k)| \quad (7)$$

The NCC, SSD and SAD methods use the same principles for spectral information extraction [51]. The spectral information is estimated by dividing the radio-frequency data into several windows [51]. An N -length fast Fourier transform is used to extract this temporal information. In contrast, to obtain the power spectrum, the magnitude of the spectral information is squared and normalized by the length of the spectral window [51].

NCC, SSD, and SAD can be extended to 3D by extending the 2D windows to 3D sub-volumes and searching in these 3D sub-volumes to obtain the displacement estimates.

3.1.3 3D cross-correlation method

A small variant of the normalized cross-correlation method for 3D motion estimation was presented in a previous study [52]. Every pixel in the pre-compressed radio-frequency volume I_t is tracked by defining a 3D kernel with a size of one speckle around that pixel [52]. This speckle size is determined from the full-width and half maximum of the pre-compressed volume [52]. Similarly a 3D kernel in the post-compressed volume is represented by I_{t+1}^* [52]. The 3D cross-correlation function is given by:

$$\rho'_{xyz}(l_x, l_y, l_z) = \frac{\sum_i \sum_j \sum_h W_{ijh} [I_t(x+i, y+j, z+h) I_{t+1}^*(x+l_x+i, y+l_y+j, z+l_z+h)]}{\left[\sum_i \sum_j \sum_h W_{ijh} |I_t(x+i, y+j, z+h)|^2 \right]^{1/2} \left[\sum_i \sum_j \sum_h W_{ijh} |I_{t+1}^*(x+l_x+i, y+l_y+j, z+l_z+h)|^2 \right]^{1/2}} \quad (8)$$

where (x, y, z) is the pixel/voxel location in the given volumes, (l_x, l_y, l_z) is the lag or displacements at which to start scanning in the post-compressed volume, I_t and I_{t+1} are the pre-compressed and post-compressed volumes respectively, the index (i, j, h) iterate through every pixel inside the given 3D kernels, and W_{ijh} are the weighting functions [52].

A unity gain function $\sum_{i,j,h} F_{ijh} = 1$ is used to extract the final 3D correlation coefficient ρ_{xyz}

, that is used to estimate the displacement in all the three directions [52]:

$$\rho_{xyz}(l_x, l_y, l_z) = \sum_i \sum_j \sum_h F_{ijh} [\rho'_{x+i, y+j, z+h}(l_x, l_y, l_z)] \quad (9)$$

3.2 Optical Flow

The optical flow technique was used in a previous study [53] to measure the elastography of the patients' skin using high-definition (HD) cameras. This technique involves subtle palpation with the hands on the skin of the patient and cameras to record the movements [53]. A dense motion field is created, from which the strain is calculated [53]. This technique is non-invasive and is useful when the instruments cannot touch the skin of the patients, (e.g. on a burned skin surface) [53].

A second order partial differential equation between two consecutive images is used to estimate a dense motion field in the optical flow technique [53]. The underlying condition for optical flow is the brightness constancy equation [53] given by:

$$(\nabla g)^T \bar{u} + g_t = 0 \quad (10)$$

where $g(x, y, t)$ are the spatial and temporal representation of the input image holding the input image with a parameter t that stands for time and (x, y) indicates the spatial location for an image with a given t [53]. In contrast, ∇g is the spatial derivative and g_t is the temporal derivative with $\bar{u} = (u, v)^T$ as the flow vector [53]. For the optical flow to minimize errors the necessary conditions are as follows: (a) the brightness or intensity of the object in motion inside the image remains constant and (b) the pixels in the given kernel around a reference pixel have a similar speed [53].

The strain map can be represented by the Cauchy strain tensor [53] (ε):

$$\varepsilon = \frac{1}{2} [\nabla \bar{u} + (\nabla \bar{u})^T] \quad (11)$$

where $\nabla \bar{u}$ is defined by [53]:

$$\nabla \bar{u} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}. \quad (12)$$

Combining the above two equations, we obtain a revised strain map as follows [53]:

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} = \frac{\partial u}{\partial x} & \varepsilon_{yx} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \varepsilon_{xy} = \frac{1}{2} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & \varepsilon_{yy} = \frac{\partial v}{\partial y} \end{bmatrix} \quad (13)$$

where the spatial derivatives are given by the convolution of Sobel filters and their respective directional components of \bar{u} [53], as shown in the following equations:

$$\frac{\partial u}{\partial x} = S_x * u \quad \frac{\partial v}{\partial x} = S_x * v \quad (14)$$

$$\frac{\partial u}{\partial y} = S_y * u \quad \frac{\partial v}{\partial y} = S_y * v. \quad (15)$$

The Sobel filters [53] in the (x, y) direction are given by

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (16)$$

Finally, the strain map [53] is given by

$$\varepsilon_m = \sqrt{\varepsilon_{xx}^2 + \varepsilon_{xy}^2 + \varepsilon_{yx}^2 + \varepsilon_{yy}^2} \quad (17).$$

A large cumulative displacement estimate for better strain imaging can be achieved by adding a series of small motions, as follows:

$$u(i, i+n) = u(i, i+1) + u(i+1, i+2) + \dots + u(i+n-1, i+n) \quad (18)$$

$$v(i, i+n) = v(i, i+1) + v(i+1, i+2) + \dots + v(i+n-1, i+n) \quad (19)$$

where the cumulative motion between frames i and $i+n$ is given by $u(i, i+n)$ and $v(i, i+n)$.

3.3 Phase-shift correlation method

In this method, the post-compression image wave r_{i1} is a time-scaled and time-shifted from the pre-compression image wave r_i [54], as follows:

$$\begin{aligned} r_i &= x(t) + n_i(t) \\ r_{i1} &= x((1+a_i)t + D_i) + n_{i1}(t) \quad -\frac{T}{2} \leq t - T_i \leq \frac{T}{2} \end{aligned} \quad (20)$$

where $n_i(t)$ and $n_{i1}(t)$ are the random noise in the pre- and post-compressed RF images [54]. The displacement and the rate of displacement with respect to time T in the post-compressed RF image are given by D_i and a_i respectively. The relationship between the rate of displacement and the strain s is given by $a = s / (1 - s)$ [54].

The time delay estimate between the initial frame and the i^{th} frame is given by multiplying $r_i(t)$ and $r_{i1}(t)$ by a weighted function $w(t)$ and taking the cross-correlation [54] as follows:

$$\phi_{rr}(\hat{D}_i) = \max_{\tau} \int_{-\infty}^{\infty} dt \quad w(t + \tau - T_i)r_i(t + \tau) \times w(t - T_i)r_{i1}(t) \quad (21)$$

where \hat{D}_i or the displacement estimate corresponds to the value of τ when ϕ_{rr} is maximized [54].

3.4 Efficient phase zero search algorithm

An efficient phase zero search was introduced in a previous study [55], in which a complex correlation coefficient was calculated as:

$$\langle a, b \rangle(n\Delta t, -t') = \sum_{t=n\Delta t}^{n\Delta t+T} a^*(t)b(t-t') \quad (22)$$

where a and b are the input complex signals, T is the window length, the search starts from $n\Delta t$, and t' is the displacement of the window in the post-compression image b [55]. Here, a is the pre-compression image [55].

To obtain a better linear interpolation accuracy for subsample precision, a baseband analytical signal is calculated with a suitable modulation frequency ω_0 [55]. The conversion to the baseband analytical signal is given by

$$a_b(t) = a(t)e^{-j\omega_0 t} \quad (23)$$

The successive iterative displacement estimates t'_k and t'_{k+1} for a given window are calculated using the iterative gradient descent method [55], as follows:

$$t'_{k+1} = t'_k + \frac{\arg(e^{-j\omega_c t'_k} \langle x_{b1}, x_{b2} \rangle (n\Delta t, -t'_k))}{\omega_c} \quad (24)$$

where x_{b1} and x_{b2} are the pre- and post-compression analytical baseband signals, the local frequency centroid is ω_c , the initial estimate for the 0th frame $t'_{0,0}$ is initialized to 0 and each consecutive displacement from the previous window is used to initialize the next window search (i.e., $t'_{0,n+1} = t'_{K,n}$). The arg function returns values in the range $-\pi$ to $+\pi$ [55].

3.5 Dynamic Programming

First we summarize the 1D Dynamic programming method [56]. The sum of absolute differences Δ is given by

$$\Delta(i, d) = |g(i) - g'(i + d)| \quad (25)$$

where $g(i)$ and $g'(i)$ are the pre- and post-compression RF data images, $d_{\min} \leq d \leq d_{\max}$ is the displacement of the sample at location i , and d_{\min} and d_{\max} are the user controlled minimum and maximum displacement ranges [56]. To reduce the impact of the change in the gain of the RF signals, the pre- and post-compression images are divided by the maximum intensity values between the two images [56].

The smoothness of the displacement [56] is given by

$$S(d_i, d_{i-1}) = (d_i - d_{i-1})^k \quad (26)$$

where d_i and d_{i-1} are the displacement of two consecutive samples in the image $g(i)$

[56]. To avoid large jumps in S, S should follow the following strictly convex criteria

[56]:

$$\alpha(d_{i1} - d_{i-1})^k + (1 - \alpha)(d_{i2} - d_{i-1})^k > [\alpha d_{i1} + (1 - \alpha)d_{i2} - d_{i-1}]^k, 0 < \alpha < 1 \quad (27)$$

Here the value of $k=2$ and for the values $k>2$, the jumps in S can be drastic and affect CNR ratio [56].

The 1D cost function C for a given sample location i and related displacement d_i [56] is given by

$$C(i, d_i) = \min\{C(i-1, d_{i-1}) + wS(d_i, d_{i-1})\} + \Delta(i, d_i) \quad (28)$$

where w is the regularization weight to control smoothness [56]. The displacement d_i that minimizes C_i is stored in M_i [56], that is given by:

$$M(i, d_i) = \arg \min_{d_{i-1}} \{C(i-1, d_{i-1}) + wS(d_i, d_{i-1})\} \quad (29)$$

The cost function C_i is calculated for $i=1 \dots m$ where m is the maximum number of samples in the axial direction [56].

The per sample displacement [56] is calculated as follows:

$$D(i) = \arg \min_{d_i} \{C(i, d_i)\}, i = m \quad (30)$$

$$D(i) = M(i+1, D(i+1)), i = 1 \dots m-1 \quad (31)$$

Similarly, the 2D displacement is generated using the sum of absolute difference in the lateral directions [56], as follows:

$$\Delta(i, j, d_a, d_l) = \left| g_j(i) - g'_{j+d_l}(i + d_a) \right| \quad (32)$$

where d_l is the displacement in the lateral direction, d_a is the displacement in the axial direction, and the index $j=1 \dots n$ traverses the n A-lines [56].

The 2D smoothness regularization term [56] is given by:

$$S(d_a, d_l, d_{a_{i-1}}, d_{l_{i-1}}) = (d_a - d_{a_{i-1}})^2 + (d_l - d_{l_{i-1}})^2 \quad (33)$$

Similar to the 1D case, the 2D cost function [56] is given by

$$C_j(d_a, d_l, i) = \Delta(d_a, d_l, i) + \min_{\delta_a, \delta_l} \left\{ \frac{C_j(\delta_a, \delta_l, i-1) + C_{j-1}(\delta_a, \delta_l, i)}{2} + wS(d_a, d_l, \delta_a, \delta_l) \right\} \quad (34)$$

The displacement map can be generated as described for the 1D case [56].

3.6 Analytical Minimization

The 2D Analytical minimization method uses the non-integer displacements calculated from the integer displacements of the 2D Dynamic programming method via interpolation methods, with the initial seed displacements a_i and l_i [57]. The $(\Delta d_l, \Delta a_i)$ values minimize the following regularized cost function:

$$C_j(\Delta a_1, \dots, \Delta a_m, \Delta l_1, \dots, \Delta l_m) = \sum_{i=1}^m \left\{ \begin{aligned} & [I_1(i, j) - I_2(i + a_i + \Delta a_i, j + l_i + \Delta l_i)]^2 + \\ & \alpha (a_i + \Delta a_i - a_{i-1} - \Delta a_{i-1})^2 + \\ & \beta_a (l_i + \Delta l_i - l_{i-1} - \Delta l_{i-1})^2 + \beta'_l (l_i + \Delta l_i - l_{i,j-1})^2 \end{aligned} \right\} \quad (35)$$

where $I(i, j)$ is the i^{th} and j^{th} index in the image, $l_{i,j-1}$ is the total lateral displacement of the previous A-line, and $\alpha > 0$, and $\beta_a > 0$ are the tunable axial and lateral regularization parameter. The parameter β'_l helps to reduce the propagation of errors from one RF-line to the next RF-line [57] and is given by:

$$\beta'_l = \frac{\beta_l}{1 + |r_{i,j-1}|} \quad (36)$$

Here $r_{i,j-1}$ is the residual associated with the errors from the previous RF-lines [57]. We simplify the term related to the post-compressed image by applying Taylor's expansion [57], as follows:

$$I_2(i + a_i + \Delta a_i, j + l_i + \Delta l_i) \approx I_2(i + a_i, j + l_i) + \Delta a_i I'_{2,a} + \Delta l_i I'_{2,l} \quad (37)$$

Where $I'_{2,a}$ and $I'_{2,l}$ are the derivatives of I_2 for the location $(i + a_i, j + l_i)$ [57].

We now arrange the analytical minimization function in the following form

$$(I_2'^2 + D_1 + D_2)\Delta d = I_2' l - D_1 d \quad (38)$$

$$D_1 = \begin{bmatrix} \alpha & 0 & -\alpha & 0 & 0 & 0 & \cdots & 0 \\ 0 & \beta_a & 0 & -\beta_a & 0 & 0 & \cdots & 0 \\ -\alpha & 0 & 2\alpha & 0 & -\alpha & 0 & \cdots & 0 \\ 0 & -\beta_a & 0 & 2\beta_a & 0 & -\beta_a & \cdots & 0 \\ 0 & 0 & -\alpha & 0 & 2\alpha & 0 & \cdots & 0 \\ \vdots & & & & & & \ddots & \\ 0 & 0 & 0 & \cdots & -\alpha & 0 & \alpha & 0 \\ 0 & 0 & 0 & \cdots & 0 & -\beta_a & 0 & \beta_a \end{bmatrix} \quad (39)$$

where $\Delta d = [\Delta a_1 \quad \Delta l_1 \quad \Delta a_2 \quad \Delta l_2 \quad \cdots \quad \Delta a_m \quad \Delta l_m]^T$,

$e = [e_1 \quad e_1 \quad e_2 \quad e_2 \quad \cdots \quad e_m]^T$, $e_i = I_1(i, j) - I_2(i + a_i, j + l_i)$, D_2 is the diagonal matrix of size $2m \times 2m$ is given by $D_2 = \text{diag}(0, \beta'_1, 0, \beta'_1, \cdots, 0, \beta'_1)$, $I_2'^2$ is the symmetric tri-diagonal matrix of size $2m \times 2m$ $I_2'^2 = \text{diag}(J'^2(1) \cdots J'^2(m))$ [57] contains

$$J'^2(i) = \begin{bmatrix} I'_{2,a}{}^2 & I'_{2,a} I'_{2,l} \\ I'_{2,a} I'_{2,l} & I'_{2,l}{}^2 \end{bmatrix}$$

where the axial and later derivative of I_2 is given by $I'_{2,a}$ and $I'_{2,l}$ at location $(i + a_i, j + l_i)$ [57]. The derivative of I_2 is given by

$$I'_2 = \text{diag}(I'_{2,a}(1), I'_{2,l}(1), I'_{2,a}(2), I'_{2,l}(2) \cdots I'_{2,a}(m), I'_{2,l}(m)) \quad (40)$$

The inverse gradient estimation and the final 2D AM displacement [57] are derived from the following expression:

$$(WI_1'^2 + ZD_1 + ZD_2)\Delta d = WI_1'e - ZD_1d + s \quad (41)$$

4 Elastography using Multi-Stream GPU

This chapter is from [2] and published with the following citation:

Nishikant P. Deshmukh, Hyun Jae Kang, Seth D. Billings, Russell H. Taylor, Gregory D. Hager, and Emad M. Boctor, "Elastography Using Multi-Stream GPU: An Application to Online Tracked Ultrasound Elastography, In-Vivo and the da Vinci Surgical System," PLoS ONE 9(12): e115881. doi:10.1371/journal.pone.0115881. PLoS ONE is an open access journal and contents are reusable under Creative Commons Attribution (CC BY) license.

In this chapter, we present an end-to-end real-time system which improves the speed of GPGPU-based implementation of normalized cross-correlation (NCC) elastography using the stream capability of CUDA. This real-time system receives radio frequency (RF) data from an ultrasound machine and processes it on a GPGPU to compute an elastography image. We designed our system to harness the CUDA stream and multiple instruction multiple data (MIMD) capability of modern GPGPU architectures. Typical elastography calculations involve several computationally intensive components including displacement map generation, post processing filters, strain calculation, dynamic range adjustment, and scan conversion. Each of these components is mapped to a CUDA kernel within the GPGPU. CUDA kernels are the basic parallelizable blocks in the CUDA programming language, similar to a function. Using CUDA stream functionality these kernels are connected to form an input-output pipeline. A CUDA stream ensures data integrity by limiting inter-component data access to within the pipeline, thereby enabling multiple CUDA streams to run in parallel. We present the benefit of our work through speed comparison of elastography on multi-stream GPU architecture, single-stream GPU architecture, and non-stream GPU architecture. The new system has achieved an

elastography image generation rate of up to 78 frames per second, nearly matching the RF data acquisition rate of ultrasound machines. We further investigate the impact of NCC window size on both speed and quality of elastography images using in-vivo pig liver data.

To showcase the adaptability of our architecture we demonstrate two applications: real-time elastography by free-hand palpation using external tracking information (Online tracked ultrasound elastography (O-TRuE)), and integration with the da Vinci Surgical System for elastography by robot-assisted palpation. The original TRuE [58] method was an offline system where the RF data and tracking data was collected offline, timestamp synchronization was performed in matlab, frame selection was done using TRuE and finally the elastography was calculated for the chosen pair of RF data. There was no real-time feedback to the surgeons, and this problem was solved using O-TRuE method. O-TRuE is an end-to-end system which involves RF data acquisition from an ultrasound machine and tracking data acquisition from an EM tracking device, synchronizing these acquired data based on timestamp, passing this data to a selection engine which performs in-plane RF data frames search using TRuE, implementation of a queue mechanism to streamline TRuE calculation and elastography computation. Furthermore, we devise a technique for output elastography image stream analysis, which we use to investigate improvement in the output stream quality of O-TRuE relative to untracked free-hand palpation. We also apply this analysis to evaluate the quality of output elastography images for different palpation motions generated by the da Vinci system. Finally, we demonstrate how multiple O-TRuE images combined by weighted averaging produce a higher quality elastography image, which we analyze using contrast-to-noise ratio (CNR) and signal-to-noise ratio (SNR) values.

Individual contributions

Author initials: Nishikant P. Deshmukh (**NPD**), Hyun Jae Kang (**HJK**), Seth D. Billings (**SDB**), Russell H. Taylor (**RHT**), Gregory D. Hager (**GDB**), and Emad M. Boctor (**EMB**)

Author contributions in the published version of this chapter [2]:

Following are the contributions from authors in the published version of this chapter [2], there are additional work done by NPD in this chapter:

1. Overall system integration: NPD
2. Multistream GPU elastography algorithm development: NPD
3. Online tracked ultrasound elastography (O-TRuE) development: NPD
4. Elastography image stream analysis method: NPD
5. The da Vinci Surgical system integration – robot side development: SDB
6. The da Vinci Surgical system integration – ultrasound side development: NPD HJK
7. Conceived and designed the experiments: NPD HJK SDB RHT GDH EMB.
8. Performed the experiments: NPD HJK SDB.
9. Analyzed the data: NPD GDH EMB.
10. Contributed reagents/materials/analysis tools: NPD HJK.
11. Wrote the paper: NPD HJK SDB RHT GDH EMB.

Systems contributions

The challenge of developing a real-time system is ensuring that there are no blocking processes waiting on input data, disk writing or network writing. This is not a trivial problem to solve, and major research worldwide is focused on ensuring that the system performs in real time. In our case, the challenge complex and noisy dataset of ultrasound, a real-time modality. Often, improving the performance of the system involves downgrading the quality of the output images. In this chapter, I designed a multi-stream elastography system that is geared towards ensuring that there is minimal overhead in the system and that the output quality is high. The system that I have designed involves a queueing mechanism, a massively multi-threaded system on both CPUs and GPUs, the use of data structures to ensure optimal thread utilization, and asynchronous data transfer to the GPUs. Much of the discussion can be found in the Methods section and in Pseudocode 4.1-4.8.

Summary

A system for real-time ultrasound (US) elastography will advance interventions for the diagnosis and treatment of cancer by advancing methods such as thermal monitoring of tissue ablation. A multi-stream graphics processing unit (GPU) based accelerated normalized cross-correlation (NCC) elastography, with a maximum frame rate of 78 frames per second, is presented in this paper. A study of NCC window size is undertaken to determine the effect on frame rate and the quality of output elastography images. This paper also presents a novel system for Online Tracked Ultrasound Elastography (O-TRuE),

which extends prior work on an offline method. By tracking the US probe with an electromagnetic (EM) tracker, the system selects in-plane radio frequency (RF) data frames for generating high quality elastograms. A novel method for evaluating the quality of an elastography output stream is presented, suggesting that O-TRuE generates more stable elastograms than generated by untracked, free-hand palpation. Since EM tracking cannot be used in all systems, an integration of real-time elastography and the da Vinci Surgical System is presented and evaluated for elastography stream quality based on our metric. The da Vinci surgical robot is outfitted with a laparoscopic US probe, and palpation motions are autonomously generated by customized software. It is found that a stable output stream can be achieved, which is affected by both the frequency and amplitude of palpation. The GPU framework is validated using data from in-vivo pig liver ablation; the generated elastography images identify the ablated region, outlined more clearly than in the corresponding B-mode US images.

4.1 Introduction

Quasi-static elastography involves comparing pre-compression and post-compression ultrasound (US) images to measure the displacement of speckles [1]. This measurement is used to determine elasticity of the tissue, which is useful in distinguishing hard and soft areas [1]. Visualization of the strain map calculated from this displacement can help identify tissue features, such as malignant tumors [1]. This technique is commonly known as elastography [1]. Elastography can be used as an early diagnosis tool for cancer, where early detection is critical in reducing the number of cancer related deaths [59].

Elastography has been evaluated in human trials for breast [60], [61], prostate [62], [63], liver fibrosis [14], [64], ovarian [65], skin [66], and thyroid cancers [67], [68]. Thermal ablation monitoring involves ablating the cancer tumor with RF ablator; an ultrasound guided needle is placed near the target region predetermined by a CT scan [69]. An ablated region increases the stiffness of the burned tissue, which is easier to visualize in elastography [69]. Elastography helps to accurately position the needle near the target region with the assistance of B-mode images and to monitor the size of the burn [69]. Ablation needs to be stopped for the acquisition of RF data; this duration needs to be very small to maintain the target ablation curve [69]. Collection of this data and calculating elastography in real-time are challenges.

Newer ultrasound imaging techniques like shear wave elastography [70] (focused ultrasound induced shear wave) and vibro-elastography (external vibration with a mechanical excitation) [71] can generate very high frame rates of up to 10 kHz and 300 kHz respectively [70], [71]. These techniques also use correlation to measure elasticity; hence a very high speed matching engine is needed. These techniques require special devices and US machines to record the RF data. Additionally, these systems are expensive and not widely available; hence a low-cost and high performing elastography implementation is necessary.

Elastography is computationally expensive. Given the high acquisition speed of modern US systems, there is need for a real-time implementation of elastography. This paper details a novel complete system of GPU-based elastography. The first known elastography implementation of a general-purpose graphics processing unit (GPGPU, commonly known

as GPU) was published by [72]. This implementation was based on time domain analysis of RF data. An implementation based on Fourier domain analysis was published in [73], where a hybrid CPU-GPGPU model was proposed. In this implementation the GPU computes displacement estimation using CUFFT library, whereas median filtering and strain estimation is performed by a CPU. This implementation [73] does not have a real-time pipeline to accept RF data from an acquisition system; moreover, the CPU implementation would increase the CPU utilization in an attempt to have a threaded model of this pipeline. This work was further extended to calculate the time constant estimator for visco-elasticity and poro-elastography [74]. Due to the CPU-GPGPU nature of the work, a threading model is difficult to synchronize and requires the stream scheduling capacity of Compute Unified Device Architecture (CUDA) [75].

Field programmable gates array (FPGA) and Digital signal processor (DSP) based implementations of elastography and ultrasound systems have been reported by [76]–[78]. FPGA are on-board chips which tightly integrate with the underlying ultrasound hardware, thereby helping these systems to obtain direct, rapid access to the raw data from the ultrasound transducers. This hardware is expensive and difficult to program. A GPU-based implementation is a much less expensive and more flexible option. Several ultrasound devices by companies such as Ultrasonix, Siemens, Philips, GE, Toshiba, Supersonic, and Hitachi come equipped with built-in elastography modules [79]–[81]. These machines generally use a CPU implementation, which puts a strain on the system resources. However, many of the existing ultrasound systems deployed around the world come equipped with external PCI express cards. In these cases, connecting an external GPU card to a machine is fairly straightforward.

Real-time feedback for intra-operative tasks needs fast elastography in order to correct the deformation caused by the movement of the organ, varying compression and the hand tremor of the operator [58]. Typically, these corrections need calculation of multiple pairs of elastography from a pair of RF data. When tracking information is acquired, good RF pairs can be presorted by exploiting the geometric position of the probe with respect to a reference tracker. An EM tracked ultrasound elastography method has been introduced by [58]. The disadvantages are that EM trackers cannot be used in ferromagnetic environments. The use of robot controlled motion inducers is another option. Real-time elastography on the da Vinci robotic system and on a snake robot have been integrated in [82] and [83], respectively. This system generates a pre-defined palpation motion to generate a high quality elastogram, but relies on the assumption that the underlying organ is attached to a rigid body. This motion can be compensated by a high speed real-time system to generate high-quality elastogram.

4.2 Background

Several supporting systems and methods are used in the development of this work. This section briefly introduces the reader to these concepts.

4.2.1 General Purpose Graphic Processing Units (GPGPU)

A GPGPU is composed of many core streaming processors working in synchronization with each other. Early models of GPGPU's were single instruction, multiple data (SIMD) processors, for which a single CUDA kernel executes on all cores at a given time [2].

Implementations based on this architecture were commonly limited by poor utilization of the GPGPU. Newer versions of GPGPU's, such as the Fermi-architecture from NVidia, resolve this problem by introducing multiple instructions, multiple data (MIMD) processors [75]. This paper exploits this new capability of the GPGPU by scheduling all elastography processing components into individual CUDA streams, which enables greater utilization of the GPGPU (henceforth referred to as GPU).

4.2.2 Normalized Cross-Correlation (NCC) based Elastography

Normalized cross-correlation (NCC) is used for calculating tissue displacements between pre- and post- compressed RF data images by measuring the speckle shift [72], [84].

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\left\{ \sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2 \right\}^{0.5}} \quad (1)$$

Equation 1 defines the NCC function for comparing two RF image regions where f is a template window in the first RF image and t is a target window in the second RF image, $\bar{f}_{u,v}$ and \bar{t} are the mean respective intensities within each window, and $x, y, u,$ and v denote pixel position within a windowed region [84]. This window based approach allows processing individual window searched on separate CUDA threads [72]. A correlation map generated from eq. 1 is used to build a displacement map using cosine fit interpolation [72]. Median and moving average filters are applied on this displacement map to remove outliers [72]. The median filter and the moving average filter again allow for individual output

pixels to be computed on different threads [72]. A strain map is finally generated from the displacement map using the least squares approach which can similarly be scheduled on individual threads [72].

4.2.3 Tracked Ultrasound Elastography (TRuE)

Foroughi et. al. [58] developed and validated TRuE on offline data using electromagnetic (EM) tracking of an ultrasound probe. As per [58], a cost function (eq. 2) is used to rank the quality of physical alignment between different RF data frame pairs, which is computed from the corresponding EM tracking data

$$\text{Cost}(D) = K_x D_x^2 + K_y \frac{|D_y - t_{opt}|^3}{D_y + c} + K_z D_z^2 \quad (2)$$

where $\{(D_x, K_x), (D_y, K_y), (D_z, K_z)\}$ are the displacements and sensitivities of the motion in the lateral, axial and out-of-plane directions respectively, calculated using tracking information in the pair of RF data. A user input $0.2 \leq t_{opt} \leq 0.4$ regulates the maximum displacement expected in the axial direction; $c = 0.0001$ is a small constant to compensate for zero compression. The input D to the cost function is the distance vector $[D_x \ D_y \ D_z]^T$, which is calculated as follows:

$$D_x = \sqrt{\frac{1}{y_2 - y_1} \int_{y_1}^{y_2} (-a_z y + t_x)^2 dy} \quad (3)$$

$$D_y = \sqrt{\frac{1}{x_2 - x_1} \int_{x_1}^{x_2} (a_x x + t_y)^2 dx} \quad (4)$$

$$D_z = \sqrt{\frac{1}{(x_2 - x_1)(y_2 - y_1)} \int_{x_1}^{x_2} \int_{y_1}^{y_2} (a_x y - a_y x + t_z)^2 dy dx} \quad (5)$$

where x_1 , y_1 , x_2 and y_2 form a region of interest for each frame. The axis-angle representation of the rotation between frames is $a = [a_x \ a_y \ a_z]^T$ and the relative translation is $t = [t_x \ t_y \ t_z]^T$. In our paper we have used $x_1 = -x'/2$, $x_2 = x'/2$, $y_1 = 0$ and $y_2 = y'$, where x' and y' denote the image pixel width and height, respectively. These values are multiplied by pixel spacing values to convert to millimeter scale. Since we are primarily interested in axial motion analysis, the axial sensitivity is controlled as follows

$$K_y = (4\tau_y^2)^{-1} \quad (6)$$

$$\tau_y = \sigma \times 0.1 \quad (7)$$

where σ is a user defined variable of type natural number in the range $1 \leq \sigma \leq 15$ [58]. The value of $\sigma = 1$ indicates lower sensitivity in the axial direction and $\sigma = 15$ indicates higher sensitivity. Similarly, K_x and K_z are defined as

$$K_x = (4\tau_x^2)^{-1} \quad (8)$$

$$K_z = (4\tau_z^2)^{-1} \quad (9)$$

where $\tau_x = \tau_z = 0.2$ is fixed in our experiments since we are interested only in axial direction and the value of 0.2 can accommodate our small motions in lateral and out-of-plane directions. Finally, we define pseudo correlation value as the exponential of the negative cost value (eq. 2) as

$$Crr(D) = e^{-Cost(D)} \quad (10)$$

which provides a value in the range [0, 1] to rank the quality of physical alignment between different RF frame pairs. RF frame pairs with high Crr values have image planes closely aligned in physical space, making them ideal candidates for elastography image computation. As previously described in discussion of eq. 2, from which Crr is derived, the Crr value is computed for an RF frame pair by analyzing the tracking information associated with each frame. This analysis is performed for all $\binom{N}{2}$ frame pair combinations amongst the N most recent RF frames stored in a buffer. The reader is referred to [58] for further details.

4.3 Methods

This section describes our approach to GPU-based elastography. The system setup and configuration is presented in subsection *System Overview*. Subsection *Multi-Stream GPU-based Elastography* describes the architecture of the multi-stream implementation of GPU-based elastography. The online tracking implementation is discussed in subsection *Online Tracked Ultrasound Elastography*.

4.3.1 Ethical Statement

Experiments were performed on a healthy pig as per the protocol number SW11M128 approved by Johns Hopkins University Institutional Care and Animal Use Committee. The

experiments were conducted on pig liver since the pig liver is anatomically close to human liver. The experiment in-vivo is needed to reflect the conditions during surgery and validate the algorithm. The data for this paper was reused from earlier study in [58] to minimize animal experiments needed. We also extensively performed experiments on phantom to measure speed and define metric for O-TRuE to minimize experiment on animals.

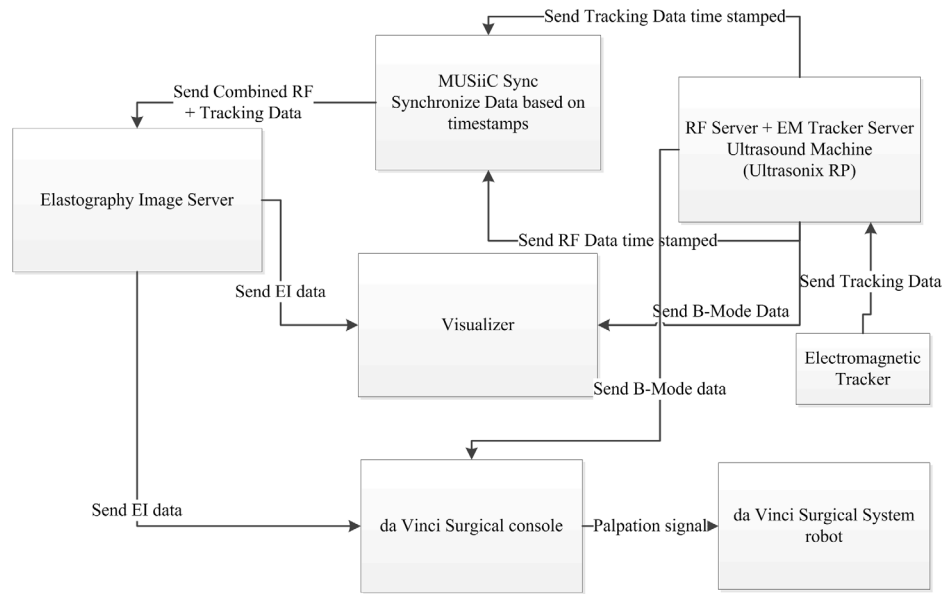


Figure 4.1 Overall System Diagram: The figure shows overall system and data flow diagram of elastography image server which runs on a machine equipped with a GPU. The system is modular with each module configurable to run on different machines or on a same machine (exception is hardware dependent da Vinci surgical system, RF Server and EM Tracker Server). The elastography image server is based on multi-stream elastography algorithm and with little change can handle both tracked and untracked RF data. The MUSiiC Sync synchronizes tracking and RF data based on timestamp to be processed by elastography image server. The system is flexible to be connected with da Vinci Surgical console to allow overlay of elastography and b-mode image stream.

4.3.2 System Overview

Figure 4.1 illustrates the overall system showing the various components comprising the real-time multi-stream GPU-based elastography system. This is an application view of the system by inclusion of the GPU, ultrasound machine, tracking system, da Vinci Surgical

System, image visualizer, and the MUSiiC Sync application. Communication between all system components is accomplished using the OpenIGTLinkMUSiiC library [23]. This library assists to make the system highly modular, allowing deployment of the components on different machines. The MUSiiC Sync application serves to synchronize all time-stamped data sources within the system. In our configuration, the MUSiiC RF server and MUSiiC EM tracker server were run on the ultrasound machine in order to get synchronized timestamps. The MUSiiC RF server, which is part of MUSiiCToolkit [23], collects RF data from the ultrasound machine and sends it to the network. This data can be collected by several listening clients. Similarly, the MUSiiC EM tracker server sends real-time tracking information to the network. MUSiiC Sync synchronizes the RF and tracking data based on their timestamps and forms a single data packet from each synchronized data pair which is sent to the Elastography Image Server. The Elastography Image Server processes the synchronized data to choose RF data pairs for elastography computation, which it computes using GPU. Output elastography images are then sent from the Elastography Image Server to a visualizer.

The EM tracker, which is attached to the ultrasound probe, provides the position of the probe in 3D space. Applying TRuE to the input stream of position data gives the best frame selection capability. In the da Vinci Surgical System environment, where usage of an EM tracker is not possible due to presence of ferromagnetic materials, we rely on steady palpations generated by robot to grant a good quasi-static elastography [82]. When no tracking data is available, we do not need MUSiiC sync. In this case, the RF Server directly outputs to the Elastography Image Server. An advantage of this system is its high modularity, enabling various software modules to lie on the same machine or different

machines. In some of the experiments two elastography image servers are run on the same computer using different GPU cards; one computing O-TRuE elastography and the other computing untracked elastography.

<p><i>A. Network receiver:</i></p>	<p><i>D. Online-Tracked Ultrasound Elastography(O-TRuE)</i></p>
<ol style="list-style-type: none"> 1. Receive RF data on OpenIGTLink and call a predefined call back function to send data over the queue Q. 2. Call back function inserts data at the end of a queue. 3. Invoke wake call to the task distributor thread (in B.3, D.1 and D.3) which invoked wait call on this queue. 4. This function is called again after the next RF data packets are received. 	<ol style="list-style-type: none"> 1. Wake up on data receive on the queue Q (from A.1). 2. Accumulate n-1 RF data frames in RF-Data-buffer. 3. Wake up on data receive on the queue Q and add the RF data frame at the end of RF-Data-buffer. 4. k:=1; 5. for i=1 to n-2 <ol style="list-style-type: none"> for j=i+1 to n <ol style="list-style-type: none"> a. if i==j break; b. Crr(k)=TRuE(RF-Data-buffer(i), RF-Data-buffer(j)) [Crr from section II. C] c. k:=k+1;
<p><i>B. Task Distributor thread:</i></p>	
<ol style="list-style-type: none"> 1. Initialize an array (thread-handle-array) of size L to handle threads for an accumulator. 2. Wake up on data receive on the queue Q (from A.1) and add a RF data frame at the end of RF-Data-buffer. 3. for thread_count = 1 to L <ol style="list-style-type: none"> a. Wake up on data receive on the queue Q (from A.1) and add a RF data frame at the end of RF-Data-buffer. b. Spawn a thread and call <i>NCC-EI-thread</i> function, input (a pair in RF-Data-buffer, thread_count, L, and thread-handle-array). c. Remove the RF data frame at first position from the RF-Data-buffer. 4. end for 5. Got to 1. 	<ol style="list-style-type: none"> 6. Choose L highest values from Crr buffer (size k) and store the RF data pairs corresponding to these values into RF-final-buffer-pair (buffer size L). 7. Initialize a thread handle array (thread-handle-array) of size L to handle top L Crr valued RF data thread pairs. 8. for thread_count=1 to L <ol style="list-style-type: none"> a. Spawn a thread and call <i>NCC-EI-thread</i> function, input (RF data frame pair from buffer RF-final-buffer-pair(thread_count), thread_count, L, and thread-handle-array). 9. end for 9. Remove the RF data frame at first position from the RF-Data-buffer. 10. Go to 3.
<p><i>C. NCC-EI-thread:</i></p>	<p><i>E. NCC-GPU-function:</i></p>
<ol style="list-style-type: none"> 1. Call the NCC-GPU-function to generate an EI frame. 2. Scan-convert the EI frames to adjust the dimension to the depth of scan and width of the probe. 3. If this is the L^{th} thread then <ol style="list-style-type: none"> a. Wait for (L-1) threads to finish <ol style="list-style-type: none"> for i = 1 to L <ol style="list-style-type: none"> thread-handle-array(i).join(); end for b. Write the data of this thread L^{th} thread to a shared data space. c. Perform the assigned task of the accumulator thread. d. Send the data over the network using MUSiiTCPSTerver. else <ol style="list-style-type: none"> a. Write the data of this thread_count to the shared data space. 	<ol style="list-style-type: none"> 1. Initialize CUDA stream. 2. Initialize data on CPU memory (receive from GPU) and a corresponding data on GPU memory (receive from CPU). Attach this GPU buffer to CUDA stream created in 2. 3. Transfer the data from CPU memory to GPU memory. This is an asynchronous transfer using GPU streams. 4. Execute the kernels specified in Fig. 3 on GPU attached to the CUDA stream. 5. Receive the output data to CPU from GPU asynchronously.

Figure 4.2 Algorithm of multi-stream GPU elastography and O-TRuE: The multi-stream GPU elastography algorithm is described on the left and the corresponding O-TRuE, which reuses several components of the multi-stream GPU elastography is on the right.

4.3.3 Multi-Stream GPU-based Elastography

Figure 4.2 details the multi-stream GPU-based elastography algorithm. An elastography thread is a collection of normalized cross-correlation (NCC) based elastography algorithm modules as shown in Fig. 4.3. First, a displacement map is calculated between two RF images on GPU using NCC (eq. 1); this data is filtered using a moving average and a median filter to remove outliers from the displacement map; then strain estimation is performed using least squares fitting, followed by scan conversion. This is an extension of the work in [72] with all of these modules executing on the GPU. When multiple threads are invoked, a mechanism is needed to ensure data integrity and to prevent threads from simultaneous access to the shared data. Instead of implementing a complex mechanism of synchronizing data using indexing techniques and monitoring resource allocations, these modules are held together by a CUDA stream, which ensures data integrity within a set of CUDA kernels. Modern NVidia GPU architectures, such as Fermi, allow multiple CUDA kernels to execute in parallel with concurrent IO operations between the GPU and CPU. This high-level parallelism enables optimal utilization of GPU resources.

In our real-time ultrasound elastography system shown in Fig. 4.4, RF data is sent from the ultrasound machine to the elastography server. This data is then passed into a queue where the RF data is distributed over different elastography threads, each accepting a pair of RF data. Queuing mechanism helps receiver and the processing threads to work independently. The processing threads simply go to sleep when no data is available. If a data receiver thread receives data then it simply invokes a wakeup call to these threads. The Boost thread library is used for thread synchronization. The elastography threads dispatch their RF data

to the GPU for elastography computation and then send the output elastography data to the output data queue of the *MUSiiCTCPServer* running an independent thread. The *MUSiiCTCPServer* may have several clients connected to it, which are typically visualizers for viewing the elastography data. To adapt this system to other usage, the n^{th} thread can simply collect $n-1$ threads data to perform aggregate operations as averaging or weighted averaging of selective elastography images.

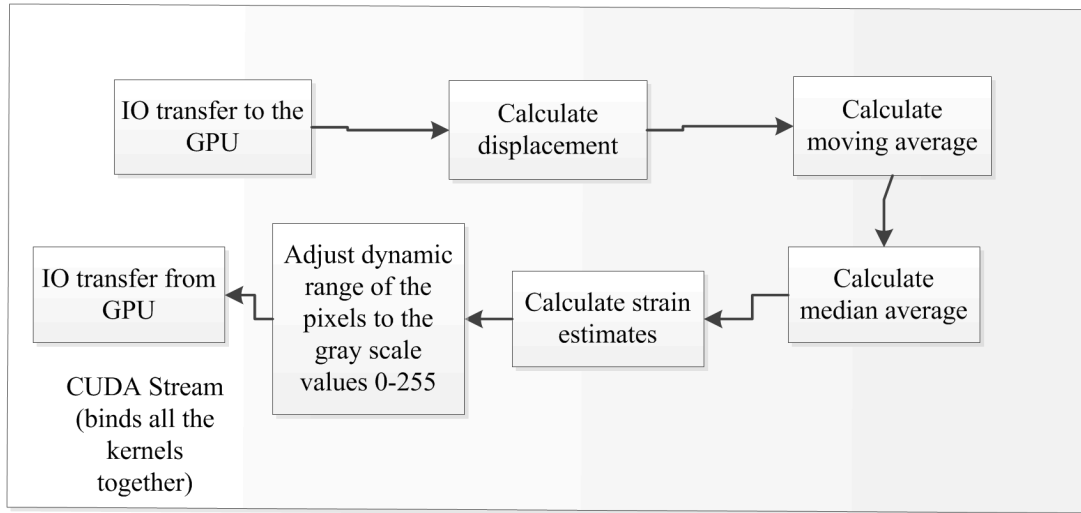


Figure 4.3 Elastography stream pipeline: Figure shows contents of the elastography image stream. These are collection of kernel calls in CUDA necessary to generate elastography images [89]. Since these streams support data integrity, they can be plugged into distinct threads.

4.3.4 Online Tracked Ultrasound Elastography

In Online Tracked Ultrasound Elastography (O-TRuE), a buffer of n RF data frames is analyzed; the C_{rr} value from eq. (10) is calculated for these RF data frames by extracting the tracking data embedded in each RF frame. To find well-aligned RF pairs, the C_{rr} is computed for all $\binom{N}{2}$ combinations of RF data frames and the top m matches are chosen

to compute an elastography image. The algorithm is detailed in Fig. 4.2.

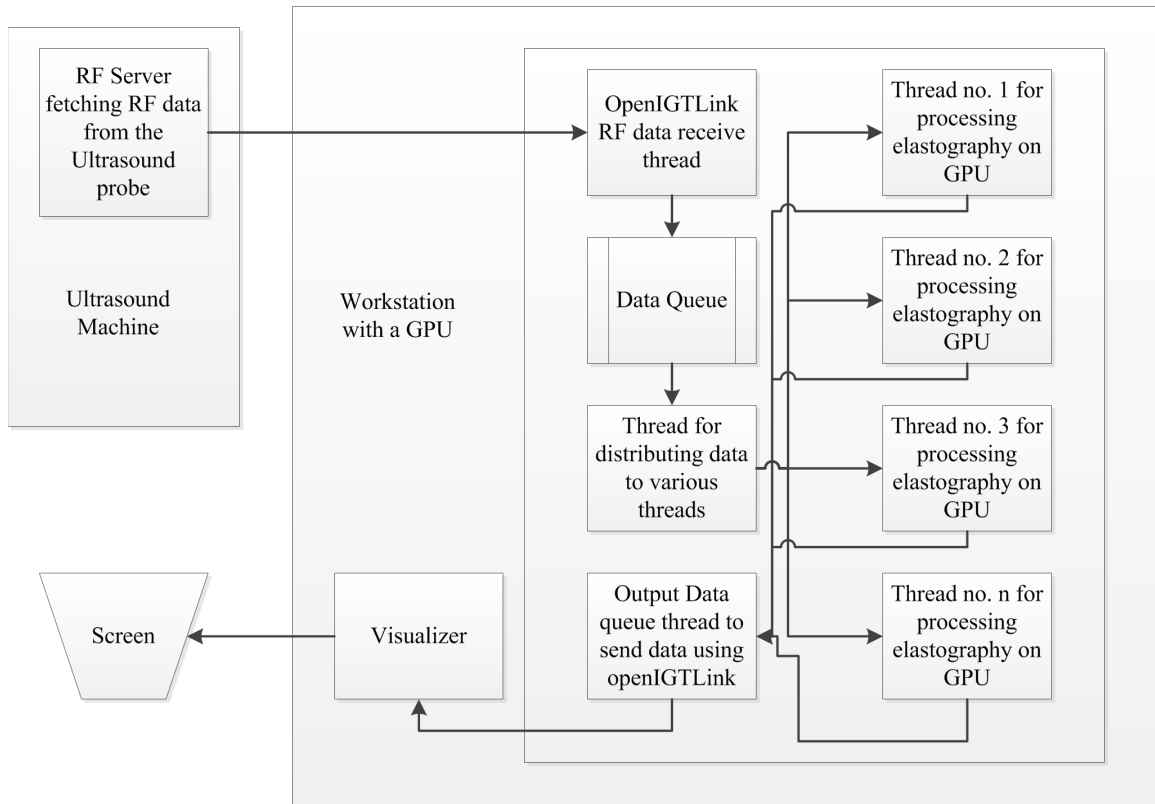


Figure 4.4 Elastography Server: This figure shows real-time pipeline where data is acquired through a radio-frequency (RF) server which runs on a US machine. As can be seen, a combination of queue and threading mechanism is implemented to connect all the components efficiently. Queuing mechanism allows the receiver and processing threads to work independently. The processing threads sleep if there is no data available to process and are triggered by data receiving component whenever data is ready. Elastography threads are the multiple threads that are spawned per consecutive or selected pair of RF data received. Every thread can send out the data over the network using IGTLMessages. The n^{th} thread can collect data from all the other $n-1$ threads to perform aggregate operations as averaging or weighted averaging of selective elastography images.

4.3.5 O-TRuE Image Fusion

Image fusion of multiple elastography images may be used to compensate for global deformation, as well as improve SNR and CNR. By applying a weighted fusion, less weight may be given to the noisier images in each fusion [10]. We investigate applying this

technique to each set of m best matches chosen by O-TRuE as described in subsection *Online Tracked Ultrasound Elastography*. A fused image I_F may be defined as

$$\forall x, y : I_F(x, y) = \frac{\sum_{i=1}^m \alpha_i I_i(x, y)}{\sum_{i=1}^m \alpha_i} \quad (11)$$

where I_i are the m images being fused and α is an image weighting factor equal to the average of the correlation map generated by NCC (Subsection *Normalized Cross-Correlation (NCC) based Elastography within Background* section). Figure 4.5 shows the flow of O-TRuE image fusion where the top m strain images are fused together in real-time by weighted averaging. We implement the image fusion operation by customizing one elastogram thread as an accumulator thread. Once the other $m-1$ threads have finished calculating their elastogram images, they store these images in a shared buffer which is then accessed by the accumulator thread to compute the fused image.

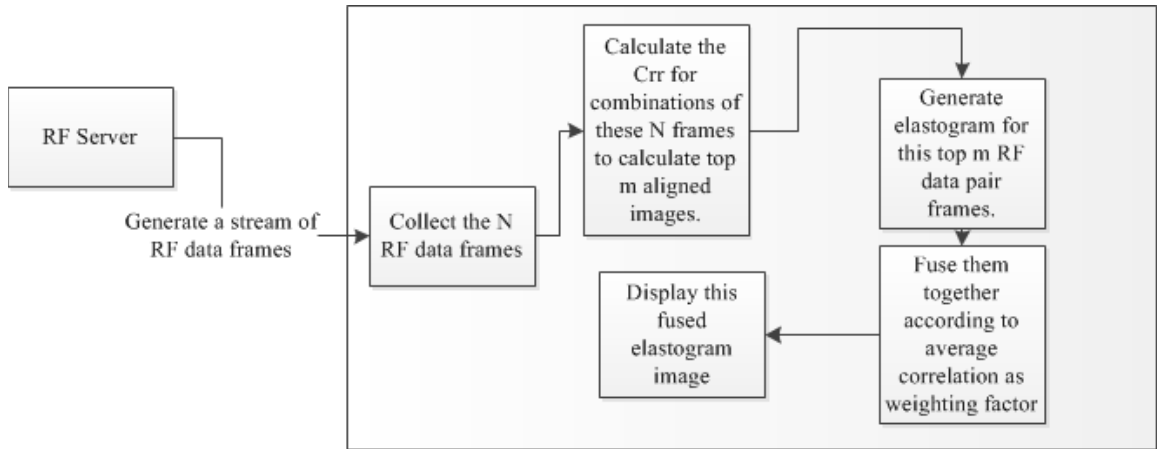


Figure 4.5 Real-time Online tracked Ultrasound Elastography (O-TRuE): Figure shows the real-time online tracked US elastography (O-TRuE) where the cost function is calculated from combinations of the tracked RF data. Then the elastography images are computed for the top m RF data pairs according to the Crr values. The elastography images can then be fused together by simply averaging the images or by weighted averaging based on average correlation values of each elastography image.

Pseudocode 4.1 RFDataReceiver: This module receives the RF data from the RF server or from a file and transmits the data to the EI processing engine via a queueing mechanism. This module also receives the used RF data images to be freed in the memory.

Function RFDataReceiver

Input: RF data from the RF server in OpenIGTLinkMUSiiC format. In the case of online tracked ultrasound elastography, this function receives tracking information embedded inside the RF data OpenIGTLinkMUSiiC packet.

Output: RF data in a local structure format that is easily usable by different EI modules. In the case of online tracked ultrasound elastography, this function sends tracking information for each RF data packet.

```
i := 0
While(true)
    Sleep on OpenIGTLinkMUSiiC receiver for data over the network
    Receive RF(i) data image from OpenIGTLinkMUSiiC SmartPointer
    Allocate GPU CUDA direct memory access (DMA) memory M(i)
    Transfer RF(i) to M(i)
    Insert RF(i) into RFInputQueue(x) - Send Wakeup to listening processes
    Test if RFOutputQueue(y) has any data returning from EI server
    (Here we do not sleep to avoid race condition)
    If (y == dataPresent)
        Free (y) with special CUDA call
        Remove reference to OpenIGTLinkMUSiiC SmartPointer
    end
    i := i + 1
LoopbackToWhile
```

EndFunction RFDataReceiver

Pseudocode 4.2 EIProcessor: This module receives the RF data from the RFDataReceiver and processes the data to compute elastography images and transmit the images over the network. This module is the basic version without the CPU threading model.

Function EIProcessor

Input: RF data from the RFDataReceiver in simple local structure format.

Output: Selected RF data image pair (elastography data outputted via EIComputation). RF data sent back to RFDataReceiver for recycling.

```
i := 0
j := 0
Create a threadArray that acts a Queue (maxSize: N)
While (true)
    Sleep on RFInputQueue(x) to receive RF(i)
    When RFInputQueue receives data - Wakeup
    RF (i) = x
    i := i + 1
    Allocate EI_parameters_structure for GPU computation
    (This structure contains all parameters necessary for EI computation)
    EIComputation (RF(i-1), RF(i), EI_parameters_structure, j, 'nonThreaded')
    Send the RF data from this queue to RFOutputQueue(y)
    j := mod((j + 1), N);
LoopbackToWhile
```

EndFunction EIProcessor

Pseudocode 4.3 EIProcessorAdvanced: This module receives the RF data from the RFDataReceiver and processes the data to compute elastography images and transmit the images over the network. This module is the advanced threaded version with the CPU threading model.

<p>Function EIProcessorAdvanced</p> <p>Input: RF data from the RFDataReceiver in simple local structure format. Output: Selected RF data image pair (Elastography data outputted via EIComputation). RF data sent back to RFDataReceiver for recycling.</p> <pre> i := 0 j := 0 Create a threadArray that acts a Queue (maxSize: N) While (true) Sleep on RFInputQueue(x) to receive RF(i) When RFInputQueue receives data - Wakeup RF (i) = x i := i + 1 Allocate EI_parameters_structure for GPU computation (This structure contains all parameters necessary for EI computation) threadArray(j) = EIComputation (RF(i-1), RF(i), EI_parameters_structure, j, 'Threaded') for k = 0 to N Test if threadArray(k) exited if threadArray(k) exited then free the memory and reuse Send the RF data from this queue to RFOutputQueue(y) endFor j := mod((j + 1), N); LoopbackToWhile </pre>
<p>EndFunction EIProcessorAdvanced</p>

Pseudocode 4.4 EIProcessorOTruE: This module receives the RF data from RFDataReceiver that has a tracking information embedded from a tracking device and combined by the MusiiCSync program. This module calculates a permutation of all images for a buffer size M and selects the top X images to be sent on the network.

Function EIProcessorOTruE

Input: a) RF data from RFDataReceiver with tracking information in a simple data structure. b) Buffer size M where $M*(M-1)/2$ comparisons are made. c) Value of X, that are the top images to be selected.

Output: Selected RF data image pair and tracking information (Elastography data outputted via EIComputation). RF data sent back to RFDataReceiver for recycling.

```

i := 0
j := 0
Create a double array threadArray that acts as a Queue (maxSize: N x X)
While (true)
    Sleep on RFInputQueue(x) to receive RF(i)
    When RFInputQueue receives data - Wakeup
    Loop until M RF image buffers are collected
        RF (i) = x
        i := i + 1
    endLoop
    Rfdatapairs = CalculateTopXTruE (RF, i-M, i, M, X)
    m := 0;
    foreach Rfpair in Rfdatapairs
        (Here Rfpair is a structure that contains an RF pair)
        Allocate EI_parameters_structure for GPU computation
        (This structure contains all parameters necessary for EI computation)
        threadArray(j, m) = EIComputation (Rfpair(0), Rfpair(1), EI_parameters_structure, j)
        m := m + 1
    endfor
    for k = 0 to N
        for n = 0 to X
            Test if threadArray(k, n) exited
            if threadArray(k, n) exited then free the memory and reuse
            Send the RF data from this queue to RFOutputQueue(y)
        endFor n
    endFor k
    j := mod((j + 1), N);
    LoopbackToWhile

```

EndFunction EIProcessorOTruE

Pseudocode 4.5 EIProcessorOTruEAveraging: This module receives the RF data from RFDataReceiver that has tracking information embedded from a tracking device and combined by the MusiiCSync program. This module calculates a permutation of all images for a buffer size M and selects the top X images to calculate a weighted average based on the average correlation of each image and send the data on the network.

Function EIProcessorOTruEAveraging

Input: a) RF data from RFDataReceiver with tracking information in a simple data structure. b) Buffer size M where $M*(M-1)/2$ comparisons are made. c) Value of X that are top images to be selected.

Output: Weighted average elastography images for the selected RF data image pair and tracking information for one of the input RF images (via EIComputationAveraging).

```

i := 0
j := 0
Create a double array threadArray that acts as a Queue (maxSize: N x X)
While (true)
    Sleep on RFInputQueue(x) to receive RF(i)
    When RFInputQueue receives data - Wakeup
    Loop until M RF image buffers are collected
        RF (i) = x
        i := i + 1
    endLoop
    RFdatapairs = CalculateTopXTruE (RF, i-M, i, M, X)
    m := 0;
    foreach RFpair in RFdatapairs
        (Here, RFpair is a structure that contains an RF pair
        Allocate EI_parameters_structure for GPU computation
        (This structure contains all parameters necessary for EI computation)
        threadArray(j, m) = EIComputationAveraging (RFpair(0), RFpair(1),
                                                    EI_parameters_structure, j, m, X)

        m := m + 1
    endfor
    for k = 0 to N
        for n = 0 to X
            Test if threadArray(k, n) exited
            if threadArray(k, n) exited then free the memory and reuse
            Send the RF data from this queue to RFOutputQueue(y)
        endFor
        j := mod((j + 1), N);
    LoopbackToWhile

```

EndFunction EIProcessorOTruEAveraging

Pseudocode 4.6 EIComputationAveraging: This module computes the average of the top X ranked elastography images. This module computes the average based on the weighted average of the average correlation values. The last Xth thread computes the average. Other threads simply deposit their results in a common location.

EIComputationAveraging

Input: a) RF data from RFDataReceiver with tracking information in a simple data structure. b) Value of X that are top images to be averaged

Output: Weighted averaged elastography images for the selected X top RF data image pairs and the RF data pairs.

```

Receive RF(n) and RF(n+1)
If (modulus(threadID, X) == 0)
    First thread,
        hence allocate memory for averagingStructure (size: X)
endIf
(correlationMap, displacementMap) := ComputeEI (RF(n), RF(n+1))
if (average(correlationMap) < 0)
    (correlationMap, displacementMap) := ComputeEI (RF(n+1), RF(n))
endIf
displacementMapRefined = movingAverage(medianFiltering(displacementMap))
strainMap := strainEstimation (displacementMapRefined)

if (modulus(threadID, X) != X-1)
    store strainMap in averagingStructure (threadID % X);
else
    sumAverageCorrelation := 0
    memset(finalAverageStrain, 0)
    (now calculate weighted average)
    for i := 0 to X
        averageCorrelation(i) := average (correlationMap(i))
        sumAverageCorrelation := sumAverageCorrelation + averageCorrelation(i)
    endFor
    for i := 0 to X
        finalAverageStrain := averageCorrelation(i) *
strainMap(i)/sumAverageCorrelation;
    endFor
endIf

if (clientConnected)
    send finalAverageStrain on network
endIf
return RF(n) and RF(n+1)

```

EndFunction EIComputationAveraging

Pseudocode 4.7 CalculateTopXTruE: This module returns the Top X RF data image pairs that are in close proximity to obtain a good quality elastography output.

<p>CalculateTopXTruE Input: Tracking information of M RF data images, value of M, and value of X Output: Top X RF data image pairs and corresponding correlation values</p>
<pre> i := 0 j := 0 topXTruECorrelation stores indexes and corresponding TrueCorrelation values for i := 0 to M for j := 0 to M * (M - 1)/2 topXTruECorrelation (i, j).correlation := calculateCorrelation(RF(i).trackingMatrix() RF(j).trackingMatrix()); topXTruECorrelation (i, j).index := (i, j); endFor j endFor i sort in descending order topXTruECorrelation based on index values return topXTruECorrelation structure for (0 to X-1) indexes as an array of RFpair (We call the array of RFpair as the RFdatapairs) </pre>
<p>EndFunction CalculateTopXTruE</p>

Pseudocode 4.8 EIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography.

<p>EIComputation Input: RF data to be processed. Here n means any value greater than or equal to 0. n and n+1 simply indicate two different RF data images. Output: Generated strain image sent over the network and the RF data pairs</p>
<pre> if 'Threaded' then this function runs on a separate thread Receive RF(n) and RF(n+1) (correlationMap, displacementMap) = ComputeEI (RF(n), RF(n+1)) if (average(correlationMap) < 0) (correlationMap, displacementMap) = ComputeEI (RF(n+1), RF(n)) endif displacementMapRefined := movingAverage(medianFiltering(displacementMap)) strainMap := strainEstimation (displacementMapRefined) if (clientConnected) send strainMap on network endif Free (strainMap, displacementMap, displacementMapRefined, correlationMap); return RF(n) and RF(n+1) </pre>
<p>EndFunction EIComputation</p>

4.4 Experiments

This section details the experiments performed to show the effectiveness of the multi-stream elastography algorithm, stability of O-TRuE and stability achieved with da Vinci surgical system. Two ultrasound machines are used for the experiments, which are Sonix RP (Ultrasonix Co.) for phantom and da Vinci surgical system experiments, and Sonix CEP (Ultrasonix Co.) for in-vivo animal experiments. A high-performance Tesla C2070 GPU card is used for elastography computations. The machine that is running elastography computation has 12 GB of RAM and a 2.13 GHz Intel Xeon processor.

4.4.1 Phantom Experiments

In phantom experiments, a CIRS Elasticity QA Phantom Model 049, which has background elasticity of 33 kPa and lesions with 7, 15, 39, and 58 kPa elasticity, is used. The purpose of this experiments is to determine speed increment achieved by multi-stream GPU approach towards elastography, as compared to single stream and no stream GPU approach. We would like to determine whether the performance of the system in speed and quality remains stable over time on the given phantom. These results are important to establish multi-stream GPU elastography as enabling method for O-TRuE and integration with da Vinci surgical systems. In relation to this, we want to see whether O-TRuE supported freehand palpation gives any benefits over untracked freehand palpation in terms of stable correlation of consecutive images generated by both system.

The experiments are performed on the 58 kPa lesion, measuring 2 cm in diameter. The flat upper surface of this phantom helps to ensure that in-plane RF data frame detection by O-TRuE gives a very high quality elastogram. The O-TRuE algorithm consistency is measured by applying the same RF data stream as an input to both tracked and untracked version of elastography. This ensures that consistent data is used to compare the two methods. The results are saved to the disk as OpenIGTLink message files. These message files can be later retrieved for further programmatic analysis. The output elastography image is measured for consistency by measuring the correlation value of consecutive elastogram generated from O-TRuE and Untracked elastography. To compare actual sequential elastography image generation and the ones selected by O-TRuE, we save the elastography image for all permutations of the given buffer of RF data frames. We also save the information of the ranking of the elastography image frames and corresponding C_{rr} values. The elastography frames are arranged in grid form to showcase the effectiveness of O-TRuE selection. Fusion data is also generated by combining the top m frames and evaluated for varying values of m . Typical values for m are 1, 3 or 5. The same RF data is used to compare different values of m to enable direct comparison of the results.

Multi-stream elastography algorithm speed is measured by inserting timers just before the first thread of elastography calculation is fired called t_0 and after N^{th} elastography thread completion. All the thread handles are collected in a dynamic array of size N and passed onto a separate thread along with value of t_0 . This thread waits for all N thread handles to indicate thread execution completion before measuring the time t_N ; this helps to avoid

delay in logging the data and prevents the elastography processing from being impacted. The time to calculate one elastography image frame is given by

$$\Delta t = (t_N - t_0) / N . \quad (12)$$

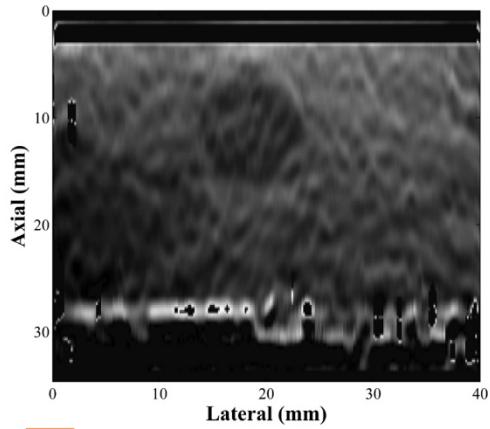
4.4.2 In-Vivo Animal Experiments

The phantom experiments provide a baseline for comparison of O-TRuE and untracked elastography. The animal experiments replicate various conditions that phantom experiments cannot demonstrate. Palpation motion is not necessarily parallel to the axial motion of the probe, and the organ surface is slippery due to blood or US gel. We want to determine whether the real-time elastography implementation, due to its high speed, compensates for the small lateral and elevational motion to give a good elastography image. A few regions of the pig liver were ablated in-vivo using RITA ablator and the ablated region was visualized using real-time untracked elastography [58]. The pig was euthanized and the liver was extracted for gross pathology study [58]. Data collection was performed by connecting a listener to the RF server. This listener saved data from the RF server in OpenIGTLink message files. The files were saved with filenames containing timestamps or sequence numbers to aid in re-playing the data at a later date. The saved RF data files were read by the untracked elastography server in the same order as they were generated. The experimental data was collected during offline TRuE evaluation as described in [58]. The depth of acquisition is 3 cm. A trend of NCC window size vs. speed of elastography image generation and image quality is conducted on the output from this

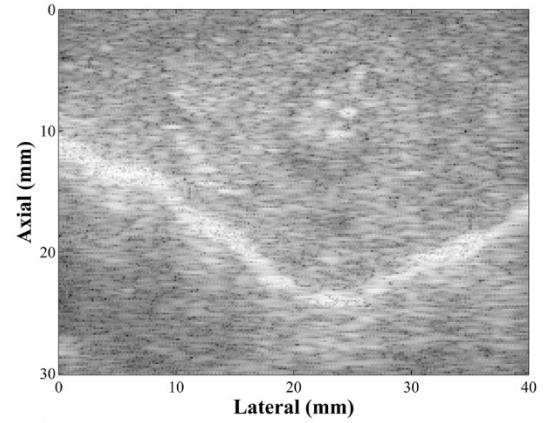
in-vivo data (Table 4.3). The SNR is calculated for the entire image, and CNR is calculated for target and background image region of (30x30 pixel square).

4.4.3 da Vinci Surgical System Experiments

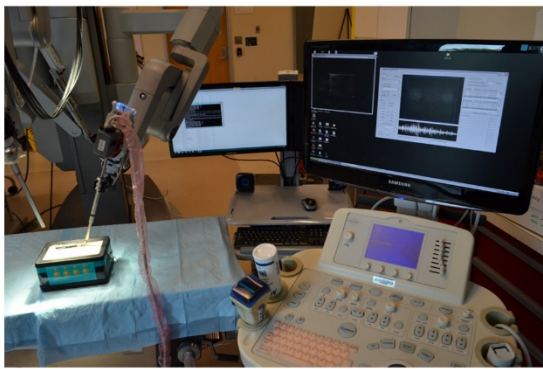
Under some constraints O-TRuE is hard to implement using EM tracker due to presence of ferromagnetic materials in the surrounding area. One such case is da Vinci surgical robot where the US probe is mounted on one of the arms of the robot [82]. We demonstrate the feasibility of integration of elastography with a da Vinci surgical robot where controlled palpation motions are performed. We want to determine what type of palpation motion can give a steady elastography image stream and if high speed elastography has any advantage. As shown in Fig. 4.6, the da Vinci surgical robot is connected with the US machine using OpenIGLTLINKMUSIC where the B-mode and its corresponding elastography images are sent over the network. The console of the da Vinci surgical robot has live frames of elastography and B-mode overlaid within surgeon's field-of-view; the enabled status, position and size of these frames can be adjusted using the master manipulator of the robot to provide the surgeon full control. Palpation motion is generated by the robotic arm of the da Vinci using the da Vinci research API to autonomously control the frequency, amplitude, and direction of robot motion [82].



A



B



C

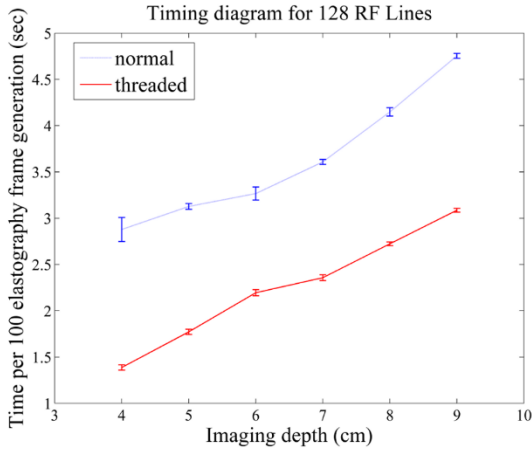


D

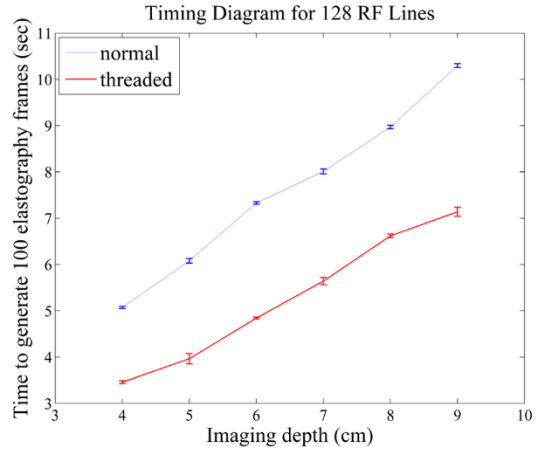
Figure 4.6 Integration with da Vinci surgical systems: Untracked elastography has been integrated with da Vinci surgical systems using a laparoscopic probe controlled by an arm of the da Vinci surgical robot. (C) Shows the overall setup. (D) Shows the view from surgeon's console of how B-mode (B) and Elastography image (A) appear when overlaid in the console display.

Table 4.1 Test results for comparing frame rate performance of multi-stream GPU elastography (threaded) with single-stream (streamed) and non-stream (normal) GPU elastography. This table reports average frames per second (with standard deviation in brackets) of images generated by various versions of the elastography program. The term *normal-N* indicates the basic GPU implementation of NCC elastography, *streamed-N* indicates the streamed GPU implementation, and *threaded-N* indicates the multi-streamed GPU implementation, where *N* indicates the number of RF lines in each RF image. Four test cases were performed at different NCC window sizes, NCC maximum search distances (displacements), and NCC search step sizes (specified as percentage of window overlap). The computational load increases with larger window size, displacement, and percent overlap. As seen in the results, the highest speed obtained is 78 frames per second (fps) running the multi-streamed GPU implementation.

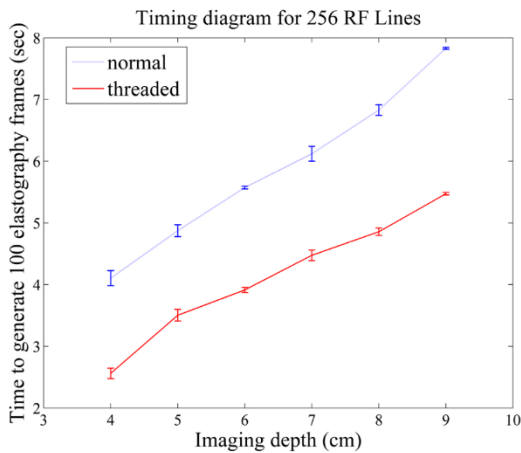
		Depth in cm	4	5	6	7	8	9
		Depth in pixels	1024	1296	1552	1808	2064	2336
Case Window size (pixel) Displacement (mm)	1	normal-128	36.56(±1.75)	31.10(±0.61)	31.56(±0.29)	29.10(±0.39)	25.20(±0.20)	22.39(±0.28)
	10	threaded-128	78.02(±0.92)	61.14(±0.53)	48.40(±0.87)	46.19(±0.48)	39.36(±0.58)	34.33(±0.34)
	2	streamed-128	36.31(±2.22)	29.01(±6.02)	30.76(±0.97)	29.41(±0.34)	25.19(±0.47)	22.24(±0.11)
Overlap (%)	98	normal-256	26.10(±0.16)	22.72(±0.41)	19.08(±0.06)	17.80(±0.07)	15.66(±0.32)	13.89(±0.03)
		threaded-256	42.55(±0.96)	31.48(±0.41)	26.78(±0.63)	24.23(±0.33)	21.78(±0.36)	19.60(±0.09)
		streamed-256	26.00(±2.49)	22.60(±0.38)	19.13(±0.09)	17.77(±0.19)	15.27(±1.38)	13.75(±0.21)
Case	2	normal-128	34.74(±1.65)	31.98(±0.32)	30.62(±0.68)	27.71(±0.21)	24.11(±0.26)	21.03(±0.12)
Window size (pixel)	12	threaded-128	72.07(±1.54)	56.43(±0.89)	45.55(±0.70)	42.42(±0.57)	36.72(±0.27)	32.41(±0.22)
Displacement (mm)	2	streamed-128	34.68(±3.02)	31.91(±0.50)	30.56(±1.15)	28.29(±0.79)	24.49(±0.46)	21.20(±0.22)
Overlap (%)	98	normal-256	24.36(±0.75)	20.52(±0.41)	17.94(±0.07)	16.34(±0.33)	14.64(±0.19)	12.77(±0.03)
		threaded-256	39.04(±1.34)	28.54(±0.80)	25.55(±0.27)	22.35(±0.44)	20.59(±0.26)	18.27(±0.07)
		streamed-256	24.16(±2.20)	21.21(±1.57)	18.08(±0.43)	16.53(±0.33)	14.82(±0.05)	12.78(±0.07)
Case Window size (pixel) Displacement (mm)	3	normal-128	21.53(±0.13)	21.54(±0.12)	19.73(±0.08)	19.19(±0.03)	16.98(±0.12)	14.24(±0.04)
	14	threaded-128	46.43(±0.44)	35.44(±0.42)	29.53(±0.29)	27.05(±0.20)	24.46(±0.35)	21.30(±0.07)
	4	streamed-128	21.76(±0.13)	21.60(±0.13)	19.84(±0.17)	19.33(±0.11)	16.63(±0.25)	14.35(±0.09)
Overlap (%)	98	normal-256	15.74(±0.07)	13.67(±0.03)	11.56(±0.15)	10.69(±0.03)	9.67(±0.04)	8.12(±0.02)
		threaded-256	25.89(±0.19)	19.53(±0.15)	16.62(±0.12)	14.47(±0.21)	13.37(±0.12)	11.67(±0.06)
		streamed-256	15.83(±0.09)	13.68(±0.13)	11.50(±0.04)	10.68(±0.06)	9.68(±0.02)	8.12(±0.01)
Case	4	normal-128	19.71(±0.09)	16.45(±0.13)	13.65(±0.05)	12.49(±0.08)	11.15(±0.05)	9.71(±0.04)
Window size (pixel)	16	threaded-128	28.93(±0.24)	25.23(±0.68)	20.65(±0.08)	17.73(±0.24)	15.11(±0.09)	14.01(±0.19)
Displacement (mm)	4	streamed-128	19.81(±0.10)	16.24(±0.50)	13.66(±0.19)	12.48(±0.18)	11.12(±0.13)	9.69(±0.04)
Overlap (%)	99	normal-256	11.64(±0.02)	9.17(±0.02)	7.64(±0.02)	6.82(±0.06)	6.08(±0.02)	5.20(±0.01)
		threaded-256	15.70(±0.07)	13.02(±0.07)	10.21(±0.11)	9.00(±0.13)	8.10(±0.02)	6.64(±0.04)
		streamed-256	11.57(±0.31)	9.20(±0.06)	7.65(±0.05)	6.80(±0.08)	6.02(±0.11)	5.19(±0.02)



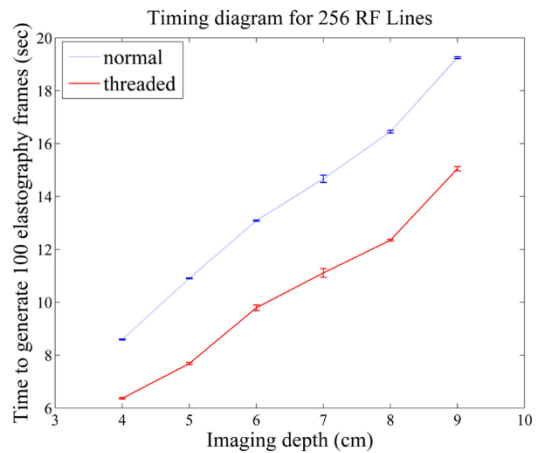
A



B



C



D

Figure 4.7 Timing graph to show speed comparison of multi-stream elastography (threaded) and non-stream elastography (normal): The graphs indicates run times and standard deviation of run time for window size 12, displacement 2 mm, overlap 98% (A, B) and Window size 16, displacement 4 mm, overlap 99% (C, D). The results are per 100 frames. The standard deviation is max 0.13 for Fig. (A), 0.122 for Fig. (B), 0.136 for Fig. (C), 0.167 for Fig. (D), which is very small for 100 frames. This graph also shows that the increased window size reduces the performance of the algorithm due to higher serial search within the large windows.

4.5 Results

4.5.1 Speed Analysis

Results of a run-time performance comparison of multi-stream GPU elastography with single-stream and non-stream GPU elastography is provided in Table 4.1. Frame rate averages and standard deviations are computed for each GPU implementation under different test cases of varying number of RF lines, NCC window size, NCC maximum search distance, and NCC search step size. As seen in Table 4.1, the runtime differences between non-stream and single-stream are negligible. This indicates low overhead in our implementation of streamed data processing. For the multi-stream implementation, the runtime speedup is very significant and in some cases more than double the frame rate of the other implementations. In some cases, the multi-stream implementation with 256 RF lines is even faster than the normal and single-stream implementations with 128 RF lines. This indicates that our multi-streamed implementation provides higher utilization of the GPU with greater runtime efficiency, even though the multi-stream implementation is controlled from multiple CPU threads. As seen in the Table 4.1, the highest speed achieved is 78 fps while running the multi-stream implementation with 128 RF lines. This is a significant improvement given that the corresponding rate of elasticity image generation nearly matches the image acquisition speed of the RF server as presented in [85].

Figure 4.7 provides a graph of inverse results of a subset of Table 4.1, which shows the average generation time in seconds for 100 elastography frames estimated over 20 trials. Figure 4.7 compares non-stream and multi-stream GPU implemenations, making clear that

multi-stream outperforms non-stream. The bars in the figure indicate the standard deviation of runtime among the 20 trials. A stable runtime is important to ensure fast system response over all periods in time. Figure 4.7 shows that the standard deviation for both GPU implementations is stable, the standard deviation is max 0.13 for Fig. 4.7-A, 0.122 for Fig. 4.7-B, 0.136 for Fig. 4.7-C, 0.167 for Fig. 4.7-D. A worst case standard deviation of 0.167 seconds to generate 100 elastography frames (Fig. 4.7-D) indicates a stable runtime.

pair	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10
actual image									
rank, Crr	7, 0.910	21, 0.307	39, 0.000	38, 0.000	32, 0.003	17, 0.739	4, 0.946	6, 0.931	9, 0.893
CNR, SNR	0.68, 2.53	0.22, 0.89	0.09, 0.82	0.32, 1.34	0.04, 0.59	0.7, 1.33	1.3, 3.26	1.27, 3.38	0.51, 1.97
pair	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	3,4
actual image									
rank, Crr	20, 0.457	36, 0.000	33, 0.001	30, 0.015	10, 0.833	3, 0.964	2, 0.972	1, 0.979	19, 0.515
CNR, SNR	0.36, 2.1	0.24, 1.41	0.27, 0.91	0.91, 2.16	0.98, 3.12	0.9, 2.5	0.84, 2.37	0.78, 2.99	0.11, 1.06
pair	3,5	3,6	3,7	3,8	3,9	3,10	4,5	4,6	4,7
actual image									
rank, Crr	18, 0.555	16, 0.747	13, 0.785	26, 0.189	24, 0.222	23, 0.246	12, 0.791	8, 0.910	29, 0.027
CNR, SNR	0.63, 2.3	0.68, 2.49	0.38, 1.47	0.13, 0.82	0.03, 0.77	0.3, 1.48	0.63, 2.85	0.55, 2.5	0.37, 1.29
pair	4,8	4,9	4,10	5,6	5,7	5,8	5,9	5,10	6,7
actual image									
rank, Crr	45, 0.000	44, 0.000	43, 0.000	5, 0.943	28, 0.048	42, 0.000	41, 0.000	40, 0.000	25, 0.206
CNR, SNR	0.05, 0.81	0.06, 0.74	0.26, 1.29	0.65, 2.65	0.69, 2.28	0.53, 1.79	0.34, 1.44	0.36, 1.49	0.68, 2.15
pair	6,8	6,9	6,10	7,8	7,9	7,10	8,9	8,10	9,10
actual image									
rank, Crr	37, 0.000	35, 0.000	34, 0.001	15, 0.765	14, 0.781	11, 0.828	27, 0.052	22, 0.273	31, 0.009
CNR, SNR	0.67, 2.22	0.6, 1.83	0.12, 0.93	0.73, 2.4	0.38, 1.65	1.21, 2.57	0.21, 3.29	1.19, 3.52	1.01, 3.27

Figure 4.8 Selection map of O-TRuE images: The row above each image sequence indicates the RF data pair index. For e.g. the pair identifier (n1, m1) indicates comparison of radio frequency (RF) data frame acquired at time t_{n1} with that of the frame acquired at time t_{m1} . The pair (image rank, Crr value) below the image sequence indicates the rank and Crr value generated by O-TRuE. The pair (CNR, SNR) indicates contrast-to-noise ratio and signal-to-noise ratio values for each image. O-TRuE selected 90% good elastography images in top 20 ranked images with good CNR and SNR above 0.51 and 2.37 respectively. The Crr above 0.457 is observed to provide with good elastography images.

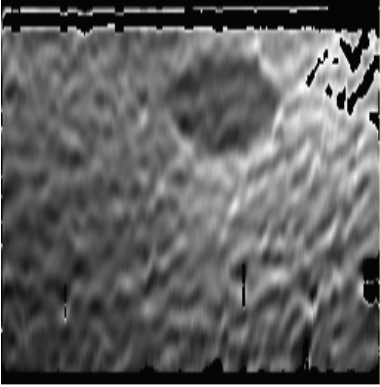
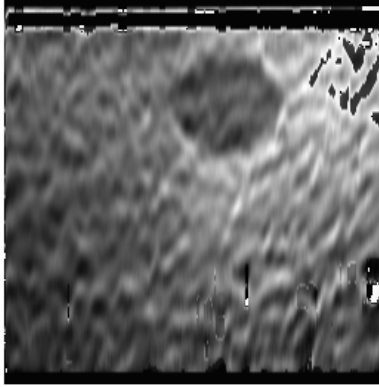
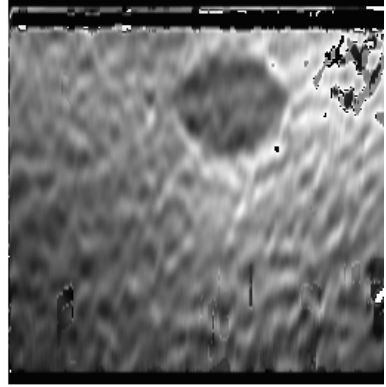
		
a) No Averaging CNR = 1.431 SNR = 2.351	b) Average of 3 images CNR = 1.541 SNR = 2.584	c) Average of 5 images CNR = 1.566 SNR = 2.588
Average (Std. Dev.) CNR 1.226 (0.33) Average (Std. Dev.) SNR 2.031 (0.43)	Average (Std. Dev.) CNR 1.321 (0.24) Average (Std. Dev.) SNR 2.189 (0.35)	Average (Std. Dev.) CNR 1.327 (0.24) Average (Std. Dev.) SNR 2.210 (0.35)

Figure 4.9 Elastography image fusion: The images displayed in (a) is elastography image with single image (best O-TRuE) selection, (b) is elastography image for average of top 3 O-TRuE image selections, and (c) is elastography image for average of top 5 O-TRuE image selections. The results indicates that the fusion by averaging the top 5 elastography images from O-TRuE gives good quality indicated by the average CNR and SNR values of 1.327 and 2.210 respectively.

4.5.2 Validation of O-TRuE Frame Selection

A validation of O-TRuE is performed by computing a C_{rr} pseudo correlation value (eq.

10) and a corresponding elastography image for all possible $\binom{N}{2}$ RF frame pairs in an N

sized buffer with N equal to 10 and σ equal to 1 (see eq. 2 and 7). In a non-validation context, only the frames with highest C_{rr} values would have been chosen for computing corresponding elastography images. Figure 4.8 presents the generated elastography images from this test, which are arranged by order of RF frame acquisition. Visual inspection

reveals that 90% of the top 20 frames chosen by O-TRuE show clear presence of the lesion being imaged. As a quantitative assessment, the CNR and SNR values of each elastography image is calculated and listed as a pair (CNR, SNR) below each image. It is found that top ranking elastograms have either a very good CNR or a very good SNR value, whereas the O-TRuE images of lower rank (i.e. lower Crr) have poorer values of CNR and SNR. For example, the image with rank 26 has a very low CNR of 0.13 and SNR of 0.82. It is observed that the 10 highest ranking O-TRuE images (shown as red text in the figure) all have CNR above 0.51 and SNR above 2.37, which indicates a good elastography result. From these tests, we observe that choosing elastography images with Crr values above 0.457 provides a mostly stable result. There are few anomalies, such as the image with rank 12 having better image quality than the image of rank 9. Such anomalies could be corrected by considering the CNR and SNR values in the ranking system, but at the cost of reduced speed due to the added burden of generating additional elastogram images in order to compute the CNR and SNR values across an extended range of Crr ranked RF frame pairs. In general, these test results show that O-TRuE performs very well in selecting the best RF frame pairs to generate high quality elastograms.

4.5.3 O-TRuE Image Fusion Evaluation

An analysis of the effects of image fusion by averaging is presented in Fig. 4.9 for elastography images generated by O-TRuE. As seen in Fig. 4.9, (a) represents the O-TRuE output for single elastography image of highest Crr value with no averaging, (b) represents the O-TRuE output when averaging the top 3 images of highest Crr value, and (c)

represents the O-TRuE output when averaging the top 5 images. Figure 4.9 shows the CNR and SNR values of each image, as well as average CNR and SNR values evaluated from approximately 220 output images for each case, indicating that fusion by averaging of 5 images provides the best CNR and SNR with average values of 1.327 and 2.210 respectively. This indicates that SNR and CNR performance improves by averaging, but it may fail in cases where some of the top images are noisy as suggested by Fig. 4.8 where images ranked 2 and 3 contain noise in the top right corner of the images. This test indicates that averaging of 5 O-TRuE images is useful for performing image fusion.

4.5.4 Elastography Image Stream Analysis

A consistency analysis is performed for the elastography image streams of both O-TRuE and untracked elastography by applying NCC to a sub-region of the output elastography images as shown in Fig. 4.10-A, where the left-hand image shows the defined size and position of the NCC template window for a given elastography image and the right-hand image shows the defined search region for the target window in the subsequent elastography image. Similar to eq. (1), a correlation map is generated within the target search region and the maximum correlation value is selected from the map as the stream quality measurement

$$\alpha_l = \max_{u,v \in R} \left(\frac{\sum_{x,y} [f_l(x,y) - \bar{f}_l][f_{l+1}(x-u, y-v) - \bar{f}_{l+1}]}{\left\{ \sum_{x,y} [f_l(x,y) - \bar{f}_l]^2 \sum_{x,y} [f_{l+1}(x-u, y-v) - \bar{f}_{l+1}]^2 \right\}^{0.5}} \right) \quad (13)$$

where l is an output elastography image sequence number with value from 1 to the number of RF image pairs minus one, x (lateral) and y (axial) are pixel positions within the template window f_l in image l , $(x-u)$ and $(y-v)$ are pixel positions within a target search window f_{l+1} in image $(l+1)$. Note that \bar{f}_l and \bar{f}_{l+1} are the mean pixel intensities of the f_l and f_{l+1} window regions respectively. The maximum correlation value of the map corresponds to the position of optimal alignment between image features contained by the template and target windows.

Thus, the maximum correlation value is the primary value of interest within the map and is sufficient to serve as an indicator of stability for the image stream.

The region of interest (ROI) in Fig. 4.10-A was manually selected following data acquisition in such a way to contain the lesion being imaged while being large enough to accommodate small displacements of the lesion due to hand motion. This enables running the NCC analysis on a continuous stream of data while ensuring that the lesion remains within the imaged area. A larger window size would risk inclusion of noise along the boundaries of the elastography image, so the window size and position is appropriately defined for the image target in this study. The window size and position is kept constant for both O-TRuE and untracked elastography image streams.

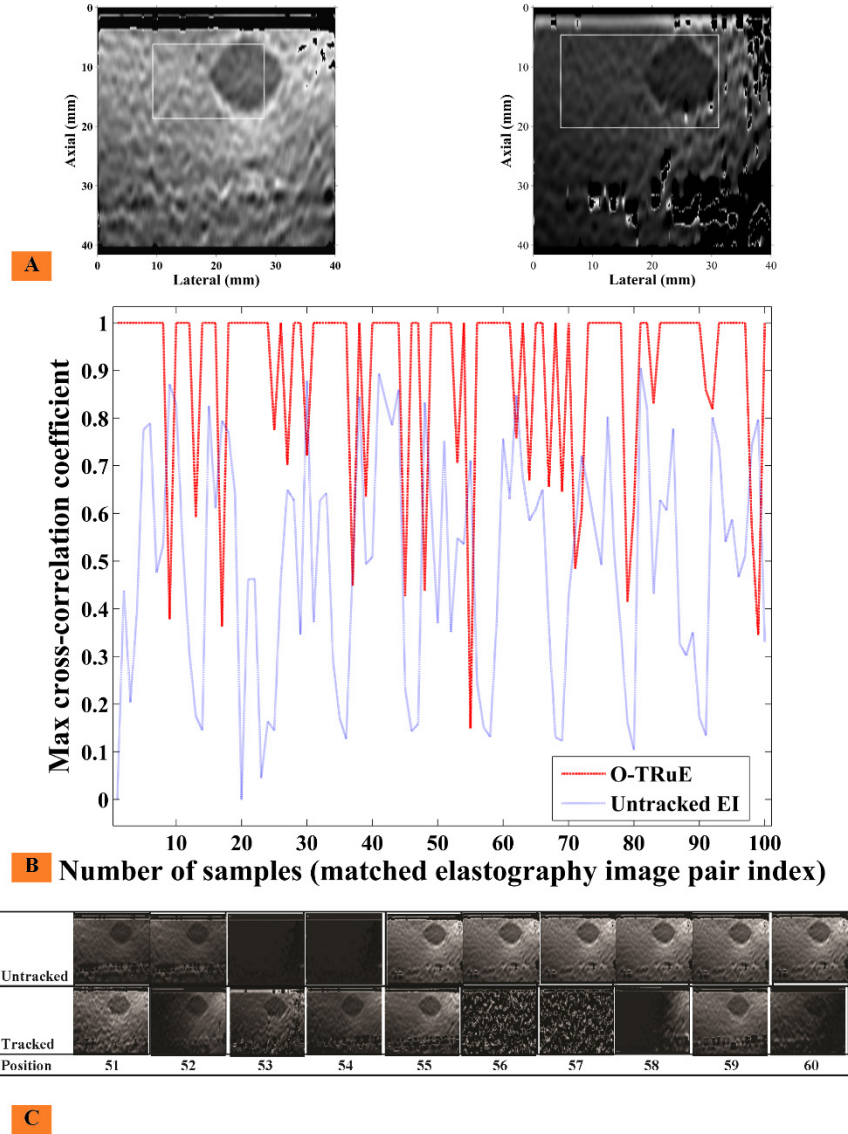
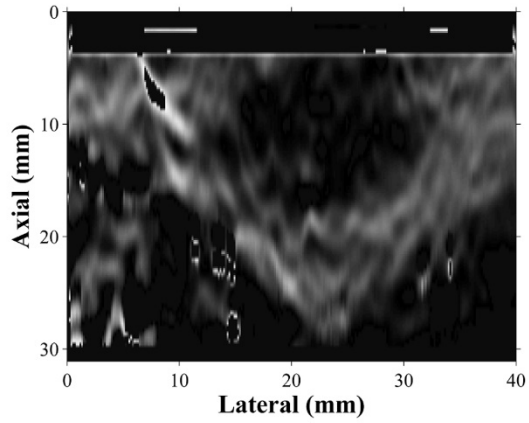
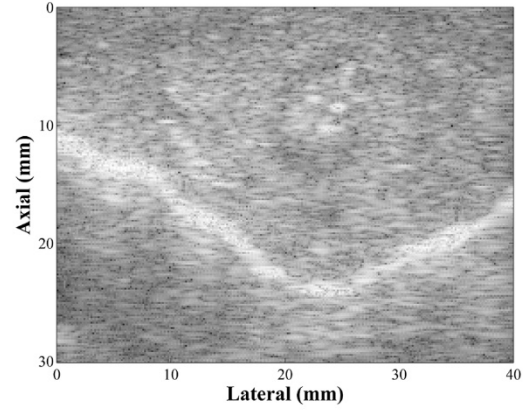


Figure 4.10 Elastography image stream analysis of consecutive frames in O-TRuE and Untracked elastography: An analysis of consecutive frames is done to understand the quality of strain images generated by O-TRuE and untracked elastography. (A) Shows a template region selected in the leftmost image and a target region selected in the rightmost image. We apply normalized cross-correlation in these regions as shown in eq. 13 to find max correlation value. A max correlation graph for 100 elastography image pairs is shown in (B), where the red dashed line is for O-TRuE and a blue dotted line is for untracked elastography. O-TRuE has a more consistent high correlation value across consecutive images. As indicated in Table 4.2, O-TRuE (β values) performs better than untracked elastography. (C) Shows the dataset for frames in range [51, 60]; here O-TRuE has its lowest cross-correlation value from 53 to 54; as can be seen, the image quality drastically changes in this range.



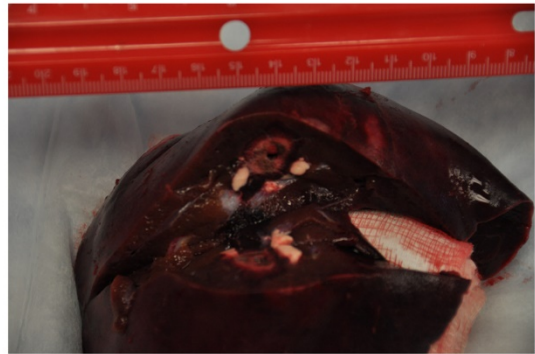
A



B



C



D

Figure 4.11 Animal Experiment setup: An in-vivo animal experiment was performed on a pig liver; an ablation was induced in the liver using RITA ablator as shown in (C). Elasticity image can be seen in (A), corresponding B-mode image in (B). The ablation region was approximately 2 cm in diameter as validated by gross pathology of the liver in (D).

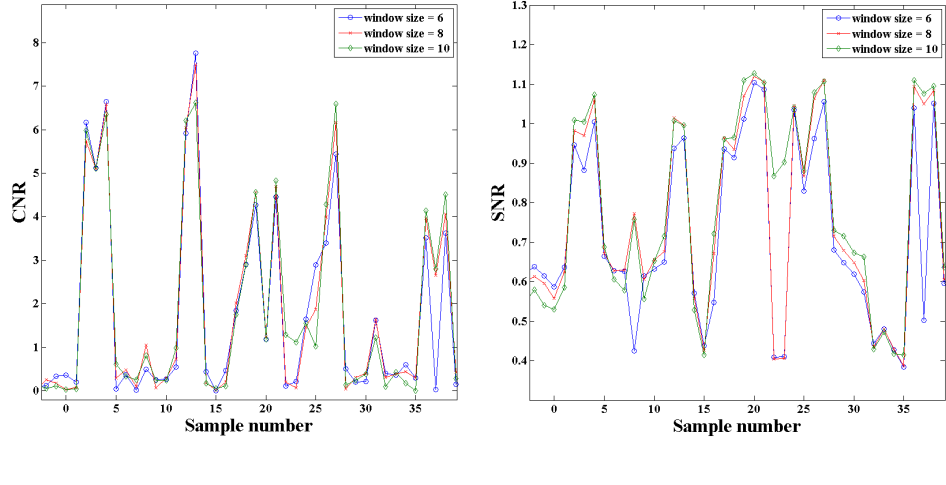


Figure 4.12 Trend of untracked elastography for in-vivo pig data: NCC window size vs. CNR and SNR: The graph shows variation of CNR and SNR of individual sample points for different NCC window sizes with untracked elastography. The data was obtained from in-vivo experiments on 350 samples and 199 samples were selected after ignoring invalid strain values. (A) Shows snapshot of CNR values and (B) shows snapshot of SNR values varying for a small subset of the 199 samples. The average/min/max values of the CNR and SNR are listed in Table 4.3. The CNR and SNR across different window sizes are closely related per sample but the global variation in CNR and SNR is high due to wide range of values.

Figure 4.10-C shows a subset of the sample elastography image stream with corresponding quality measurements for the full stream sample in Fig. 4.10-B where consecutive elastography images are compared using NCC. When comparing the max correlation of consecutive frames obtained from O-TRuE and normal (untracked) elastography, it is found from the graph in Fig. 4.10-B that O-TRuE has high and relatively stable correlation, whereas untracked elastography has correlations of low and more rapidly varying values.

We represent the percentage of images having a correlation value above a user-defined threshold ρ for a given value of σ (eq. 2, 7, and 10) as

$$\beta_{\sigma} = \frac{\#\{x_{\sigma} | x_{\sigma} > \rho\}}{\#\{x_{\sigma}\}} \times 100 \quad (14)$$

where x_σ is an array of correlation values for an image stream acquired using the setting σ , $0 \leq \rho \leq 1$ is the threshold on the correlation values, and ‘#’ is the standard set notation indicating the number of elements in a set. Table 4.2 shows the percentage of frames having max correlation values above 0.6 ($\rho = 0.6$) for different O-TRuE buffer sizes and for $1 \leq \sigma \leq 15$ with step size 1 (see eq. 2, 7, and 10). Figure 4.10 shows results for $\sigma = 1$ and buffer size 10. The last column in the table provides results for normal/untracked elastography. Table 4.2 indicates that O-TRuE outperforms untracked elastography at all buffer sizes by a factor of $\sim 1.9X$ to $\sim 2.7X$. It is thus observed that O-TRuE is more stable than untracked elastography in terms of consistent image quality. A snapshot of an elastography image frame sequence for frames 51-60 is shown in Fig. 4.10-C. Due to the moving window buffer, O-TRuE sometimes picks up the same RF frame pair as the previous image in the stream sequence. It can be observed that the frame correlation for O-TRuE drops dramatically from 52 to 53. There is a rise between frames 53 and 54 because the images are constant although void of features. The correlation drops again from frame 54 to 55 but is stable for the remainder of the subset. On the other hand, for the untracked image stream correlation is low for the majority of frames.

4.5.5 In-Vivo Animal Experiments

Here we present results from the in-vivo animal ablation study described in subsection *In-Vivo Animal Experiments* within *Experiments* section. As seen in Fig. 4.11-C, a 4DL14-5/38 probe was placed just above the ablation region of the liver for collecting 2D data. Figure 4.11-A shows the elastography image of the ablated region. As can be seen, it has a

Table 4.2 Percentage of consecutive frame pairs above a certain threshold of max correlation for varying σ values as described in eq. 2, 7, 10 and eq. 14. As an example $\sigma = 1$ (β_1) and buffer size 10 indicates percent of correlation values above the range 0.6 for the graph in Figure 4.10. As can be seen in this table, in most cases, the quality of the output system improves with the increasing buffer size.

Buffer size	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}	β_{13}	β_{14}	β_{15}	Untracked elastography
10	91.07	90.05	88.78	86.99	86.73	84.69	83.42	84.18	82.14	83.93	84.69	84.18	84.18	84.18	84.69	43.11
20	95.26	93.95	93.79	93.79	92.81	91.18	90.85	89.54	88.56	88.24	87.91	86.93	85.29	86.27	85.95	36.60
30	94.95	95.90	92.74	92.74	93.69	93.38	92.74	91.80	91.17	90.85	92.11	93.38	92.11	92.11	91.48	33.44
40	97.11	96.06	96.33	97.11	95.80	95.28	94.23	95.54	95.28	93.96	92.91	92.65	92.91	93.18	92.13	41.73
50	98.66	98.39	98.12	97.31	96.77	95.97	94.09	94.35	94.89	94.89	94.09	93.28	92.74	93.82	93.28	37.37

better contrast than the corresponding B-mode image in Fig. 4.11-B. Figure 4.11-D shows the gross pathology of the ablated region, which shows an ablation of approximately 2 cm in diameter. In Fig. 4.11-B the contour of the ablated region is not clearly visible since the thermal transfer did not drastically change the acoustic impedance of the tissue. Elastography images as shown in Fig. 4.11-A clearly shows a better contrast and boundary of the ablated region. This indicates that the multi-stream GPU-based elastography functions well in in-vivo experiments.

CNR and SNR for the elastography images were calculated at NCC window sizes of 6, 8, 10, 12, 14, and 16 using a fixed maximum NCC search distance of 2 mm and step-size overlap of 98% on 350 images. Around 200 images were chosen after ignoring decorrelated RF image pairs due to the effect of out-of-plane motion. Elastography computation speed was assessed by processing the first 100 RF image pairs 20 times. The effect of varying window size on speed and on mean, max, and min CNR and SNR values is presented in Table 4.3. It can be seen that optimal mean CNR (3.56) and near-optimal mean SNR (0.94) is achieved for window size 10. The SNR value increases as we increase the window size; this happens because increasing the window size while keeping the percentage of overlap the same results in cross-correlation being computed on a bigger area to find the best match between the template and target areas in two images. CNR initially increases with window size but decreases moving beyond window size 10. There is a wide range of CNR values observed in the images, as evidenced by the high standard deviation and min/max values at each window size. A closer look at this variation is provided in Figure 4.12, which plots the CNR and SNR computed for each sample image at the first three window sizes.

Table 4.3 Untracked elastography: NCC window size vs. speed and image quality: The table shows the change in frame rate and in CNR and SNR according to NCC window size of the multi-stream elastography. We varied the window size while fixing the maximum NCC search distance at 2 mm and overlap of 98%. The CNR and SNR were averaged for 198 images. The speed was calculated by calculating elastography images for the first 100 RF pairs 20 times. It is found that window size of 10 is optimal with high mean CNR and a good mean SNR value; although the highest mean SNR value corresponds to window size 14. This table indicates that as the window size increases the mean CNR and SNR increase along with a reduction in frame rate. Intermediate frame rates corresponding to window sizes 8 and 10 give satisfactory mean CNR and SNR and a high frame rate of 52.07 and 48.16 respectively.

window size	fps	CNR		SNR	
		min/max	mean	min/max	mean
6	56.23(± 0.71)	0.0027/9.05	2.92(± 2.31)	0.3572/1.43	0.82(± 0.25)
8	52.07(± 0.96)	0.0054/9.05	3.41(± 2.44)	0.3782/1.43	0.90(± 0.24)
10	48.16(± 0.57)	0.0075/9.18	3.57(± 2.41)	0.4128/1.39	0.94(± 0.21)
12	44.87(± 0.57)	0.0091/9.20	3.53(± 2.37)	0.4140/1.37	0.95(± 0.20)
14	41.68(± 2.47)	0.0396/8.76	3.40(± 2.21)	0.3993/1.36	0.97(± 0.19)
16	39.65(± 0.66)	0.0673/8.33	3.28(± 2.11)	0.3455/1.28	0.97(± 0.18)

4.5.6 da Vinci Surgical Robot Palpation Analysis

We apply elastography stream analysis on untracked elastography images generated by robot assisted palpation using the da Vinci Surgical System. The normalized cross-correlation between matched template and target regions of sequential output elastography images for different palpation frequencies and commanded amplitudes is shown in Fig. 4.13-B. Figure 4.13-A shows the NCC template region and the NCC target search region applied to the output elastography images. At a frame rate of 20 fps with a laparoscopic ultrasound (LUS) probe, it is observed that a very high β value (as defined in eq.14) of 96.58 is obtained corresponding to the palpation frequency of 5 Hz and commanded amplitude of 3 mm giving the most stable elastography stream. Each β value is calculated for 1200 elastography image pairs. The graph in Fig. 4.13 follows a sinusoidal pattern; this pattern reflects the sinusoidal motion of the LUS probe attached to the arm of the da Vinci

system. This sinusoidal motion is reflected in all the cases presented in Fig. 4.13. The results of the remaining combinations of frequency and amplitude are above 0.6. These results quantify the quality of the output elastography stream for robot controlled elastography.

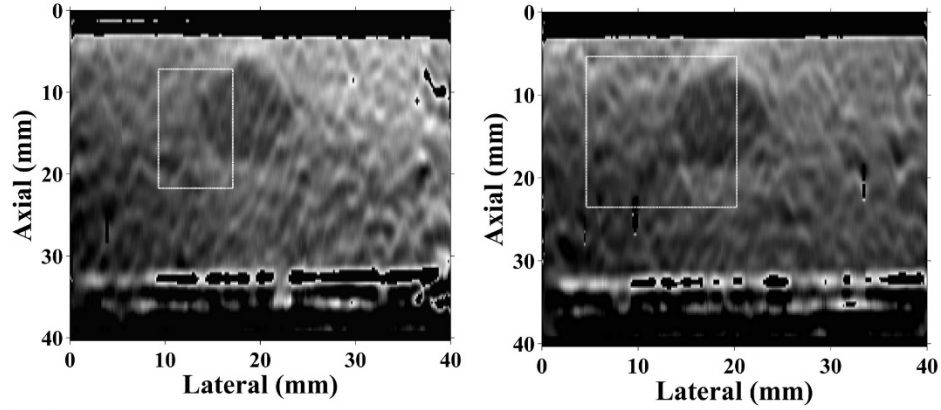
4.6 Discussion

This paper presented an exhaustive study of a real-time multi-stream GPU-based elastography system with demonstration in three applications including tracked (O-TRuE) phantom experiments, untracked in-vivo experiments, and untracked phantom experiments with robot controlled palpation. The real challenge is to tune the implementation of the complex elastography algorithm to meet the needs of a practical real-time system, i.e. it has to be reliable, have constant response time, and provide high-quality results. In addition, our system is highly modular and cost effective due to increasing performance of main-stream GPGPUs.

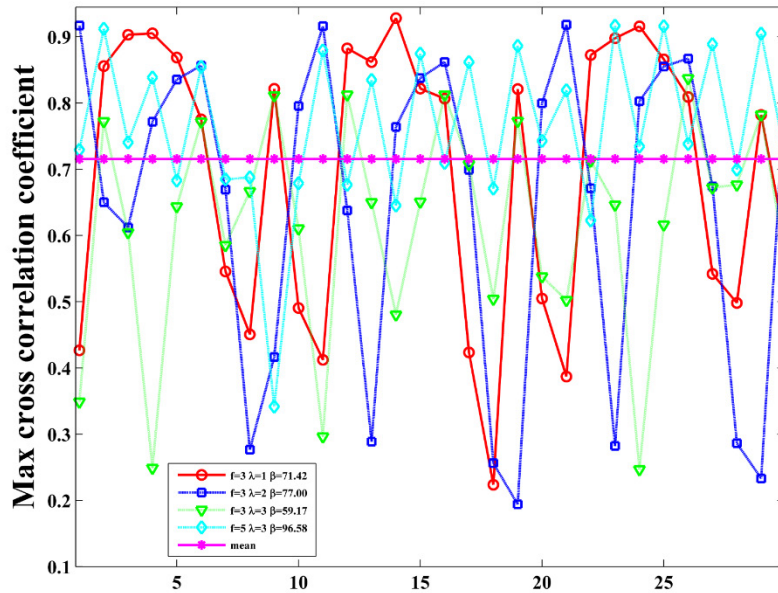
The maximum speed of 78 frames per second achieved by our elastography system approaches the RF data acquisition speed of current ultrasound systems. These results were obtained by calculating elastography over the entire image. Further performance improvement could be achieved by limiting the elastography computation to a sub-region of the image once it has been established that a target of interest, such as a tumor, is located in a particular area of the image. Initial improvement can be achieved by ignoring the border samples of RF data since tissue compression due to transducer motion is more axial in the central area of the image.

The system is highly modular and connected via the OpenIGTLinkMUSiiC API. This grants the ability to connect our system to various open source frontend modules, such as 3D Slicer for advanced visualization of the image stream. Since tracking information is embedded in each frame, advanced visualizers could, for example, allow spatial visualization of the elastography data in correspondence with 3D B-mode data. This feature also enables the ability to store and retrieve elastography data based on tracked position and timestamp. The highly modular framework enables the algorithm to run on multiple GPU's stationed at one or multiple computers and to combine streams of data from various sources. Data synchronization and ordered sequencing from multiple GPU's can be a challenge, but it is achievable.

Our current system suffers from some network latency between the different system components, including the ultrasound machine, GPU server, and data synchronizer. One potential solution is to integrate the system onto one ultrasound machine and connect various components through memory mapped inter-process communication. A clear advantage of our current implementation, however, is that the CPU of the ultrasound machine is not tasked beyond its primary function of RF data acquisition.



A



B

Figure 4.13 Max cross-correlation graph of consecutive images from robot assisted palpation: Max cross-correlation graph performed on consecutive frames for da Vinci surgical system. (A) Shows a template region selected in the leftmost image and a target region selection in the rightmost image. We apply normalized cross-correlation in these regions as shown in eq. 13 to find max correlation value. (B) Shows a mean correlation graph of initial 30 elastography image pairs out of 1200 elastography image pairs. Palpation parameters is expressed as frequency (f) in Hz and amplitude λ in mm. High correlation indicates a good match in consecutive frames; clearly, $f = 5$, $\lambda = 3$ indicates a very stable and consistent result with $\beta = 96.58$, where β is defined in eq. 14. Each β value is calculated for 1200 elastography image pairs.

The ranking of O-TRuE images in Fig. 4.8 show that a Crr value above 0.45 provides a good quality elastography image. The relatively stable correlation values shown in Fig. 4.10 for images generated by O-TRuE indicates that the O-TRuE algorithm increases the stability of the output image stream over untracked data. A possible optimization of the algorithm to further stabilize the output image stream would be to filter out frames having poorly correlated RF image pairs by collecting empirical evidence to establish a lower threshold on the Crr value required to produce a good elastography result. In addition to transducer motion, movement of the patient may also affect the quality of an elastography image. Since patient motion is untracked, O-TRuE cannot currently account for this. An image based tracking mechanism could be used as an adjunct to detect patient motion, such as by applying NCC to a small region in the center of the RF data to compute image motion in both lateral and axial directions. This information could then be used by O-TRuE when computing Crr values in order to make the algorithm more robust to patient motion. The O-TRuE algorithm could also be applied in the context of robot assisted palpation using the robot's kinematics to track the position of the ultrasound probe.

Image fusion of multiple elastography images helps to improve the quality of the elastography image, but can potentially add noise to the image. Possible approaches to address this could be enforcing a minimum threshold on the Crr value as discussed earlier and increasing the buffer size with the support of multiple GPUs if needed.

As indicated by our assessment of robot assisted palpation in Fig. 4.13, the stability and quality of the output elastography image stream is affected by variation of the palpation frequency and displacement. An enhancement to the system would be to use the measured

image correlation as feedback to the robot to autonomously vary the frequency and amplitude of palpation to determine the optimal setting. A high speed elastography engine as we have presented is a necessary prerequisite to enable such an approach.

The in-vivo animal experiment showed good contrast between the ablated region and background tissue. Table 4.3 and Fig. 4.12 indicate that high speed untracked elastography provides good quality CNR and SNR values. A more exhaustive study would help to more fully understand the effects of window size on speed, CNR and SNR.

5 Elastography image pair selection via elastography computation on limited regions of interest

Summary

We present a method and a real-time system for selecting a radio frequency (RF) data pair to generate high-quality freehand elastography images. We have developed an image-based in-plane slice detection method. GPU-based approaches have decreased the computational time for computing elastography, and we present a method for pre-computing elastography in multiple regions of interest in pre- and post-compressed RF data pairs. The elastography computation in the limited region of interest reduces the need to calculate the entire image to determine whether an elastography image is good or bad. The average correlation and signal-to-noise ratio (SNR) of the elastography map in the region of interest in the RF data pair are computed to estimate the quality of the full elastography image. Weighted averaging of the different regions of interest is performed to determine the quality of the image.

In this chapter, I designed the algorithm, performed the experiments, analyzed the data, and wrote the chapter. Dr. Russell H. Taylor reviewed the work and recommended the statistical tests to validate the results for statistical significance.

5.1 Introduction

5.1.1 Motivation

Ultrasound elastography involves a palpation motion of the ultrasound transducer on the surface of the organ. Pre- and post-compression radio frequency (RF) data are collected to determine the correlation and generate a displacement map of speckle movement using techniques such as normalized cross-correlation (NCC) [1], [84], convolution [73], dynamic programming [56], and analytical minimization [86]. The displacement map is further processed to calculate the strain to differentiate the soft and hard regions [1].

The out-of-plane motion of two RF data frames is one of the major causes of de-correlation in the RF images, which affects several real-time freehand elastography implementations. Surgeon hand tremors, patient movement and organ movement can cause this motion. A real-time online tracked ultrasound elastography solution that was implemented in a previous study [2] uses external tracking information and robotic palpation to generate in-plane RF data pairs. External tracker-based in-plane motion detectors require specialized hardware, such as an electromagnetic tracker, an optical tracker or a Kinect system. However, the in-plane motion of the probe does not guarantee the steadiness of the underlying tissue. Apart from this limitation, it is difficult to use an electromagnetic tracker in an operating room environment due to the presence of ferromagnetic materials; optical tracking systems suffer from full or partial occlusion due to interference from various objects between the camera and the optical tracker [2]. An image-based in-plane motion estimator is needed to overcome this problem. Due to the high-speed computation achieved

using GPU-based elastography methods, it is possible to divide the incoming RF image pairs into several regions of interest and compute elastography on those regions of interest. These regions of interest can then be analyzed, and only an elastography image with high CNR and SNR values may be selected. A previous study [87] involved computing the motion estimation using RF data tracking and combining that estimate with the quality of the continuous output elastography images. From this assessment, a performance descriptor is calculated to give a quantitative measure of the output elastography images. A similar study [88] combined the performance of consecutive B-mode images and consecutive strain images to obtain a similar performance descriptor. However, these methods employ analysis and strain generation for entire RF data images that may lead to slower performance. In our method, we calculate elastography on selective RF lines that encapsulate the region of interest. By using GPU, we take the advantage of high-speed elastography calculation, that allows us to select larger regions of interest. We combine the average correlation and SNR values for these regions of interest as a weighted average to obtain an assessment of future elastography images.

5.1.2 Statements of Significance

In computer-assisted interventions, a very high temporal sensitivity is needed to obtain real-time feedback for surgeons. In ideal scenarios, the elastogram displayed in the real-time elastography system should be correlated, in terms of time, to the RF data collected during the palpation motion. The network latency is dynamic but can be considered to be constant in our case. A GPU-based elastography algorithm offers rapid computation of

NCC elastography [2][89]. We integrated this GPU-based elastography algorithm with real-time online tracked ultrasound elastography in a previous study [2]. Assuming that the axial motion is parallel to the palpation motion, this system cannot process de-correlated frames due to high lateral or elevational motion of the organs. These systems must also be calibrated, and errors in calibration can lead to cascaded errors in the tracking data. Additionally, we synchronize this tracking data with the RF data based on a timestamp [24].

Several commercially available ultrasound systems perform post-processing of these elastography images to filter out low-quality strain output images. Post-processing is a time-consuming step in which a processor performs computationally expensive elastography image calculation, followed by another layer of computationally expensive post-processing operations.

Despite advancements in GPU-based techniques that accelerate performance, an even faster filtering mechanism before computing the GPU-based version will assist in improving the response time of the system. We can define the response time as the time from ultrasound RF data collection to the display of elastography images on the final screen. We must discard low-quality and highly de-correlated elastography images, or the gain obtained from GPU acceleration is diminished [69].

5.1.3 Contributions

In this chapter, we propose a real-time system that computes elastography for RF lines in which the limited regions of interest lie, calculates the average correlation and SNR values

in those regions of interest, takes a weighted average of those regions according to their locations with respect to a central axial RF line, and filters images above a certain threshold of these values. The proposed method will efficiently determine the quality of an elastogram through the application of elastography on user-specified regions of interest between the images in the incoming RF data pair. This chapter analyzes and validates the results using in vivo animal data.

5.2 Methods

Our method estimates the quality of the elastography image by first calculating elastography on RF lines in a given region of interest. If we imagine the free-hand palpation motion of an ultrasound transducer, in most cases, the area corresponding to the good elastography strain images corresponds to the central RF lines, and this area receives predominantly axial motion with a small amount of lateral motion. The areas corresponding to the region of interest at the top left and right corners, and at the bottom left and right corners are also displaced axially but also experience lateral displacements. Thus, we give more weight to the quality of strain at the center of the images to estimate the quality of the output strain when we calculate the elastography for the entire sample inside the RF data. We define the cumulative weighted average correlation (cCorr) as the weighted average of the average correlation (aCorr) values in different regions of interest. The following equation gives the cCorr score:

$$cCorr = \frac{\sum_i \lambda_i \times aCorr_i}{\sum_i \lambda_i} \quad (1)$$

where $\lambda_i, i = 1 \dots n$ are the weights for the regions of interest with the average correlation $aCorr_i$. Similarly, we define the cumulative weighted SNR (cSNR) value as the weighted average of the SNR values in different regions of interest. This factor is given by

$$cSNR = \frac{\sum_i \gamma_i \times SNR_i}{\sum_i \gamma_i} \quad (2)$$

where $\gamma_i, i = 1 \dots n$ are the weights for the regions of interest with signal-to-noise-ratio SNR_i .

As seen in Fig. 5.1 and Algorithm 5.1, the system receives RF image pairs from the ultrasound machine and performs elastography on RF lines that encapsulate the region of interest. The elastography is computed on the GPU and uses less overhead memory by transferring only the RF data for regions of interest rather than the total data for the RF. Then, the program extracts the average correlation and SNR values from the calculated regions of interest. Figure 5.2 illustrates how this 1:1 correspondence may appear. Using Eqs. 1 and 2, we calculate the $cCorr$ and $cSNR$. If the values of $cCorr$ and $cSNR$ are above a certain threshold, then the elastography is calculated for the entire region. This verification step is called just before the multistream GPU thread specified in Chapter 4 is called by the program. The generated elastography image is then passed on to the network or saved in the files.

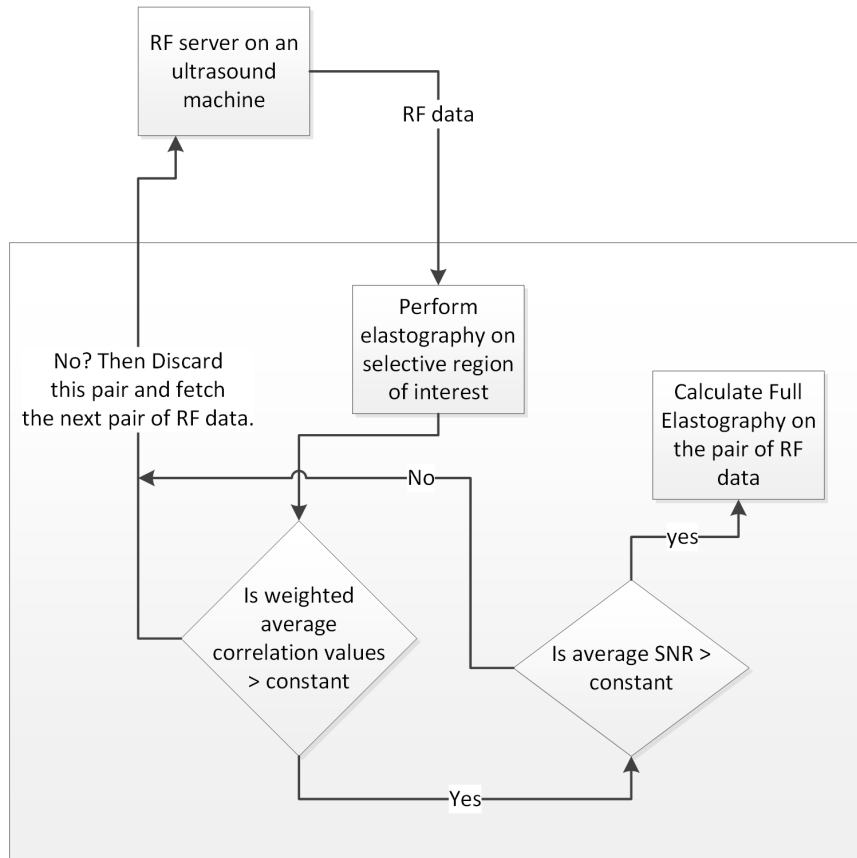


Figure 5.1 System diagram of an image-based elastography quality detection system: The figure shows the flow of our method, in which the RF server sends the RF data to the selector, where the elastography is computed for a selected region of interest. If the cumulative weighted average correlation and the cumulative weighted SNR values pass a certain threshold, then the elastography is computed for the full images for the RF image pairs.

Algorithm 5.1: Elastography image selection based on elastography on the RF lines containing the regions of interest

1. Receive RF data pair.
 2. Select ROIs in both RF data images.
 3. Compute elastography on the corresponding RF lines such that the ROIs are enclosed within those RF lines.
 4. Calculate the cCorr on the average correlation from the elastography on the ROIs, as specified in Eq. 1.
 5. Calculate the cSNR from the elastography on the ROIs, as specified in Eq. 2.
 6. If $cCorr > \text{constant_1}$ and $cSNR > \text{constant_2}$,
 calculate elastography for the received RF data pair
 Otherwise,
 discard the current RF data pair and go to 1
-

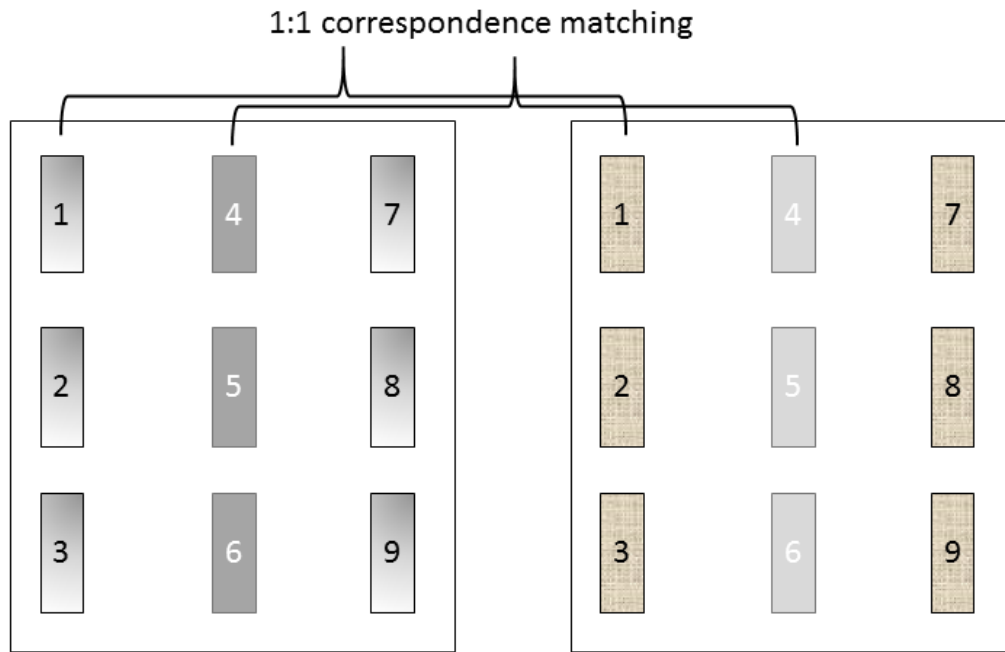


Figure 5.2 Elastography on selective regions of interest: The RF images are divided into regions of interest, and elastography is computed for the entire RF lines such that the selected regions of interest are calculated. We then calculate the average correlation and SNR values in the regions of interest. Greater weight is given to values in the regions of interest that are near the central areas of the image.

5.3 Results and Discussion

5.3.1 Experiments

For the purpose of this experiment, we have reused the animal data as specified in Chapter 4.

The experiments were performed in-vivo on a live animal. Fiducials were artificially implanted inside the liver of the pig, and ablation was performed using RITA ablator. Palpation was performed, and data were recorded using OpenIGTLinkMusiiic protocol to be saved on the disk. These data are now reusable, and more information can be found in Appendix B. A single ablation burn was used to target one fiducial. Approximately 354 RF-images were used to evaluate the results. We performed elastography on consecutive RF-image pairs. The filtering algorithm filtered most of these consecutive RF-image pairs and selected N RF pairs for possible good quality elastography images. The N values are reported for each experiment. The threshold values of constant_1 that were compared with the cCorr (cumulative weighted average correlation) and the values of constant_2 (cumulative weighted average SNR values), to be compared with cSNR are also specified in the results section.

Filtered vs. unfiltered elastography

In this study, we determine how our algorithm (filtered elastography) performs in comparison to the unfiltered technique (freehand elastography without any pre-processing). For the purpose of this study, we calculate the CNR and SNR of the elastography output image for both filtered and unfiltered elastography. We plot the histogram of the CNR and SNR on separate graphs to determine how both filtered and

unfiltered elastography perform. In this experiment, N is 106. The top N EI images are selected from unfiltered elastography for comparison with this result.

Performance for varying widths and heights of the region of interest

We need to determine what happens when we change the width and height of the regions of interest for the same weight values in Eqs. 1 and 2. This study will help determine whether a substantial difference occurs when the width and height are changed. We again plot CNR and SNR for different widths and heights. In this experiment, N is 106. The top N -consecutive EI images are selected from each group for the comparison.

Performance for varying weights

To evaluate the performance with different weights, we changed the weights in the center of the region of interest because the ultrasound probe pressure is greater towards the central axial lines than on the sides. In this experiment, N is 80. The top N consecutive EI images are selected from each group for the comparison.

Visual classification of filtered and unfiltered data

In this experiment, we laid out the output images obtained via the freehand and filtered elastography and visually inspected the quality of the output image. We labeled good quality images with 1 (\times symbol) and bad quality images with 0 (o symbol). We then listed the CNR and SNR values of these images. The raw data can be found in Figure 10.1-10.6. The total number of samples is $N=106$. The visual classification user is Nishikant Deshmukh, and the system is subject to different results for different users.

5.3.2 Results

5.3.2.1 Filtered vs. unfiltered elastography

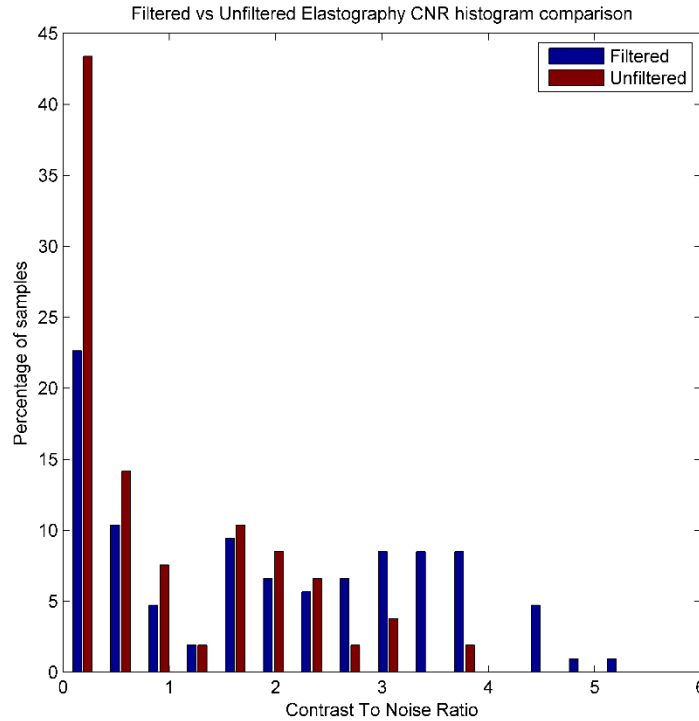


Figure 5.3 Comparison of filtered vs. unfiltered elastography based on CNR histograms: For a low average CNR value of 0.1804, nearly ~43% of the unfiltered elastography images fall at this average CNR value and only ~22% of the filtered elastography images correspond to this average value. For values above the good quality CNR value of 2.6982, a total of ~34% of the filtered elastography image fall in that region, and a total of only 7.54% of the freehand elastography data correspond to that region. This finding indicates that the filtered elastography algorithm performs better than unfiltered elastography.

We performed this experiment to determine how filtered elastography performs in comparison with unfiltered elastography. We calculated the CNR of the lesion in comparison to its background for both filtered and unfiltered elastography image sequences. Then, we created a histogram comparison of filtered vs. unfiltered elastography to determine where the distribution lies for the data. Lower values of CNR indicate poor-quality data. We define A as a filtered elastography image and B as an unfiltered

elastography image and used the convention (A, B) as the tuple for the frequency count for the histogram shown in Fig. 5.3. The centers of each histogram bin are located at {0.1804, 0.5401, 0.8998, 1.2595, 1.6192, 1.9789, 2.3385, 2.6982, 3.0579, 3.4176, 3.7773, 4.1370, 4.4966, 4.8563, 5.2160}, with CNR values (as a percentage of the total samples) with the corresponding bin tuple, as follows: {(22.64, 43.39), (10.38, 14.15), (4.71, 7.55), (1.89, 1.89), (9.43, 10.37), (6.6, 8.49), (5.66, 6.6), (6.6, 1.89), (8.49, 3.77), (8.49, 0), (8.49, 1.88), (0, 0), (4.72, 0)}. This representation means that the bin with the center value at 0.1804 has a percentage of the total samples for A and B that is represented by the tuple (22.64, 43.39). The total number of samples is 106.

Table 5.1 Filtered vs. unfiltered elastography CNR data: Statistical significance results. These data correspond to the data displayed in Fig. 5.3, with sample size N = 106. The data were tested for using both a t-test and ANOVA to determine the statistical significance of the experiment. With a p-value < 0.0001, the experiment is extremely statistically significant. These data indicate that filtered elastography performs better than unfiltered (freehand) elastography.

Group	Filtered	Unfiltered
Mean	1.9315	0.9795
Variance	2.1012	0.9581
SEM	0.1408	0.0951
p-value t-Test	<0.0001	
p-value ANOVA	<0.0001	

As shown in Fig. 5.3, in comparison to unfiltered elastography, the filtered elastography data show a smaller percentage of samples with CNR values in the range between 0 and 1. With an average CNR value of 0.1804, the percentage of samples for the unfiltered elastography algorithm is high at 43.39% and the percentage of samples for the filtered data is 22.64%. Those figures represent a nearly 50% reduction in the total number of images for filtered elastography data in that range. This trend reverses for good average

CNR values above 2.6, indicating that filtered elastography generates a better-quality output stream. Above a CNR value of 2.6982, the total percentage of the filtered elastography data is 36.79%, and the total percentage of samples for the unfiltered data is 7.55%.

Table 5.1 presents the data related to the statistical significance test. The data are also presented as a histogram chart of percentage values in Fig. 5.3, and the raw data can be found in Tables 10.1-10.2. The sample size is $N=106$. The mean of the CNR values of the filtered elastography algorithm is $1.9315 (\pm 1.44952)$. For unfiltered freehand elastography, this value is $0.9795 (\pm 0.9788)$. This finding indicates that filtered elastography performs better than unfiltered elastography with respect to the mean values. Figure 5.3 shows the histogram map that shows the underlying distribution of the values, indicating that filtered elastography performs better than freehand elastography. Table 5.1 also presents the p -value < 0.0001 for the t-test and ANOVA, indicating that this experiment was extremely statistically significant. This statistical significance result supports our assertion that filtered elastography performs better than freehand elastography.

We define C as filtered elastography images and D as unfiltered elastography images and use the Convention (C, D) as the tuple for the percentage of total samples for the histogram shown in Fig. 5.4. The $\{(1.89, 36.79), (2.83, 14.15), (17.92, 15.09), (65.09, 33.02), (10.38, 0.94)\}$ tuple values (as the percentage of total samples) have $\{0.4891, 0.6898, 0.8905, 1.0912, 1.2920\}$ corresponding centers on the x-axis for SNR values. The total number of samples is 106; the raw data are located in Tables 10.3-10.4.

As shown in Fig. 5.4, the unfiltered elastography shows higher SNR values in the range of 0 to 1. Thus, for low SNR values, more data are available for unfiltered elastography. The total percentage values are 66.03% for the unfiltered elastography data and 22.64% for the filtered elastography data. The trend reverses in the range 1 to 1.5, where the filtered elastography SNR values occupy 75.47% of the total sample space, and unfiltered elastography occupies 33.96% of the total sample space. This percentile result indicates that unfiltered elastography performs better than freehand elastography.

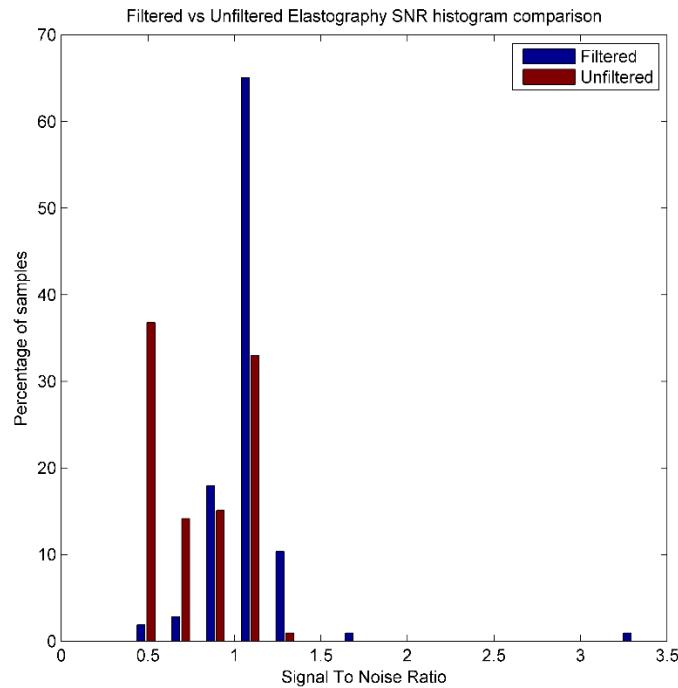


Figure 5.4 Filtered vs. unfiltered elastography SNR histogram comparison: For SNR values in the range 0 to 1, a total of ~66% of the unfiltered elastography images and 22.64% of the filtered elastography fall under these value range. Lower SNR values indicate that more low-quality images are acquired with unfiltered elastography. The values change for the range 1 to 1.5, with a total of 75.47% of the filtered elastography images falling under the threshold, and the remaining 33.96% of the freehand elastography images falling under the threshold. This finding indicates that filtered elastography performs better than unfiltered elastography.

Table 5.2 indicates the mean SNR values for filtered elastography as 1.0822 (± 0.2805) and unfiltered elastography as 0.7808 (± 0.2773). This result again reflects better performance

for filtered elastography in comparison to unfiltered elastography. The p -value < 0.0001 for the two groups for both the t-test and ANOVA indicate extremely statistically significant results. This statistical significance result supports the histogram analysis, as shown in Fig. 5.4.

Table 5.2 Filtered vs. unfiltered elastography SNR data: Statistical significance results. The data presented in this table correspond to the data displayed in Fig. 5.4. The sample size is $N = 106$. The purpose of this t-test and ANOVA is to determine the statistical significance of this experiment to indicate whether the results are repeatable and did not happen by a chance. With p -values for both the t-test and ANOVA < 0.0001 , the experiment is extremely statistically significant. This finding indicates better performance for the filtered elastography algorithm than the unfiltered (freehand) elastography version.

Group	Filtered	Unfiltered
Mean	1.0822	0.7808
Variance	0.0787	0.0769
SEM	0.0272	0.0269
p-value t-Test	< 0.0001	
p-value ANOVA	< 0.0001	

5.3.2.2 Performance for varying widths and heights of the regions of interest

We compared CNR histograms with varying widths and heights (in pixels) of the comparison window to determine if varying the width and height has any effect on the CNR values. We define the tuples in the case of Fig. 5.5 as ($E = \{w=10, h=10\}$, $F = \{w=15, h=15\}$, $G = \{w=20, h=15\}$, $H = \{w=25, h=15\}$). Here, w is the width and h is the height of the regions of interest. This w and h are consistent across all regions of interest.

The tuple values (detailed as the percentage of total samples) shown in Fig. 5.5 are $\{(33.02, 27.35, 21.69, 22.64), (8.49, 6.6, 7.55, 6.6), (8.49, 8.49, 9.43, 7.55), (3.77, 3.77, 5.66, 5.66),$

(9.43, 9.43, 11.32, 11.32), (12.26, 12.26, 10.38, 12.26), (6.6, 9.43, 10.38, 8.49), (6.6, 6.6, 7.55, 7.55), (6.6, 6.6, 4.72, 4.72), (0.94, 0, 0.94, 0.94), (1.89, 4.72, 5.66, 6.6), (0.94, 0.94, 0.94, 1.89), (0.94, 1.89, 0.94, 0.94), (0, 0.94, 0.94, 0.94), and (0, 0.94, 1.89, 1.89)} located at {0.1756, 0.5237, 0.8718, 1.2199, 1.5680, 1.9161, 2.2642, 2.6123, 2.9604, 3.3085, 3.6566, 4.0047, 4.3528, 4.7009, 5.0490} center locations on the x-axis of the CNR values. For the values between region 0 and 1.22, E has a total of 53.77% of its samples, F has 46.22% of its samples, G has 44.34% of its samples, and H has 42.45% of its samples. For CNR values between 1.5 and 3, E has a total of 41.5% of its samples, F has a total of 44.34% of its samples, G has a total of 44.34% of its samples, and H has a total of 44.34% of its samples. For values, above 3, E has a total of 4.71% of its samples, F has a total of 9.43% of its samples, G has a total of 11.32% of its samples, and H has a total 13.2% of its samples. As can be seen from this data, the w and h corresponding to G and H have a slight advantage when compared to the values corresponding to E and F. To obtain additional insight into this trend, we look at the SNR values in Fig. 5.6. The threshold for $cCorr$ is 0.15, and the threshold for $cSNR$ is 0.01.

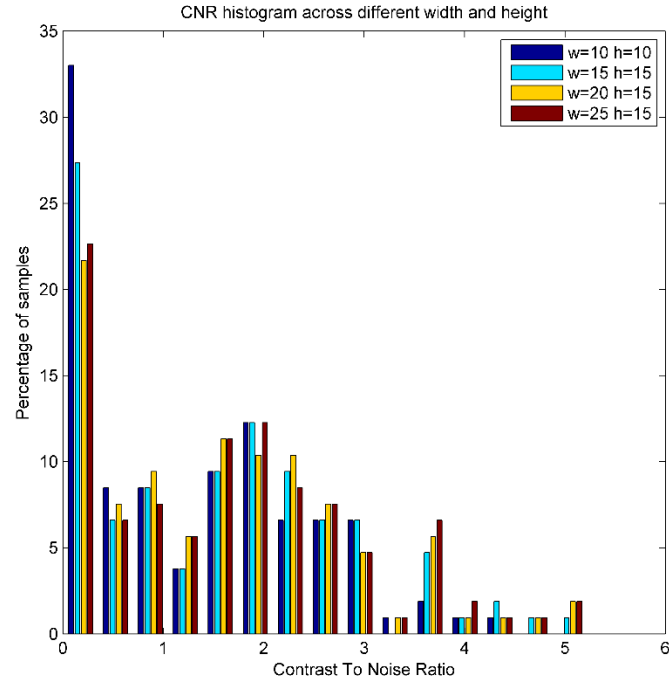


Figure 5.5 CNR histogram across different width and height values: We represent width (in pixel) with w and height (in pixel) with h for the regions of interest. For the values between region 0 and 1.22, a total of 53.77% of images with $w = 10$ and $h = 10$ fall in this region, which is higher than the percentage observed for other widths and heights. For the CNR values between 1.5 and 3, the total percentage of samples for $w = 10$ and $h = 10$ is 41.51%. This value is slightly less than observed for other widths and heights. For CNR values above 3, $w = 25$ and $h = 15$ and $w = 20$ and $h = 15$ fare better, with 11.32% and 13.2% of their images, respectively. From the CNR data, we can conclude that $w = 25$ and $h = 15$ has a slight advantage than other values.

In Table 5.3, we calculate the mean and variance of the data that correspond to Figure 5.5, for which the sample size $N=106$ and the raw data can be found in Tables 10.5-10.8.

The groups in Table 5.3 indicate different width (w) and height (h) values for the ROI used in the algorithm for different CNR values. The mean and standard deviation of the data are 1.2887 (± 1.1164) for group 1 ($w=10, h=10$), 1.5387 (± 1.2537) for group 2 ($w=15, h=15$), 1.6194 (± 1.2517) for group 3 ($w=20, h=15$), and 1.666 (± 1.2848) for group 4 ($w=25, h=15$). These data corresponds to the data shown in Table 5.5, and the raw data can be found in Table 10.5-10.8. The sample size is $N=106$. As can be seen from the mean values, groups 3 and 4 fare better than the other groups with group 4 slightly better than rest of the

groups. Therefore, we can conclude that group 3 and 4 perform better than the rest of the groups. We compare group 1 (with the lowest mean) with the other groups to attempt to establish a p-value by doing a t-test. As shown, t-tests for the comparisons of groups 1 and 3 and groups 1 and 4 are statistically significant. This result indicates that groups 3 and 4 perform better than the other groups. The ANOVA test with a p-value of 0.11483 test fails for all of the groups. This failure of some p-value tests is due to the low sample values for four columned data.

For Fig. 5.6, the tuples are defined as ($I = \{w=10, h=10\}$, $J = \{w=15, h=15\}$, $K = \{w=20, h=15\}$, and $L = \{w=25, h=15\}$). Here, w and h are again the selected widths and heights of the regions of interest. The tuples have corresponding tuple values (percentage of sample space) of $\{(33.96, 24.53, 22.64, 22.64), (63.2, 73.58, 75.47, 76.41), (2.83, 0, 0.94, 0.94)\}$ located at the center locations of $\{0.5828, 1.1149, 1.6469\}$. As can be seen from the data presented in Fig. 5.6, the SNR data corresponding to the width and height of I have the highest total, with 33.96% of the samples in the range 0 to 1, while J has a total of 24.53%, and K and L each have totals of 22.64% of the samples. For the range 1 to 2, I has a total of 66.03% of the samples, J has a total of 73.58% of samples, K has a total of 76.42% of samples, and L has a total of 77.36% of samples.

This finding shows that the width and height values that correspond to K and L give better results for SNR. In contrast, for the same width, we have seen that CNR has a slight advantage. Thus, it is easy to conclude that for $(w=20, h=15)$ and $(w=25, h=15)$ the algorithm gives a better result.

Table 5.3 CNR data for different widths and heights of the region of interest: Statistical significance results. The data in this table correspond to the results presented in the Fig. 5.5. The sample size is N=106. We performed pairwise t-test comparison between Group 1 (minimum mean value) and the other groups 2, 3, and 4. We found statistical significance for the comparison between Groups 1 and 3 and Groups 1 and 4, with the values of Groups 3 and 4 higher than the mean values. This finding supports our claim that the width and height values that correspond to Group 3 and 4 give better results for the CNR values. The ANOVA result of p-value 0.11483 indicates that the overall experiment is not statistically significant. This outcome can be attributed to the low sample size for a four-group scenario. We expect that the results will improve with more data validation.

Group	1	2	3	4
ROI dimensions	w=10, h=10	w=15, h=15	w=20, h=15	w=25, h=15
Mean	1.2887	1.5387	1.6194	1.6666
Variance	1.2462	1.5718	1.5666	1.6506
SEM	0.1084	0.1218	0.1216	0.1248
p-value t-test Groups 1 and 3	0.04975		<0.05	
p-value t-test Groups 1 and 2	0.1166		>0.05	
p-value t-test Groups 1 and 4	0.0307		<0.05	
p-value ANOVA	0.11483		>0.05	

Table 5.4 shows the data that correspond to Fig. 5.6, and the corresponding raw data can be found in Tables 10.9-10.12 for the SNR data values. The sample size is N=106. The purpose of this measurement is to establish t-test and ANOVA test value comparisons to determine the statistical significance of the experiment. The mean and standard deviation are 0.9013 (± 0.2782) for group 1 (w=10, h=10), 1.0335 (± 0.8044) for group 2 (w=15, h=15), 0.9749 (± 0.2697) for group 3 (w=20, h=15), and 0.9601 (± 0.2177) for group 4 (w=25, h=15). As can be seen from the table, the mean value for group 2 is highest, while the closest values are those obtained for group 3 and group 4. However, the variance for

group 2 is higher than observed for the other groups. As indicated in the histogram analysis presented in Fig. 5.6, the ROI $w=20, h=15$ is preferable. We performed t-tests by making pairwise comparisons between groups 1 and 2, groups 1 and 3, and groups 1 and 4 because group 1 has the lowest mean value. As shown, the groups 1 and 3 passes the statistical significance test. Therefore, we can say that group 3 with ROI $w=20, h=15$ has the preferred ROI dimensions. The t-test failed for the rest of the groups, with an ANOVA p-value of 0.219. However, this outcome can be improved by obtaining more result sets.

Table 5.4 SNR data for different widths and heights of the region of interest: Statistical significance results. The table results presented in the table correspond to the result presented in Fig. 5.6. The t-test is statistically significant only for groups 1 and 3. The p-value for groups 1 and 4 is also close to 0.05, which is the conventional threshold used to establish statistical significance. Like the histogram, this test also indicates that $w=20, h=15$ fares better than the other groups and yields performance results close to $w=25, h=15$. The ANOVA test fails to establish statistical significance. This failure can be corrected by including more samples in the experiments.

Group	1	2	3	4
ROI dimensions	w=10, h=10	w=15, h=15	w=20, h=15	w=25, h=15
Mean	0.9013	1.0335	0.9749	0.9601
Variance	0.0774	0.6470	0.0727	0.0474
SEM	0.0270	0.0781	0.0262	0.0211
p-value t-test Groups 1 and 3	0.0396		<0.05	
p-value t-test Groups 1 and 2	0.1274		>0.05	
p-value t-test Groups 1 and 4	0.085		>0.05	
p-value ANOVA	0.219		>0.05	

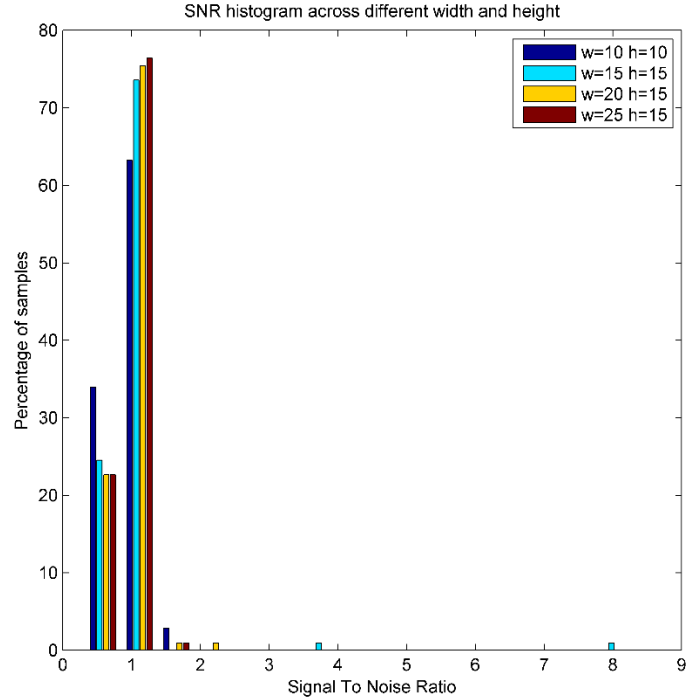


Figure 5.6 SNR histogram across different width and height values: We again represent width (in pixels) with w and height (in pixels) with h for the regions of interest. For SNR values between 0 and 1, $w = 10$ and $h = 10$ has a total frequency of 33.96% of the elastography images falling under the threshold. This is higher than the values obtained for the other w and h values. For CNR values between 1 and 2, the total frequency values for $w = 25$ $h = 15$ and $w = 20$ $h = 15$ are 76.41% and 77.35% respectively. This finding shows that $w = 25$ $h = 15$ and $w = 20$ $h = 15$ yield better results than the other width and height values. Thus, greater width and height values generate better quality data.

5.3.2.3 Performance for varying weights

In Fig. 5.7, we define the tuples as $\{(M = \{1\ 1\ 1\ 2\ 4\ 2\ 1\ 1\ 1\}), N = \{1\ 1\ 1\ 1\ 8\ 1\ 1\ 1\ 1\}), O = \{1\ 1\ 1\ 1\ 12\ 1\ 1\ 1\ 1\})\}$, where the values for M , N , and O are the weights for the different regions of interest, as shown in Fig. 5.2. For shorter representation in Fig. 5.7, we represent M as $\{2, 4, 2\}$, N as $\{1, 8, 1\}$, and O as $\{1, 12, 1\}$. The values of the tuples (percentage of t) are $\{(17.5, 15, 23.75), (7.5, 5, 3.75), (8.75, 8.75, 8.75), (6.25, 3.75, 5), (11.25, 6.25, 6.25), (8.75, 15, 12.5), (8.75, 13.75, 11.25), (7.5, 10, 6.25), (7.5, 6.25, 7.5), (6.25, 6.25, 6.25), (2.5, 3.75, 1.25), (1.25, 1.25, 1.25), (2.5, 2.5, 2.5), (1.25, 1.25, 1.25), (2.5, 1.25, 2.5)\}$,

which correspond to the central locations $\{0.1661, 0.4669, 0.7678, 1.0686, 1.3694, 1.6702, 1.9710, 2.2718, 2.5726, 2.8734, 3.1742, 3.4750, 3.7759, 4.0767, 4.3775\}$ on the x-axis of the CNR values. The CNR for window (1, 8, 1) performs better in the range (0, 0.5). The CNR performs better for the range 1.5 to 2.5, indicating that this window size is best. These results are inconclusive and can be indicated as strong in the SNR results. For low SNR values (0, 1) and high SNR values (1, 1.5), the window (1, 8, 1) is better in comparison to the other window ranges. A total of 80 samples are included in the dataset. The region of interest's width is 25 pixels and the height is 15 pixels. The threshold for $cCorr$ is 0.15, and the threshold for $cSNR$ is 0.01.

In Table 5.5, group 1 indicates the CNR value result for central weights of $\{2\ 4\ 2\}$, group 2 indicates the result for central weights of $\{1, 8, 1\}$, and group 3 indicates the result for weights of $\{1, 12, 1\}$. This result corresponds to Fig. 5.7, and the raw data are presented in Tables 10.13-10.14. The mean and standard deviation is 1.5967 (± 1.1580) for group 1, 1.6985 (± 1.0738) for group 2, and 1.5495 (± 1.1774) for group 3. This result supports the claim made in the histogram study that the weights that correspond to group 2 (i.e. $\{1\ 1\ 1\ 1\ 8\ 1\ 1\ 1\ 1\}$) perform better due to high mean and standard deviation values. The results obtained for the t-test and ANOVA ($p\text{-value} > 0.05$) fails to establish statistical significance. This failure in some p-test results is due to the small sample number of $N=80$ for three-column data. From this t-test result, the group pairs 2,3 ($p\text{-value } 0.4142$) and 1,2 ($p\text{-value } 0.5126$) have lower p-values than the group pair 1,3 (0.7979) in the t-test. The common group is group 2, and a low p-value indicates statistically better p-values. This comparison is a weak analogy but the results from the histogram in Fig. 5.7 and the high mean value for the CNR data show that group 2 $\{1\ 8\ 1\}$ is a preferred weight for these computations.

Table 5.5 CNR data with different weights for the region of interest: Statistical significance results. The results presented in this table correspond to the data presented in Fig. 5.7. The t-test and ANOVA (p-value 0.699) fail to establish any statistical significance for the overall results. However, the low p-values for group-pairs 2,3 and 1,2 indicate that the common group 2 performs better than the rest of the groups in terms of the SNR values. The results can be further improved by increasing the sample size, which is N=80 for this experiment.

Group	1	2	3
Weights	2 4 2	1 8 1	1 12 1
Mean	1.5967	1.6985	1.5495
Variance	1.3410	1.1530	1.3863
SEM	0.1295	0.1201	0.1316
p-value t-test Groups 2 and 3		0.4142	>0.05
p-value t-test Groups 1 and 3		0.7979	>0.05
p-value t-test Groups 1 and 2		0.5126	>0.05
p-value ANOVA		0.699	>0.05

For the range of 0 to 1.5, the weights corresponding to M have a total frequency of 41, the weights corresponding to N have a total of 38.75% of samples, and the weights corresponding to O have a total of 47.5% of samples. For the range 1.6 to 3, M has a total of 38.75% of samples, N has a total of 51.25% of samples, and O has a total of 31.25% of samples. For CNR values above 3.1, M has a total of 10% of samples, N has a total of 10% of samples, and O has a total of 8.75% of samples. These data indicate that the weights corresponding to N perform better, as indicated by a low-frequency count for CNR values in the range of 0 to 1.5 and a high-frequency count for CNR values in the range of 1.6 to 3.

Table 5.6 SNR data with different weights for the region of interest: Statistical significance results. This table indicates the statistical significance test for the results presented in Fig. 5.8. The t-test and ANOVA (p-value 0.4531) again fail to establish any statistical significance for the overall results. These results may improve with more samples, than the N=80 for this experiment.

Group	1	2	3
Weights	2 4 2	1 8 1	1 12 1
Mean	1.0044	1.0016	0.9574
Variance	0.1313	0.0344	0.0443
SEM	0.0405	0.0207	0.0235
p-value t-test Groups 2 & 3	0.1429		>0.05
p-value t-test Groups 1 & 3	0.3127		>0.05
p-value t-test Groups 1 & 2	0.9463		>0.05
p-value ANOVA	0.4531		>0.05

The tuples in Fig. 5.8 that correspond to $\{P = \{1\ 1\ 1\ 2\ 4\ 2\ 1\ 1\ 1\}\}$, $Q = \{1\ 1\ 1\ 1\ 8\ 1\ 1\ 1\ 1\}$, and $R = \{1\ 1\ 1\ 1\ 12\ 1\ 1\ 1\ 1\}$ are represented by $\{P = \{2, 4, 2\}$, $Q = \{1, 8, 1\}$, and $R = \{1, 12, 1\}\}$. The values of the tuples (percentage of total samples) are (11.25, 6.25, 12.5), (6.25, 8.75, 8.75), (33.75, 21.25, 28.75), (45, 61.25, 47.5), and (1.25, 2.5, 2.5), with the corresponding centers at 0.5229, 0.7256, 0.9283, 1.1310, and 1.3337. The experiment was performed on 80 samples.

For the SNR values from 0 to 0.95, the weights corresponding to P have a total of 51.25% of samples. For Q, this value is 36.25%, and for R, this value is 50%. For the SNR values, above 1, the weights corresponding to P have 46.25% of samples, the weights corresponding to Q have total of 63.75% of samples, and the weights corresponding to R

quality. Thus, filtered elastography performs 2 times better than freehand elastography. Horizontal and vertical lines are plotted for the values $x = 1$ and $y = 1$. Most of the images with CNR and SNR values above 1 are found to be of good quality, whereas bad quality images fall below or on the left side of these $x = 1$ and $y = 1$ lines. There are few outliers but they fall around the horizontal ($y = 1$) and vertical ($x = 1$) lines. Regarding, the min and max values, the maximum CNR value is 5.4, and the minimum value is 0.0006. In contrast, the maximum SNR value is 3.4 and minimum is 0.5. Figure 5.10 and Fig. 5.11 indicate that the filtered elastography algorithm works better than the freehand algorithm.

Similarly, for Fig. 5.11, the plot shows CNR on the x-axis and SNR on the y-axis, with good and bad quality images marked with 1 (\times symbol) and 0 (o symbol), respectively. For freehand elastography, a total of 32% of the samples are of good quality, and the rest of the samples are of lower quality. Most of the images with CNR and SNR values above 1 are of good quality. Similar to Fig. 5.10, there are few outliers, but the outliers are close to the horizontal and vertical lines at value 1. The maximum CNR value is 6.29, and the minimum CNR value is 0.0006. In contrast, the maximum SNR value is 1.35, and the minimum SNR value is 0.37. The sample size is $N=106$. The overall results indicate that filtered elastography performs better than freehand elastography. These experiments prove our already established notion that the filtered elastography algorithm performs better than unfiltered elastography. The raw data can be found in the Appendix section in Figure 10.1-10.6. The total number of samples is $N=106$.

5.3.3 Discussion

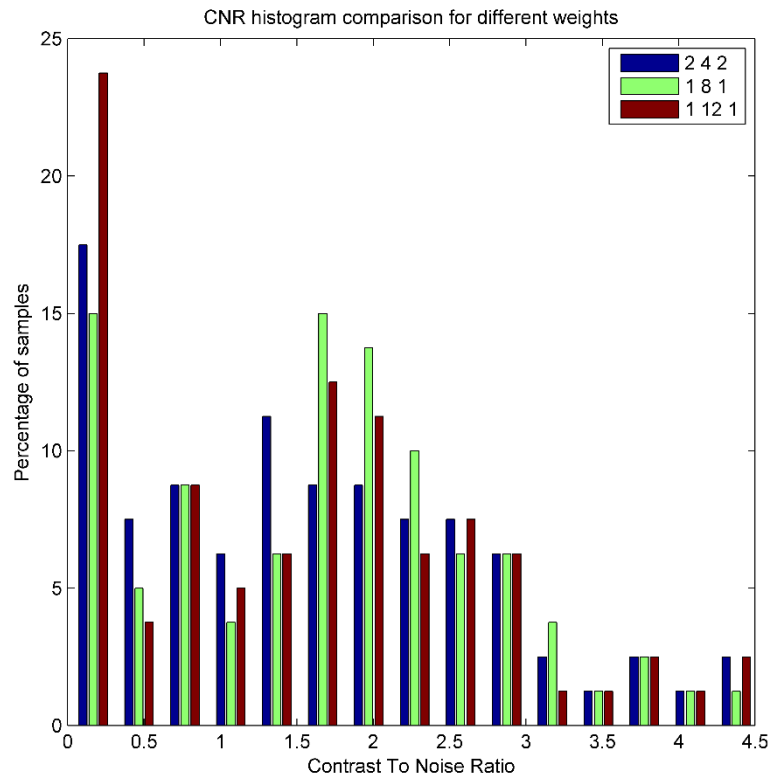


Figure 5.7 CNR histogram comparison for different weights: For CNR values in the range of 0 to 1.5, a total of 51.25% of the elastography images corresponds to the weights $\{2\ 4\ 2\}$, which is higher than the values obtained for the other weights. For the range 1.6 to 3, the weights $\{1\ 8\ 1\}$ have a total of 51.25% of the dataset fall within the range, which is higher than the values obtained for the other weights. For CNR values above 3.1, no significant difference in performance was observed. However, overall, the weights corresponding to $\{1\ 8\ 1\}$ perform better than the other weights.

In this chapter, we present a method for performing elastography on RF lines that cover the selected regions of interest and lay out experiments to prove our hypothesis. Elastography is performed for all of the RF instead of dividing the RF lines into regions of interest and then performing elastography; the latter approach would be computationally expensive because data transfer to the GPU is expensive. In addition, the elastography

computation leaves out a few samples at the top and bottom of the resulting elastography image due to the overlapping nature of the search.

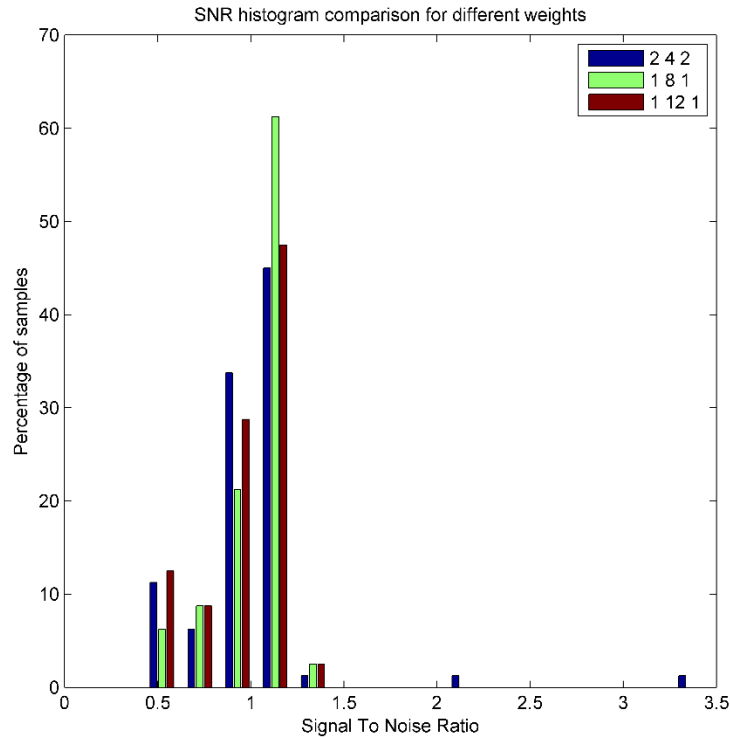


Figure 5.8 SNR histogram comparison for different weights: For the SNR range up to 0.95, the weights $\{2,4,2\}$ and $\{1,12,1\}$ have a total of $\sim 50\%$ of their elastography images in this range. The weight $\{1,8,1\}$ has a lower total percentage (36.25%) of its values in this range. For SNR values, above 1, the weights $\{2,4,2\}$ and $\{1,12,1\}$ have totals of 46.25% and 50% of the images, respectively. In contrast, the weight $\{1,8,1\}$ has a total of 63.75% of its dataset in this range. This result indicates that the weight $\{1,8,1\}$ performs better than the other weights.

This method can be extended to collect a buffer and perform permutations for the entire pool to obtain better results. After this implementation, it may be possible to compare the new algorithm directly with the O-TRuE algorithm presented in Chapter 4, as that method uses a buffer for RF data. Similarly, this buffering technique can be used to calculate 3D elastography such that the RF image data corresponding to each location of the 2D probe

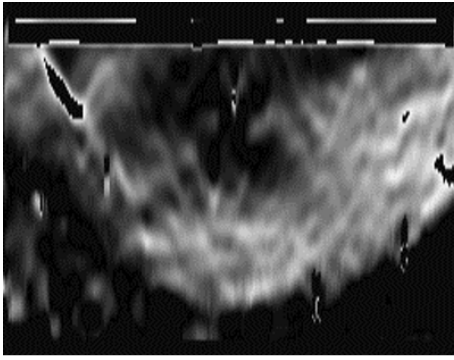
in the 4D probe results in the selection of the best elastography image generating an RF pair.

We tested this algorithm with freehand elastography to determine how much data lies in the upper and lower ranges of the CNR and SNR values. A higher range of these values indicates that the specific parameters of our algorithm result in better quality images. In contrast, a higher frequency of occurrence in the lower range of the CNR and SNR values indicates that the algorithm works less efficiently and yields more poor-quality images.

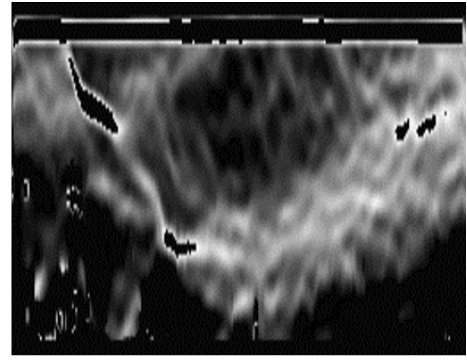
Based on the experiments presented in Section 5.3.2.1 for unfiltered and filtered elastography, for both CNR values and SNR values, filtered elastography performs better in both the high and low ranges of the spectrum of CNR and SNR values. These results are also backed with mean and standard deviation results, which were found to be extremely statistically significant for both the SNR and CNR data.

To better understand the impact of an increase in the width and height of the regions of interest, we performed the experiments in section 5.3.2.2 and found that greater width and height values perform better. This outcome is obvious given that CNR and SNR analysis with the entire elastography image as the region of interest will yield better results. One interesting observation from these results is that as the width approached 20 and 25, the results stabilized. This finding indicates that we do not need to increase the width and height of the region of interest beyond a certain limit, and low values of width 20 pixels and height 15 pixels are a good indicator of the values of the resulting elastography images. This result was reflected in the mean and standard deviation analysis. We also performed a statistical significance test. We again found that the results are better for a width of 20

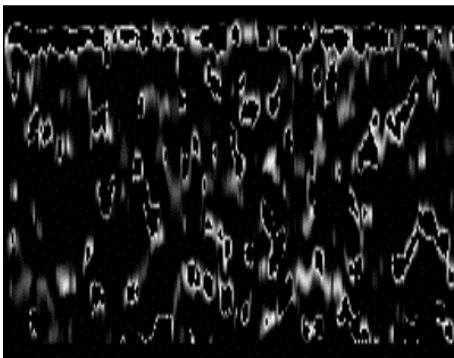
pixels and height of 15 pixels. Similarly, statistical significance was observed predominantly for results containing the group with a width of 15 pixels and a height of 20 pixels. This statistical significance result was observed for both CNR and SNR values.



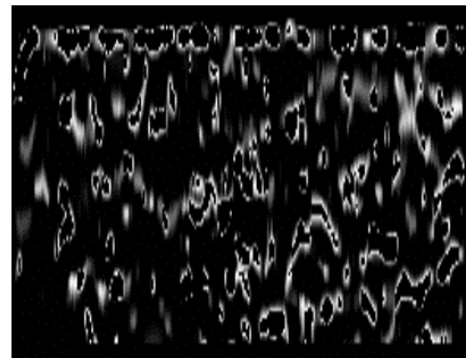
(A) CNR 1.68 SNR 1.11



(B) CNR 2.67 SNR 1.13



(C) CNR 0.22 SNR 0.49



(D) CNR 0.13 SNR 0.53

Figure 5.9 ROI image pair selection output EI images: Figure shows output EI images with varying CNR and SNR values. (A) and (B) are good quality elastography images with both CNR and SNR values above 1. (C) and (D) are poor quality elastography images with CNR values below 0.25 and SNR values below 0.6.

To better understand the weights that will yield a good result, we performed certain experiments in section 5.3.2.3 for the weights $\{1\ 1\ 1\ 2\ 4\ 2\ 1\ 1\ 1\}$, $\{1\ 1\ 1\ 1\ 8\ 1\ 1\ 1\ 1\}$, $\{1\ 1\ 1\ 1\ 12\ 1\ 1\ 1\ 1\}$, with each index corresponding to the index shown in Fig. 5.2. We found that the results for the weight $\{1\ 1\ 1\ 1\ 8\ 1\ 1\ 1\ 1\}$ were better than the results obtained for the other weights that were studied. The statistical significance and mean and standard

deviation results were in favor of {1 1 1 1 8 1 1 1 1} in the case of CNR values. However, in case of SNR values, {1 1 1 2 4 2 1 1 1} with slightly favored. In this situation, the histogram map of SNR values in Fig. 5.8 indicates that the map is symmetric {1 1 1 1 8 1 1 1 1} with an almost normal distribution. Thus, the weight {1 1 1 1 8 1 1 1 1} is recommended for future experiments and should be adopted as a standard if this method is widely used.

Figure 5.9 shows some output elastography images. The result clearly shows that the output EI has a better quality with CNR and SNR values above 1. Thus, analyzing the skewness and symmetry of the histogram hints at the performance of the filtered algorithm. The greater the number of images above CNR and SNR values of 1, the better is the performance of that particular group.

Finally, we visually classified the results as described in Section 5.3.2.4, which supports our hypothesis that the filtered elastography algorithm performs better than freehand elastography algorithms. The experiment also supports our observations that elastography images above CNR and SNR values of 1 are good quality images. Further classification results can be obtained in the future by performing more user studies.

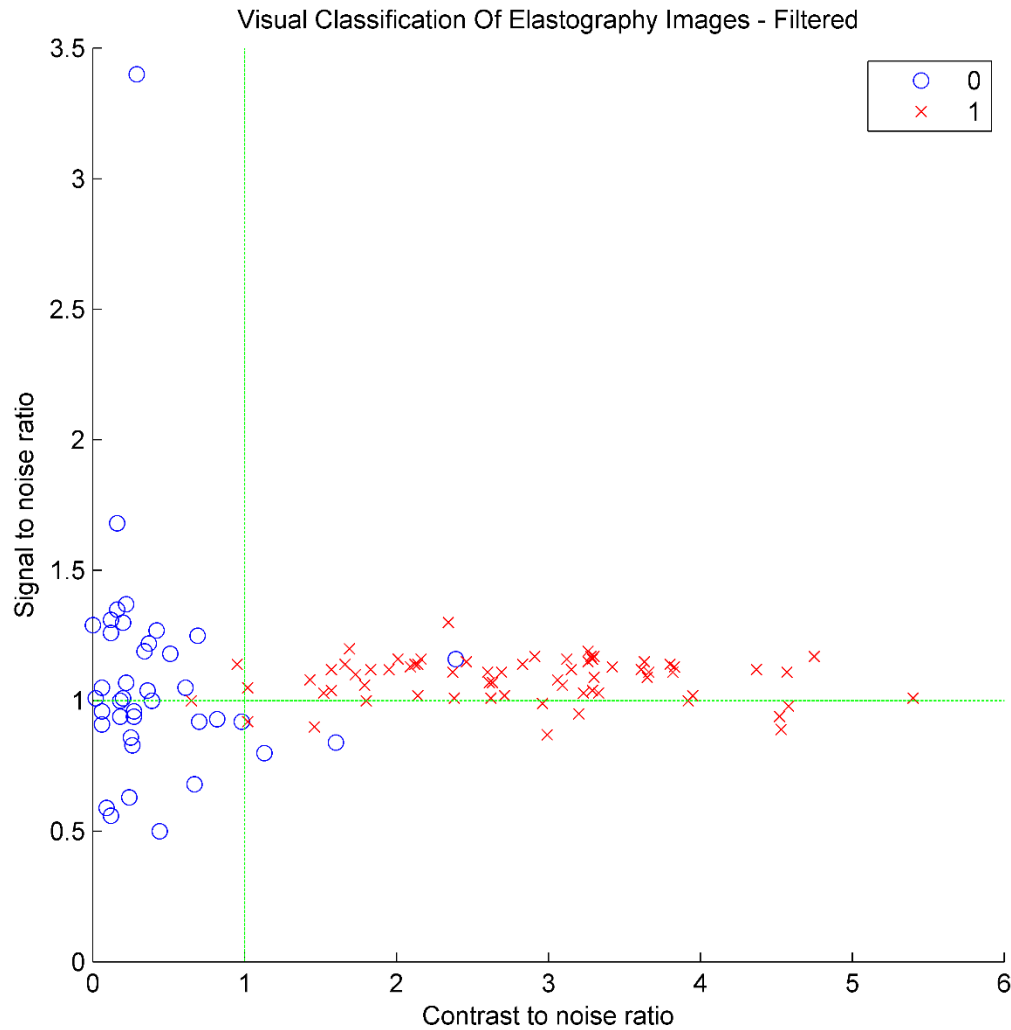


Figure 5.10 Visual classification of the filtered elastography data: In this experiment, we visually marked a good quality elastogram with 1 (\times symbol) and a bad quality elastogram with 0 (o symbol). The plot shows CNR on the x-axis and SNR on the y-axis, with each point indicating the classification of the corresponding elastography image as 0 or 1 depending on the visual quality. For filtered elastography a total of 63.2% of the images were found to be good (value 1). That result is 2 times better than the result obtained from freehand elastography. Most of the images with CNR and SNR values above 1 are found to be of good quality. The maximum CNR value is 5.4, and the minimum value is 0.0006. In contrast, the maximum SNR value is 3.4, and the minimum value is 0.5. There are few outliers, but those outliers are near the horizontal and vertical lines at value 1. This figure indicates that the filtered elastography algorithm works better than the freehand algorithm, as shown in **Fig. 5.11**.

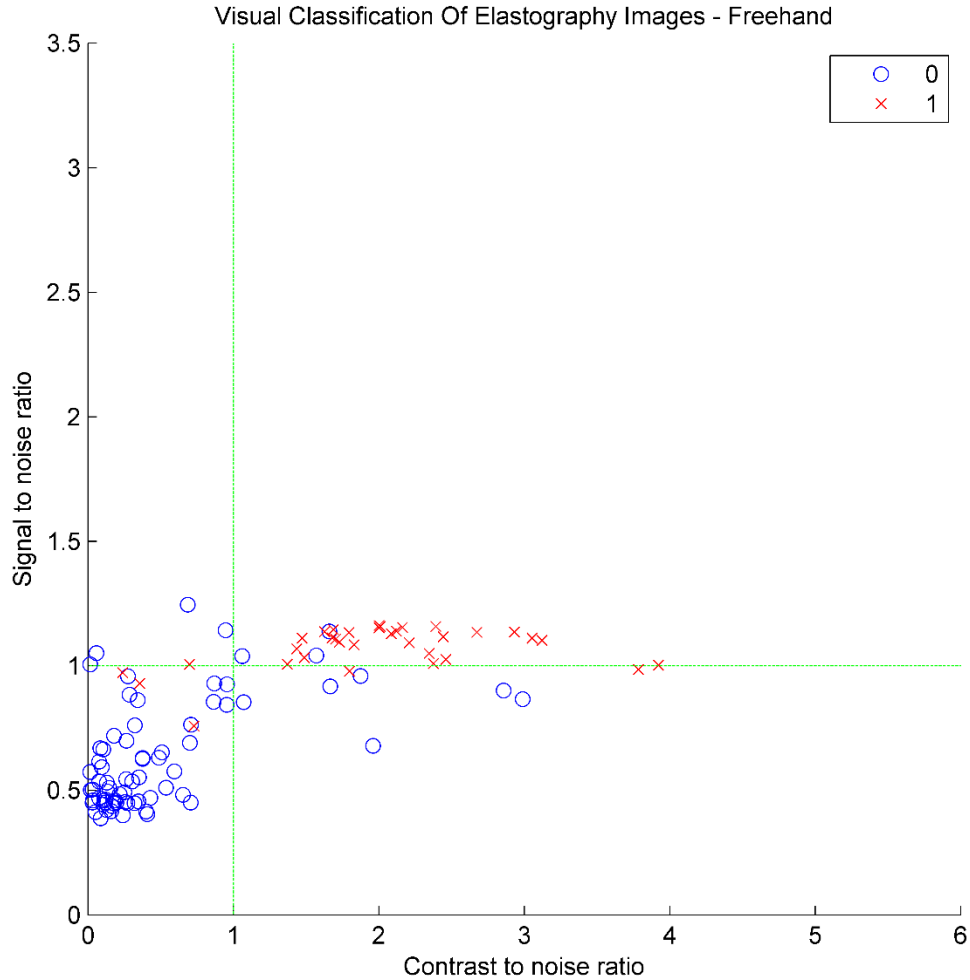


Figure 5.11 Visual classification of the unfiltered (freehand) elastography data: In this experiment, good quality elastogram is marked with 1 (\times symbol) and a bad quality elastogram is marked with 0 (o symbol). The plot shows CNR on the x-axis and SNR on the y-axis. Each elastography image is plotted at its corresponding CNR and SNR values, with 0 or 1 to indicate a good or bad quality image. For freehand elastography, a total of 32% of the samples were found to be good quality (\times symbol). Most of the images with CNR and SNR values above 1 are found to be of good quality. The maximum CNR value is 6.29, and the minimum CNR value is 0.0006. In contrast, the maximum SNR value is 1.35, and the minimum SNR value is 0.37. Similar to **Fig. 5.10**, there are few outliers, but the outliers are close to the horizontal and vertical lines at value 1. The sample size is $N=106$. The overall results indicate that filtered elastography performs better than freehand elastography.

6 Five-Dimensional Ultrasound System for Soft Tissue Visualization

This chapter is from [90] and published with the following citation:

Nishikant P. Deshmukh, Jesus Caban, Russell H. Taylor, Gregory D. Hager, and Emad M. Boctor, "Five dimensional Ultrasound Visualization Framework to Guide and Monitor Ablative Therapy," IPCAI/CARS/IJCARS, 2015, doi:10.1007/s11548-015-1277-z . © CARS 2015. Published with permission of Springer.

In this chapter, we present, to the best of our knowledge, the first implementation of a real-time 5D US system based on the fusion of 3D B-mode and 3D USE data updated over time. The existing 3D US systems display B-mode and elastography volumes in separate windows, which make them difficult to view, synchronize, and tag 3D B-mode data with strain information. A combined 3D B-mode and elastography visualized over time in 5D US solves this problem. The absence of real-time computation hardware limited the development of essential components of 5D US system such as a sophisticated real-time elastography software, real-time 3D scan conversion, and a real-time visualization system. The contributions reported in this paper include GPU-based real-time 3D elastography using a multi-stream technique, GPU-based real-time 3D volume scan conversion, and a real-time volume renderer that updates every time the 3D USE, and 3D B-mode volume is received over the network using the OpenIGTLinkMusiiic library [85][23]. The definition of a real-time computing system is a system in which all of the components combine to finish the task in a given time [91]. In our case, we define the upper bound as any system that allows a surgeon to operate the system during the real-time thermal ablation

monitoring that takes several minutes to perform [72]. A response time of 5-10 s would be sufficient for the 5D US system given that the two RF data volume capture times are in the order of 4-6 s (estimate based on 2D acquisition speed [85]). Through this paper, we attempt to answer the following questions: (1) Is it possible to achieve a real-time 5D US system? and (2) What should we expect the computation time to be?

Traditional general-purpose graphic processing units (GPGPUs) have a single instruction multiple data (SIMD) architecture. The SIMD allowed only one compute-unified data architecture (CUDA) kernel to execute on the GPGPU at a given time. With the advent of Fermi architecture [92], and the multiple instruction multiple data (MIMD) processors, modern GPGPUs now support multiple instructions and multiple data stream capability of facilitating the execution of multiple kernels simultaneously. We use the multi-stream capability of these GPGPUs to give us a faster 3D USE by extending our work in 2D elastography as described in [2]. Scan conversion is essential to convert the data acquired in the polar coordinates by a 3D wobbler probe into the Cartesian coordinates [93]. This chapter presents a simple GPGPU-based scan conversion to convert 3D B-mode and 3D USE data in real time. The 3D USE data have been visualized using a volume rendering technique [25], [94], [95], but it lacks a real-time data receiver to refresh the volume data. This paper extends the volume renderer in [25] to receive the data in real time.

The 3D B-mode and USE encounter noise artifacts like speckles and de-correlation-induced artifacts. A multi-dimensional transfer function [25] is used to reduce this effect to map B-mode and USE values as well as different color and opacity values. This transfer function gives 5D US the ability to visualize the surface of the tumor and inner-outer

surface of the cyst with a particular color and opacity from the USE data. Concurrently, the contour of the organ and biopsy/ablation needle can be visualized using different color and opacity values derived from the B-mode data. Thus, it is necessary to fuse together the 3D B-mode and 3D USE values to complement each other. In addition to a feasibility study about achieving these updates in real time, this paper also presents a validation study on the size of an elastography phantom lesion. While a 5D system based on Doppler ultrasound for monitoring the heart to visualize blood flow already exists, it requires specialized Doppler hardware [96].

Individual contributions

Author initials: Nishikant P. Deshmukh (**NPD**), Jesus Caban (**JC**), Russell H. Taylor (**RHT**), Gregory D. Hager (**GDH**), and Emad M. Boctor (**EMB**).

Author contributions in the published version [3]:

Following are the contributions from authors only in the published version [3] of this chapter (there are additional contributions by NPD):

1. Overall system integration: NPD
2. Multistream GPU elastography for 3D volume algorithm development: NPD
3. Online tracked ultrasound elastography (O-TRuE) development: NPD
4. Elastography image stream analysis method: NPD.
5. Conceived and designed the experiments: NPD JC RHT GDH EMB.
6. Performed the experiments: NPD.
7. Analyzed the data: NPD.

8. Contributed reagents/materials/analysis tools: NPD.
9. Wrote the paper: NPD JC RHT GDH EMB.

Systems contributions

Generating 3D elastography volumes is a challenging problem given that the sequentiality and one-to-one correspondence for the comparison of two RF volumes must be maintained. Additionally, the multi-threading and combining of this 3D volume data is a challenging task. In this chapter, I contributed a 3D elastography volume generation algorithm on a GPU that uses a queueing mechanism coupled with multi-threading for efficient calculation. The technique involves creating multiple threads to compute each slice of the 3D elastography volume and assigning the last slice (or any slice) as an aggregator. The challenge lies in ensuring that the next volume does not interfere with the threads created for the previous volume. Therefore, we created a thread structure to handle such scenarios. This approach leads to a minimal wait time for the system because the threads on the CPU and GPU run asynchronously to transfer data during processing. The other method developed is the 5D US system pipeline, that is based on an event-driven mechanism to receive elastography data, receive b-mode data, calculate a 3D scan conversion, select a transfer function, and perform volume rendering. The event-driven mechanism helps to reduce polling and save CPU cycles. Correctly designing such a system is a challenge because the normal workflow of a sequential system is nonexistent and may often lead to a race condition if designed incorrectly. The methods section and Pseudocode 6.1-6.5 detail the approach used.

Summary

Purpose A five-dimensional ultrasound (US) system is proposed as a real-time pipeline involving fusion of 3D B-mode data with the 3D ultrasound elastography (USE) data as well as visualization of these fused data and a real-time update capability over time for each consecutive scan. 3D B-mode data assist in visualizing the anatomy of the target organ, and 3D elastography data adds strain information.

Methods We investigate the feasibility of such a system and show that an end-to-end real-time system, from acquisition to visualization, can be developed. We present a system that consists of (a) a real-time 3D elastography algorithm based on a normalized cross-correlation (NCC) computation on a GPU; (b) real-time 3D B-mode acquisition and network transfer; (c) scan conversion of 3D elastography and B-mode volumes (if acquired by 4D wobbler probe); and (d) visualization software that fuses, visualizes, and updates 3D B-mode and 3D elastography data in real time.

Results We achieved a speed improvement of 4.45-fold for the threaded version of the NCC-based 3D USE versus the non-threaded version. The maximum speed was 79 volumes/s for 3D scan conversion. In a phantom, we validated the dimensions of a 2.2-cm-diameter sphere scan-converted to B-mode volume. Also, we validated the 5D US system visualization transfer function and detected 1- and 2-cm spherical objects (phantom lesion). Finally, we applied the system to a phantom consisting of three lesions to delineate the lesions from the surrounding background regions of the phantom.

Conclusion A 5D US system is achievable with real-time performance. We can distinguish between hard and soft areas in a phantom using the transfer functions.

6.1 Introduction

In 2013, 1,660,290 cancer incidences and 580,350 cancer deaths were reported in the USA [97]. Ultrasound elastography (USE) is an imaging tool often used in the diagnosis and treatment of cancer [1]. USE involves comparing pre- and post-compression ultrasound data to map tissue stiffness [1]. A wide array of applications have emerged using USE such as image-guided thermal ablation monitoring [72], neoadjuvant therapy monitoring [98], and intraoperative robotic surgery guidance [82][83].

US technology is safe, low cost, mobile, and real-time, and it emits zero ionizing radiation [72]. However, 2D US has problems tracking because of the absence of elevation information in the image [72]. The introduction of 3D transducers [72] that can give information about all three orthogonal planes alleviates this problem. USE combined with B-mode scan acts as a complementary technology to reduce the need for unnecessary biopsies [99]. 2D/3D B-mode image lacks the ability to differentiate between the background tissue and the *iso-echoic* tumors because the speckle information and impedance is the same in tumors and surrounding tissues [100]. The 2D/3D USE systems calculate the displacement of speckles in the radio frequency (RF) data and generate a strain map; this strain map helps to identify the boundary between the tumor and the surrounding tissues providing more accurate information [1]. 3D USE has been used for

diagnosis, monitoring and treatment of breast cancer [101][102], testicular adrenal rest tumors [103], cervical lymph nodes [104], and ablation monitoring [72][69][105].

3D B-mode/USE imaging does not reveal temporal changes in the underlying organ [99], but 4D US with scanning 3D B-mode images over time at the same or different locations does offer real-time, continuous feedback [99]. 4D US has been used in advanced in vivo studies of fetal face expression [106], determination of heart aorta elasticity [107], pelvic floor muscle monitoring [108][109], fetal heart monitoring [110], and motion tracking of the liver [111], among other applications. 4D volume contrast imaging (VCI) uses contrast enhancing measures to either elasticity data [112] or color Doppler data [113]. USE has been used in conjunction with B-mode images to increase specificity and maintain the sensitivity of tumor detection [99]. Automatic segmentation of the prostate has been achieved by the fusion and then extraction of contour boundary information from both B-mode and vibro-elastography images [114].

Ablation therapy involves detecting the tumor in a preoperative CT/MRI scan and then manually registering the location of the tumor intraoperatively using the US B-mode image guidance [72]. Ablation therapy involves inserting an ablation needle under this B-mode image guidance, and ablating the target tumor using RF ablation or high-intensity focused ultrasound (HIFU) [72]. However, there is a risk of incorrect tumor detection and needle placement. Furthermore, the treatment may not ablate the entire tumor area or may extend to affect surrounding healthy tissues [72]. To minimize this risk, 2D/3D USE-guided ablation therapy can offer precise locations of the tumor and the ablated region [72]. However, it is difficult for surgeons to monitor two different feedbacks while performing

the surgery. There is a need for overlaid mechanisms to precisely locate needle position. This requirement also exists with a US-guided biopsy to accurately monitor needle placement to extract malignant cells. The feedback needs to be fast enough to allow the surgeon to monitor this process in real time. During thermal ablation monitoring, the ablation process self-induces noise in the collection of ultrasound RF data; therefore, the time window to collect US data is quite small. The system should be fast and efficient enough to collect these data and visualize them over time to increase the accuracy of thermal ablation. Controlling the transfer function would allow distinction between USE and B-mode data because the contour of the organ will encapsulate the lesion/tumor inside it. Finally, fast scan conversion is needed to convert both the USE and B-mode volumes for the 3D wobbler probe that acquires RF data along a spherical sector.

A 5D US system involves the fusion of 3D B-mode images and 3D USE data visualized over time. The live feedback of both 3D B-mode and 3D USE data will improve the early detection and treatment of cancer. 3D B-mode provides information about *hyper-* and *hypo-echoic* tumors [100]. Apart from enhancing the boundary of the *hyper-* and *hypo-echoic* tumors, 3D USE can give information about the *iso-echoic* tumors [100]. The additional strain information from 3D USE offers more diagnostic and monitoring information in terms of shape, size, and position of the lesion [72]. In the 5D US, a multi-dimensional transfer function allows efficient segregation of the B-mode and USE data in a single texture volume. This segregation will allow for advanced segmentation where we can isolate a tumor, cyst, and organ contour information depending on the transfer function as well as a future multi-modality registration [100]. The B-mode modality can be potentially replaced with an advanced US time series tissue typing methods [115], [116]. However, in

this paper, we focus on the more ubiquitously used B-mode modality for wider acceptance and clinical trials. Several reports [117], [118] indicate that both specificity and sensitivity is very high considering the relative size of the lesion in strain to that of the B-mode. Malignant tumors tend to infiltrate into surrounding healthy tissue that necessitates real-time visualization with an adapted opacity function to show the extent of strain and B-mode of an all-in-one, 3D-rendered scene.

6.1.1 General-purpose graphics processing unit (GPGPU)

A GPGPU contains many computation cores that parallel run the similar code. Many of the components such as filters, cross-correlation, and scan conversion can independently work on subsets of data. A GPGPU allows parallelization of such components and offsets the workload from the primary CPU, which allows additional processing for the foreground CPU. Many workstations, including the ones embedded in the US systems, have onboard GPGPU and also allow the addition of extra GPGPUs via PCI Express slots [2]. The current Ultrasonix (Richmond, BC, Canada) ultrasound machine used in our experiments can acquire RF data at ~ 100 frames per second (fps) [23]. In advanced techniques like parallel beam forming where envelope detection occurs via onboard hardware, the acquisition rate has increased to nearly 860 fps [119], necessitating a real-time GPGPU-based architecture. The design choice of the architecture gives us the flexibility to run various components on the same machine or different machines depending on resource availability. Apart from that, a GPGPU frees the main CPU of the US machine to do other essential tasks; this decreases slowdown of the US system. A GPGPU will frequently be referred to as simply a GPU in the remainder of this paper.

6.1.2 Normalized cross-correlation (NCC)-based elastography on GPU

NCC helps track speckle movement when the palpation motion displaces the tissue. We assume that the direction of palpation motion is parallel to the axial direction of the RF images obtained from the ultrasound acquisition system. We calculate displacement along the axial direction by selecting a template window in a pre-compression RF image and the source window in a post-compression RF image. The template window is searched in the source window to estimate displacement using cosine curve fitting [72]. The outliers are corrected by median and averaging filters. Strain estimation uses linear regression [1]. The following equation defines the NCC score:

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x, y) - \overline{f_{u,v}}][t(x-u, y-v) - \overline{t}]}{\left\{ \sum_{x,y} [f(x, y) - \overline{f_{u,v}}]^2 \sum_{x,y} [t(x-u, y-v) - \overline{t}]^2 \right\}^{0.5}} \quad (1)$$

where $f(x, y)$ is the search window that is searched in the template window $t(x, y)$ for the displacement u and v in the x (axial) and y (lateral) directions, respectively. The variables $\overline{f_{u,v}}$ and \overline{t} are the mean of the search and template window, respectively.

Each NCC window comparison can be computed efficiently on the GPU because there is inherent data independence in the processing. This data independence explicitly reduces the need to synchronize the elastography image computation. Similarly, the median filter, average filter, and strain estimation are independent for every pixel calculation and can be efficiently parallelized on the GPU [89].

6.2 Methods

This section details the data flow diagram of the real-time 5D US with its subcomponents: 3D USE, 3D B-mode acquisition, 3D scan conversion, and visualization modules. First, we explain the overall system diagram followed by the implementation of 3D USE, 3D scan conversion, and the visualizer. The most common type of 3D probe is a wobbler probe, which has a 2D US array mounted on a motor and is attached to a rotating motor shaft. The 4D probe creates a scan depicting a spherical sector along the elevational direction. The 3D B-mode and USE data need to be scan-converted to reflect the correct shape of the underlying objects.

6.2.1 Five-dimensional ultrasound system

Figure 6.1 and Algorithm 6.1 show the overall system diagram and steps needed to create the 5D ultrasound system. The system is highly modular, and each component can run on the same machine or different machines. The OpenIGTLinkMusiic library [23][85] helps us to achieve this modularity. The RF server resides on a US machine and helps to collect real-time 3D RF data and 3D B-mode data. These 3D RF data are sent to USE/EI (elastography image) server, and the 3D B-mode data are dispatched to a 5D US visualizer. The hardware synchronizes the 3D RF data and the 3D B-mode data. Thus, it is not necessary to register them separately. After an image pair is received by the 3D EI server, it computes 3D USE to give 3D EI data for 5D US visualization.

If we visualize 3D B-mode and 3D EI as stored in the memory, then the data will appear as a rectangle. However, in our case, a 4D wobbler probe collects the data. This probe requires a 3D scan conversion module to convert the rectangular coordinates to polar coordinates depicting a spherical sector. Thus, a GPU-based scan conversion module is internally embedded inside the 5D visualizer. The scan conversion module sends scan-converted 3D B-mode and 3D EI volume to the OpenGL shading language buffer. We provide the user with an interactive transfer function mapper to draw 2D transfer function. The transfer function assists the ray tracer to assign a color value to the 3D EI and 3D B-mode voxels. The ray tracer module fuses each voxel from the two 3D datasets and displays them on the screen.

Algorithm 6.1: Five-dimensional ultrasound system

- 1: The RF server collects the US machine-generated 3D RF data and 3D B-mode data and transfers them to the elastography (EI) server and 5D US system, respectively
 - 2: The EI server generates 3D USE/EI data using the GPU by calculating the estimated displacement per slice in the two volumes of the RF data
 - 3: The 3D EI data are sent to the 5D US system
 - 4: 5D US system receives 3D B-mode and EI data volumes that are scan-converted by a GPU to give a fast volumetric scan conversion
 - 5: The scan-converted 3D volumes of 3D B-mode and 3D USE/EI are then passed to the OpenGL shader on the GPU that registers the two volumes
 - 6: The ray tracer highlights the color values for pixels in the buffer depending on the transfer function selection by the user
 - 7: New data overwrite the buffer, and the pixels are updated as per the transfer function values
-

6.2.2 Multi-threaded 3D ultrasound elastography

Figure 6.2 and Algorithm 6.2 detail how we accelerate 3D USE. An earlier version of the GPU-based 3D EI server ran on a single operating system thread [72]. A thread controls

the entire pipeline of the GPU and consists of tasks such as input/output (I/O), memory allocation, kernel function invocation, wait for kernel function invocations, and memory de-allocation [2]. Due to the SIMD architecture of the earlier version of the GPU, this thread could not be executed in parallel because they would wait for the thread to complete their task [2]. In our version, we bind this pipeline together with the help of the CUDA stream functionality similar to a previous report [2]. The CUDA streams bind the entire GPU pipeline for elastography consisting of several kernels such as displacement estimation by NCC, moving the average filter, median filter, and strain [2]. These CUDA streams are then assigned to separate threads. The CUDA maintains data independence between streams that implies that the threads do not interfere with each other and provide a robust implementation. The number of threads that create n matches is the number of slices in a scan. The n threads then execute in parallel. If the thread ID starts with 1, we assign a thread with an ID equal to n as an accumulator. The accumulator thread waits for the other $(n-1)$ threads in a batch to complete execution of elastography for their respective assigned pair in the RF frame (slice) in consecutive RF volumes. After execution is complete, the n threads store their elastography image into a commonly shared buffer with an index as the thread ID. The accumulator thread with ID n after finishing the wait for $(n-1)$ thread accesses the shared buffer and creates an OpenIGTLink message for volume consisting of n elastography image slices. This thread then sends these data over a TCP/IP network using an OpenIGTLinkMusic thread [85][23].

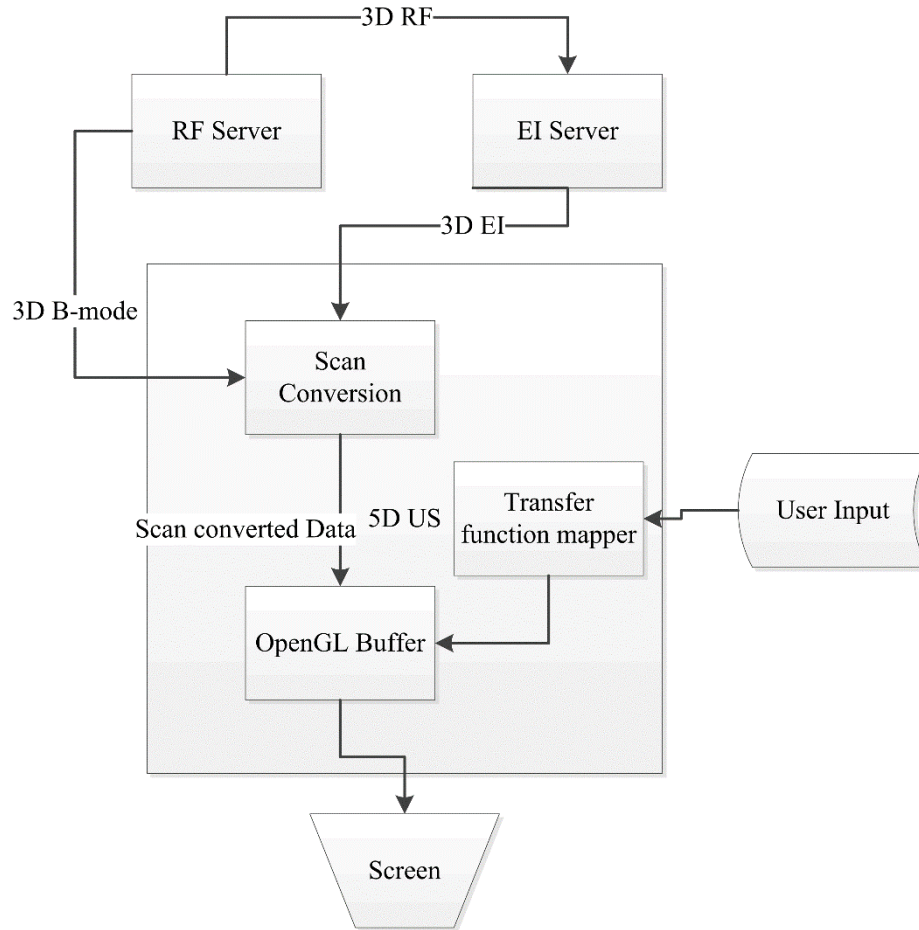


Figure 6.1 Data flow diagram of 5D US: the RF server collects the 3D data using a wobbler probe that performs a sector scan using a 2D probe in a particular field of view. The 3D RF data are passed to the elastography image (EI) server, which calculates the 3D USE data and passes them to the scan conversion module in the 5D US system. The RF server also sends the 3D B-mode data directly to the scan conversion module of the 5D US system. The 3D B-mode and the 3D USE scan-converted data are then passed to the visualization system. The users select the transfer function values to highlight different areas of the volume with different colors.

6.2.3 3D Scan conversion

In the case of 4D wobbler probes, the 2D transducer moves along a spherical sector with a fixed step angle. If the number of slices collected is n , then the field of view of the 3D scan is equal to n times the step angle. However, the 3D image data are stored in the

memory as a 2D image per slice in consecutive memory locations. A source transmits these 3D image data over the network. We performed scan conversion from rectangular coordinates to polar coordinates to convert these image data to its correct spherical sector shape from the rectangular form when accumulated together. For efficient and real-time implementation, we map these 3D image data on the GPU and perform scan conversion. This visualizer embeds the scan converter within its structure because of its small execution time. A detailed explanation of the 3D scan conversion is in “Appendix.”

Algorithm 6.2: Multi-threaded 3D Ultrasound Elastography

- 1: Receive 3D RF data from the RF server with n slices
 - 2: *Spawn* n threads to calculate 2D elastography on the GPU. The components are connected by a CUDA stream to protect the data among the threads
 - 3: The GPU threads execute in parallel calculating strain images and scan conversion according to the depth
 - 4: The n^{th} thread, at the point of its completion, waits for the $n-1$ thread to complete its task and waits on the join command
 - 5: All $n-1$ threads deposit their data in the shared buffer, which is combined with the n^{th} thread. This volume of size n slices is sent over the network
-

6.2.4 Five-dimensional ultrasound visualizer transfer function

The y axis corresponds to the grayscale intensities for B-mode images, and the x axis corresponds to the grayscale intensities for the strain (elastography) images. The user can draw various shapes such as an ellipse or rectangle of different sizes. The program then

creates a Gaussian circle with a radius as the maximum dimension of the ellipse or rectangle that, in turn, defines the transfer function. These Gaussian circles have centers equal to the center of the corresponding ellipse or rectangle drawn by the user. This transfer function maps different intensity values of B-mode and strain images. The user can also change the opacity of each ellipse or rectangle and create the corresponding transfer function. During the ray tracing, each pixel value is indexed into this transfer function, and the corresponding color and opacity value is assigned to that pixel during the accumulation process of the ray tracer.

This ray tracer then displays the combined pixel intensity along a given line in the perspective view. This process is repeated for each line path until the entire volume is complete. The user can zoom in and out. The user can change the position, color, and opacity of each ellipse or rectangle, and the program updates the corresponding 3D volume with new values. This highly dynamic visualization system may help in searching for different features such as strain, needle, and tissue boundaries as well as any foreign objects in the volume depending on the dataset.

Pseudocode 6.1 3DRFDataReceiver: This module receives the 3D RF data from the RF server or from a file and transmits it to the EI processing engine via a queueing mechanism. It also receives back the used RF data images to be freed in the memory.

Function 3DRFDataReceiver Input: 3D RF data from the RF server in OpenIGTLinkMusiiC format. Output: 3D RF data in a local structure format easily usable by different EI modules.
3DRFDataReceiver i := 0 While(true) Sleep on OpenIGTLinkMuSiiC receiver for data over the network Receive 3D RF(i) data image from OpenIGTLinkMuSiiC SmartPointer Allocate GPU CUDA direct memory access (DMA) memory M(i) Transfer 3D RF(i) to M(i) Insert 3D RF(i) into RFInputQueue(x) - Send Wakeup to listening processes Test if RFOutputQueue(y) has any returning data from EI server (Here we do not sleep to avoid race condition) If (y == dataPresent) Free (y) with special CUDA call Remove reference to OpenIGTLinkMuSiiC SmartPointer end i := i + 1 LoopbackToWhile
EndFunction 3DRFDataReceiver

Pseudocode 6.2 3DEIProcessor: This module receives the 3DRF data from the 3DRFDataReceiver and processes them to compute elastography images and transmit them over the network. This is the basic version without the CPU threading model.

<p>Function 3DEIProcessor Input: 3D RF data from the 3DRFDataReceiver in simple local structure format. Output: 3D RF volume pairs and 3D Elastography volume. 3D RF volume sent back to 3DRFDataReceiver for recycling.</p>
<pre> i := 0 j := 0 Sleep on RFInputQueue(x) to receive 3D RF(i) i := i + 1 Create a strainArray of size M While (true) Sleep on RFInputQueue(x) to receive 3D RF(i) When RFInputQueue receives data - Wakeup RF (i) = x i := i + 1 Allocate EI_parameters_structure for GPU computation (This structure contains all parameters necessary for EI computation) for j = 0 to M strainArray(j) = nonThreaded3DEIComputation (RF(i-1), RF(i) , EI_parameters_structure, j) endFor if (clientConnected) send strainArray over the network endIf LoopbackToWhile </pre>
<p>EndFunction 3DEIProcessor</p>

Pseudocode 6.3 3DEIPProcessorAdvanced: This module receives the 3D RF data from the 3DRFDataReceiver and processes them to compute elastography images and transmit them over the network. This is the advanced threaded version with the CPU threading model.

<p>Function 3DEIPProcessorAdvanced Input: 3D RF data from the 3DRFDataReceiver in simple local structure format. Output: 3D RF volume pairs (Elastography sent via threaded3DEIComputation). 3D RF volume sent back to 3DRFDataReceiver for recycling.</p>
<pre> i := 0 j := 0 Sleep on RFInputQueue(x) to receive 3D RF(i) i := i + 1 Create a double array threadArray that acts as a Queue (maxSize: N x M) (M is size of the volume) While (true) Sleep on RFInputQueue(x) to receive 3D RF(i) When RFInputQueue receives data - Wakeup 3D_RF_structure contains RF(i-1) and RF(i) volumes m := 0; foreach RFpair in 3D_RF_structure (Here RFpair is a structure contains RF pair from different volumes) Allocate EI_parameters_structure for GPU computation (This structure contains all parameters necessary for EI computation) threadArray(j, m) = threaded3DEIComputation (RFpair(0), RFpair(1) , EI_parameters_structure, j, m, M) m := m + 1 endfor for k = 0 to N for n = 0 to X Test if threadArray(k, n) exited if threadArray(k) volume exited then free the memory and reuse Send the 3D RF data (i-1) to RFOutputQueue(y) endFor n endFor k j := mod((j + 1), N); LoopbackToWhile </pre>
<p>EndFunction 3DEIPProcessorAdvanced</p>

Pseudocode 6.4 threaded3DEIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography. Note, the threaded version calculates the elastography for individual RF image pairs but the last M^{th} thread waits and collects rest of the threads to construct a volume and send it over the network.

<p>threaded3DEIComputation Input: RF image pairs although the Mth thread combines the given set as one volume of size M elastography images. Here n means any value greater than or equal to 0. n and n+1 simply indicates two different RF data images. Output: Elastography image volume computed for the given RF data volume pairs and the RF data pairs.</p>
<pre> Receive RFn and RFn+1 If (modulus(threadID, M) == 0) First thread, hence allocate memory to hold 3DStrainMap (size: M) endIf (correlationMap, displacementMap) = ComputeEI (RFn, RFn+1) if (average(correlationMap) < 0) (correlationMap, displacementMap) = ComputeEI (RFn+1, RFn) endIf displacementMapRefined = movingAverage(medianFiltering(displacementMap)) StrainMap = strainEstimation (displacementMapRefined) if (modulus(threadID, X) != X-1) store strainMap in 3DStrainMap (threadID % M); else memset(3DStrainMap, 0) for i := 0 to M-1 wait for threadIDX(i) to complete endFor store strainMap in 3DStrainMap if (clientConnected && modulus(threadID, X) == X) send 3DStrainMap on the network endIf endIf return RFn and RFn+1 </pre>
<p>EndFunction threaded3DEIComputation</p>

Pseudocode 6.5 nonThreaded3DEIComputation: This module calculates the elastography. If the average correlationMap is less than 0 it means that the comparison was done in wrong direction where it recomputes elastography. Note, the non-threaded version simply computes elastography for only one pair and the calling function creates the volume.

<p>nonThreaded3DEIComputation Input: RF image pairs. Here n means any value greater than or equal to 0. n and n+1 simply indicates two different RF data images. Output: Elastography image computed for the given RF data pairs and the RF data pairs.</p>
<pre> Receive RF(n) and RFn+1 (correlationMap, displacementMap) = ComputeEI (RFn, RFn+1) if (average(correlationMap) < 0) (correlationMap, displacementMap) = ComputeEI (RFn, RFn+1) endif displacementMapRefined = movingAverage(medianFiltering(displacementMap)) strainMap = strainEstimation (displacementMapRefined) if (clientConnected) send strainMap on network endif Free (strainMap, displacementMap, displacementMapRefined, correlationMap); return strainMap, RFn and RFn+1 </pre>
<p>EndFunction nonThreaded3DEIComputation</p>

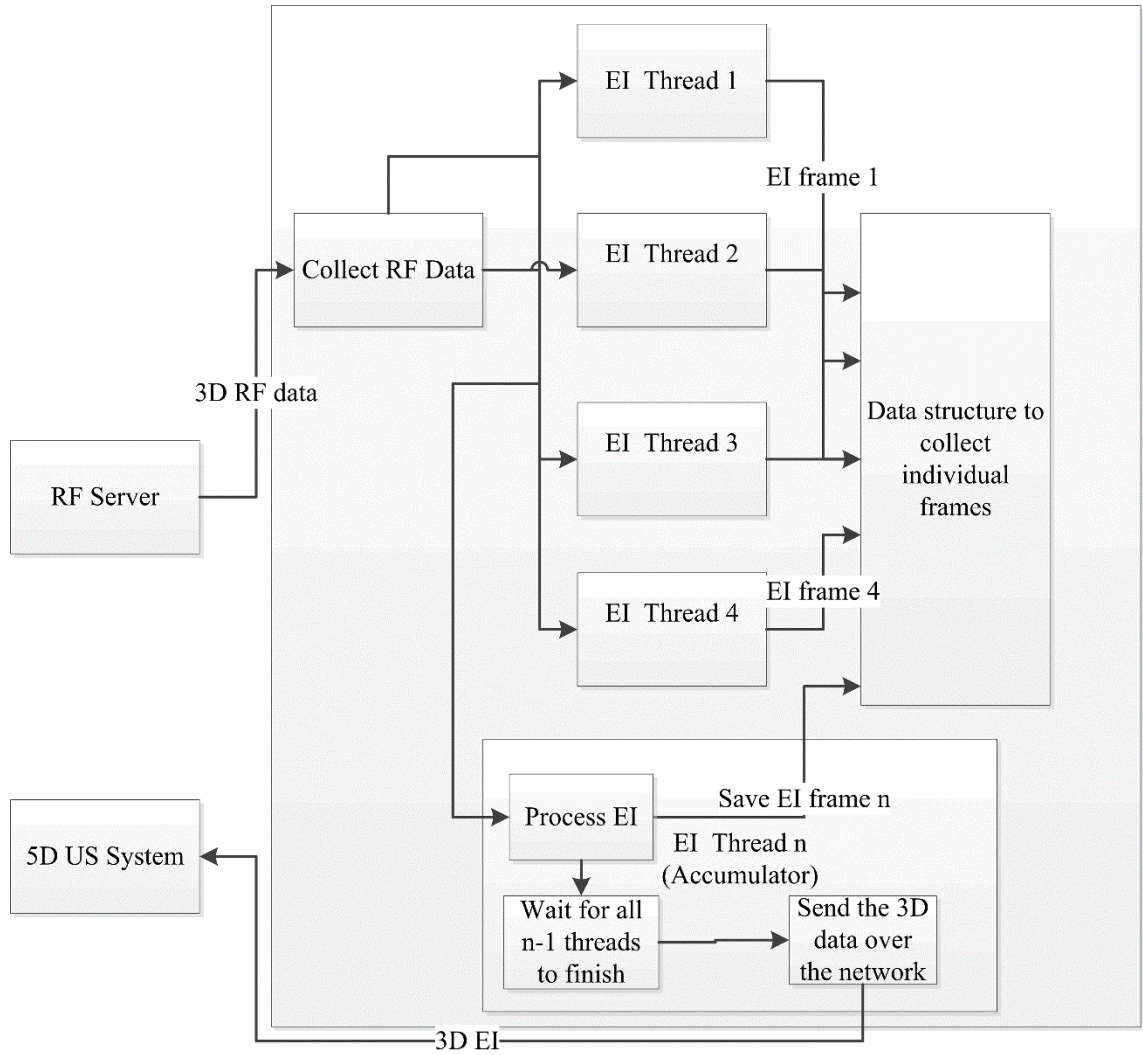


Figure 6.2 Data flow diagram of 5D US: a block diagram of the 3D USE system that collects the data. The process is distributed among the elastography image (EI) thread to calculate each slice independently. These slices are then collected in the accumulator thread that waits for the remaining threads to finish their task and then sends these USE data as one volume to the 5D US system.

6.3 Experiments

This section details the experimental setup, planning, and expected outcomes regarding the experiments and results. We evaluated 3D USE timing information and 5D US visualization on a system with Intel Xeon CPU 2.13 GHz, 12 GB RAM, Windows 7 - 64

b, NVidia C2070 GPU. We measured the 3D scan conversion timing information on a computer with Intel i7 3.2 GHz, Windows XP 64 b, and an NVidia C1060 GPU. The probe used to collect the 3D RF data is Ultrasonix 4DL14-5/38 attached to the Ultrasonix-CEP machine.

We used a CIRS Elasticity 049A QA phantom with background elasticity of 33 kPa with lesions of varying elasticity. We validated the 5D US system with a 1- and 2-cm lesion. The elasticity of the lesions is 58 and 39 kPa. While scanning the phantom surface, we generated the timing diagram of 3D USE as well as a scan conversion on the 3D B-mode data. The phantom setup is shown in Fig. 6.3 (A). The setup used to validate the scan conversion module is shown in Fig. 6.3 (B), (C).

An offline phantom study has been reported previously [25]. By offline, we mean that the data is collected using a staged robot with a 2D transducer. The stage robot performs a palpation and sends a signal to the US machine after each pre- and post-compression motion. The US machine then collects the corresponding RF data. The B-mode and elastography images in this experiment are calculated offline in Matlab because the purpose of this experiment is to show the difference between B-mode, elastography, and fused volumes as reported in Fig. 6.10. For both the 2D and 4D transducer, the number of pixels in the lateral direction of the RF data remains constant at 128 for a 4-cm 2D transducer. The number of pixels in the axial direction changes as a function of depth—this is 1024 pixels for 4 cm, 1296 pixels for 5 cm, 1552 pixels for 6 cm, 1808 pixels for 7 cm, and 2064 for 8 cm of imaging depth. The voxels per pixel are 1. The size of each pixel in the RF data is 16 b.

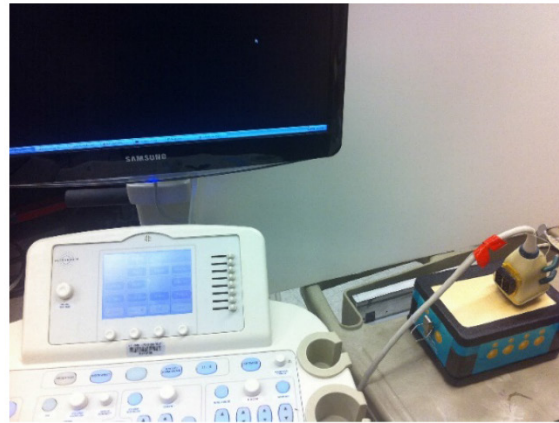
6.4 Results

We next compare the two 3D USE implementations to understand the advantages of *threaded* versus *non-threaded* 3D USE application. We set the timer just before 3D RF data are passed to 3D USE processing engine, and the program stops the timer just after the USE frames are generated. All of the threads finish the execution. In the case of single-threaded 3D USE application, only one thread calculates the USE for all the slices. Thus, we set the timer after the calculation of the whole volume is completed. Similarly, the speed of the 3D scan conversion is calculated after the thread that calculates the scan conversion has finished.

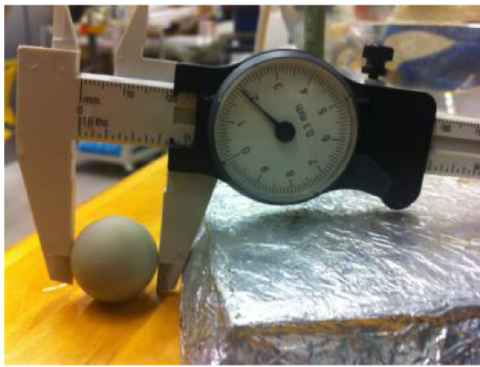
As shown in Fig. 6.4, the *non-threaded* version of the GPU-based 3D elastography (simply referred as *non-threaded version*) could achieve a maximum averaged runtime of 3.39 s/volume at 4 cm of imaging depth. The volume contains 30 slices. The corresponding average *maximum runtime* for threaded GPU-based 3D elastography (referred to simply as *threaded version*) is 0.12 s/volume at 4 cm depth. The average execution time increased gradually with an increase in depth because the number of samples increases as a function of depth. The minimum average runtime recorded for the *non-threaded version* is 6.06 s/volume when collected at 8 cm of imaging depth. We observed that the *standard deviation* for the timing value was minimal for the threaded version. This indicates a stable execution time where the runtime of generation of volumes remains constant. The maximum average *standard deviation* in the case of the *threaded version* is 0.041 s/volume at 7 cm of imaging depth. The minimum average *standard deviation* in the case of the *non-threaded version* is 0.122 s/volume at 4 cm of imaging depth.

This result suggests that the *threaded version* performs better in execution time and stability than the *non-threaded version* of the GPU-based 3D elastography. For 5 cm imaging depth, the threaded version recorded the lowest *standard deviation* of 0.015 s and the *non-threaded version* recorded the second lowest *standard deviation* of 0.17 s. The small standard deviation corresponds to an imaging depth of 5 cm—this indicates that 5 cm of imaging depth is an ideal imaging depth for both cases. It results in a stable stream of volumes. Please note that the results are average timing values calculated for just above 200 volumes. The window size for this result of 3D NCC is 12 with 2 mm maximum forward search along the axial direction and 98 % window overlap. As listed in Table 6.1, the maximum throughput of 84643.02 kB/s is obtained for imaging depth of 8 cm for a threaded version of elastography. This is slightly better than the average throughput of 80471.80 kB/s for a non-threaded version of the 3D elastography.

In Fig. 6.5, we further investigated the impact of different window sizes (8, 10, 12, and 14) on the runtime of the 3D NCC volume with constant 2 mm maximum forward search in the axial direction and 98 % overlap. (Please note that the quality of the resulting elastography images changes according to the window size as reported in [2].) Window size is equal to the number of samples in the axial direction used for comparison of source and target images. The runtime of the 3D NCC elastography is best for window size 8; the *minimum time* is 1.45 s/volume. In terms of average time for all imaging depths, the average *seconds/volume* runtime for window size 8 is 2.16 (± 0.13), window size 10 is 2.33 (± 0.16), window size 12 is 2.45 (± 0.13), and window size 14 is 2.62 (± 0.14). Here, we can see that window size 10 is slightly better than window size 12 and 14, whereas the time is taken for window size 8 is clearly much lower.



(A)



(B)



(C)

Figure 6.3 Experimental setup: this figure shows the experimental setup for our experiments. (A) Experimental phantom setup where a 4D probe is held by a passive arm on top of CIRS Elasticity 049A QA phantom. (B), (C) Experimental setup to validate scan conversion of 2.2-cm sphere under water.

The 3D scan converter is needed to reconstruct the geometry for 4D wobbler probes where the 2D array is moving around a fixed axis of rotation along a spherical sector with a limited field of view. As shown in Table 6.2, the maximum speed observed is 79.40 volumes/s for 8 cm depth for 31 frames per volume. The lowest speed is 13.81 volumes/s for 120 frames per volume at 4 cm depth. This result corresponds to the scan conversion of B-mode volumes. The speed increases with increasing depth in B-mode volume because to adjust the aspect ratio on the display screen the US machine reduces the number of pixels in the lateral direction with increasing depth. Thus, the effective size of the volume

decreases with an increase in depth. This leads to an increase in speed of scan conversion as the imaging depth increases. We validated whether the scan conversion correctly converts the dimensions by imaging a solid sphere of 2.2 cm diameter underwater with B-mode. As shown in Fig. 6.6, the dimension of the sphere in all three views is found to be approximately 2.2 cm to the scale.

Figure 6.7 shows the effect of choosing different opacity values for ellipse B. As shown in Fig. 6.7 (A), the opacity value is set to 3, but it is 50 in Fig. 6.7(B) and 100 in Fig. 6.7 (C). This shows that region B (indicated by blue arrow) corresponding to ellipse B changes the opacity in the output. The two ellipses indicated by label A have constant opacity, and the lesions are constant in the output (arrow A in the output). The opacity value of 50 for ellipse B in Fig. 6.7 (B) reduces the effect of unneeded noise and B-mode data by increasing the transparency. The unneeded noise and B-mode corresponding to ellipse B are further diluted by reducing the opacity to 3. This is indicated by label B in the output. This result suggests that we can keep the strain or elastography values visible and at the same time draw additional circles to illuminate features, objects, and tissues that might be useful in the corresponding B-mode images. Figure 6.8 shows a fusion of 3D B-mode and strain values where one ellipse highlights maximum possible dynamic range in 3D B-mode and 3D strain values. The primary strain locations are shown by arrow/label A; B-mode values are indicated by label B.

In Fig. 6.9, we investigated whether we can differentiate between hard and soft lesions. In this case, the soft tissue/lesion has an elasticity value of 39 kPa, and the harder tissue/lesion has an elasticity value of 58 kPa. As seen in the transfer function, ellipse A corresponds to the hard lesion, and ellipse B corresponds to the soft lesion. From the output,

we can see that label B clearly shows a softer lesion. There is an overlap of colors between label A and label B due to the small difference in elasticity between lesions. More rigorous evaluation of different strain strength will be done in future studies.

Figure 6.10 shows the results of a phantom experiment where three lesions are surrounded by background material. As shown in the Fig. 6.10 (A) B-mode volume, the lesion is slightly visible. In Fig. 6.10 (B), an elastography volume is displayed, and the lesion is clearly visible. In Fig. 6.10 (C) the fused B-mode and elastography volume are displayed. The selected transfer functions display the lesion region in green and the background regions by blue and pink. This verifies our claim that the volume fusion can improve the feedback of the underlying parameters in B-mode and elastography volumes.

6.5 Discussion

We demonstrated the feasibility of a 5D US system by implementing, evaluating, and validating each component of the system. Our highly modular system led to a 5D US system. This end-to-end system facilitated data acquisition from a US machine that was distributed to various elements. The timing graphs in Figs. 6.4 and 6.5 demonstrate that our new 3D elastography algorithm is fast and stable. 3D elastography is computationally expensive, and reducing the execution time of a volume comparison was the first step. The threaded elastography that we implemented outperformed the non-threaded elastography version by a factor of $4.45 \times$. The 3D scan converter on a GPU gave a maximum of 79 volumes/s. This high-speed and small standard deviation ensured that we could embed the scan conversion module inside the 5D visualizer. The multi-stream 3D elastography is stable and executes on the same computer where the visualizer was running. Because the

GPU performs most workload on the GPU, the US system can also execute all parts if equipped with a GPU. Thus, implementation of a 5D US system in the operating room is a practical solution.

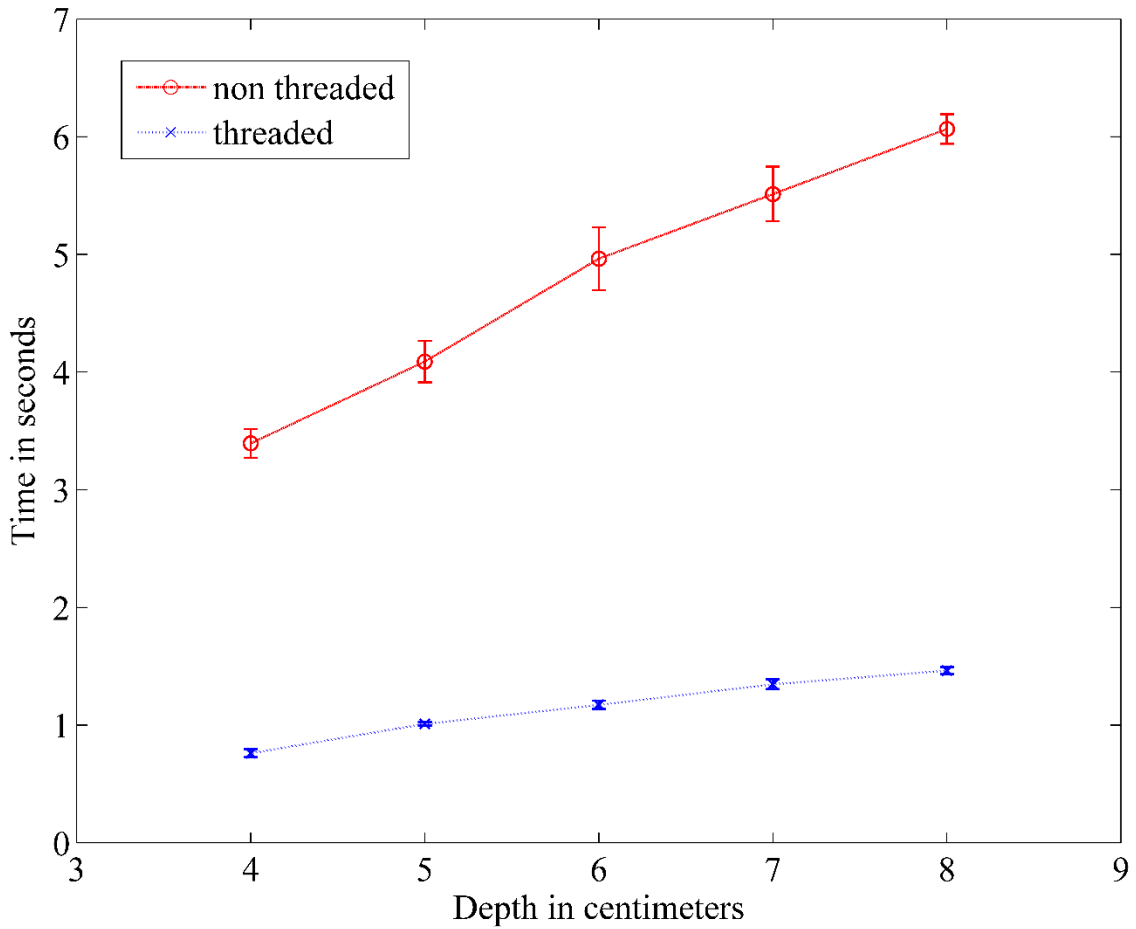


Figure 6.4 Performance of non-threaded versus threaded 3D EI: this figure shows the comparison of time taken to compute a volume of elastography for threaded versus non-threaded version to determine whether our algorithm led to any improvement in performance. As indicated in this graph, a threaded version led to a $4.45 \times$ improvement in speed. This proves that our algorithm has managed to keep the EI volume generation to just below 1 s for 4 cm imaging depth (approximately 0.75 s/volume) and 1.46 s/volume for 8 cm imaging depth.

3D USE can be improved by externally tracking the ultrasound probe with an optical tracking system. The tracked 3D USE can be an extension of online tracked USE [2]. The multi-stream 3D USE introduced here can easily accommodate tracking information, and

each multi-stream thread can then find in-plane slices among the slices across multiple scans allocated to it. The transfer function can be improvised in multiple ways to study and determine the multi-dimensional transfer functions best suited for the underlying tissue conditions.

Table 6.1 Throughput of elastography algorithm: this table lists the average throughput of the input RF data volumes for real-time elastography. The values are multiplied by 2 because generating an elastography image volume needs a pair of RF data. As can be seen, the 8 cm imaging depth for threaded 3D EI gives maximum average throughput of 84,643 kB/s. The max throughput obtained is 87273 kB/s. The lowest average throughput is 18103 kB/s for 4 cm depth with the minimum throughput value of 15519 kB/s. The sample size is on an average of approximately 145 volume computations.

Imaging Depth	Volume size (kB)	Throughput (kB/s)	
		Non-threaded	Threaded
4 cm	30,720	18,103 (\pm 607)	80,717 (\pm 3124)
5 cm	38,880	19,017 (\pm 742)	76,973 (\pm 1112)
6 cm	46,560	18,763 (\pm 967)	79,461 (\pm 2201)
7 cm	54,240	19,679 (\pm 776)	80,563 (\pm 2282)
8 cm	61,920	20,419 (\pm 403)	84,643 (\pm 1689)

Table 6.2 Speed of 3D scan conversion for a B-mode volume: the table lists the speed of real-time 3D scan conversion in *volumes per second* (vps). The columns are arranged per number of frames in each volume or *frames per volume* (fpv). The highest average speed is observed for 8 cm imaging depth at 79.40 vps for 31 fpv with max of 81.42 vps. The slowest 13.81 vps is observed for 120 fpv at 4 cm imaging depth with minimum of 13.62 vps. The result is computed on sample of 15 volume set to get time to compute per volume scan conversion in milliseconds, from this we calculated volumes per second since the standard deviation is low.

	31 fpv (volumes/s)	60 fpv (volumes/s)	90 fpv (volumes/s)	120 fpv (volumes/s)
4 cm	52.45 (\pm 2.29)	25.82 (\pm 0.40)	19.30 (\pm 0.26)	13.81 (\pm 0.09)
5 cm	60.00 (\pm 2.33)	33.33 (\pm 0.39)	22.91 (\pm 0.55)	16.18 (\pm 0.27)
6 cm	66.14 (\pm 2.33)	38.42 (\pm 1.55)	25.99 (\pm 0.16)	17.75 (\pm 0.31)
7 cm	72.44 (\pm 3.97)	40.65 (\pm 0.35)	27.77 (\pm 0.61)	16.41 (\pm 0.26)
8 cm	79.40 (\pm 2.62)	46.18 (\pm 0.87)	30.89 (\pm 0.31)	22.50 (\pm 0.11)

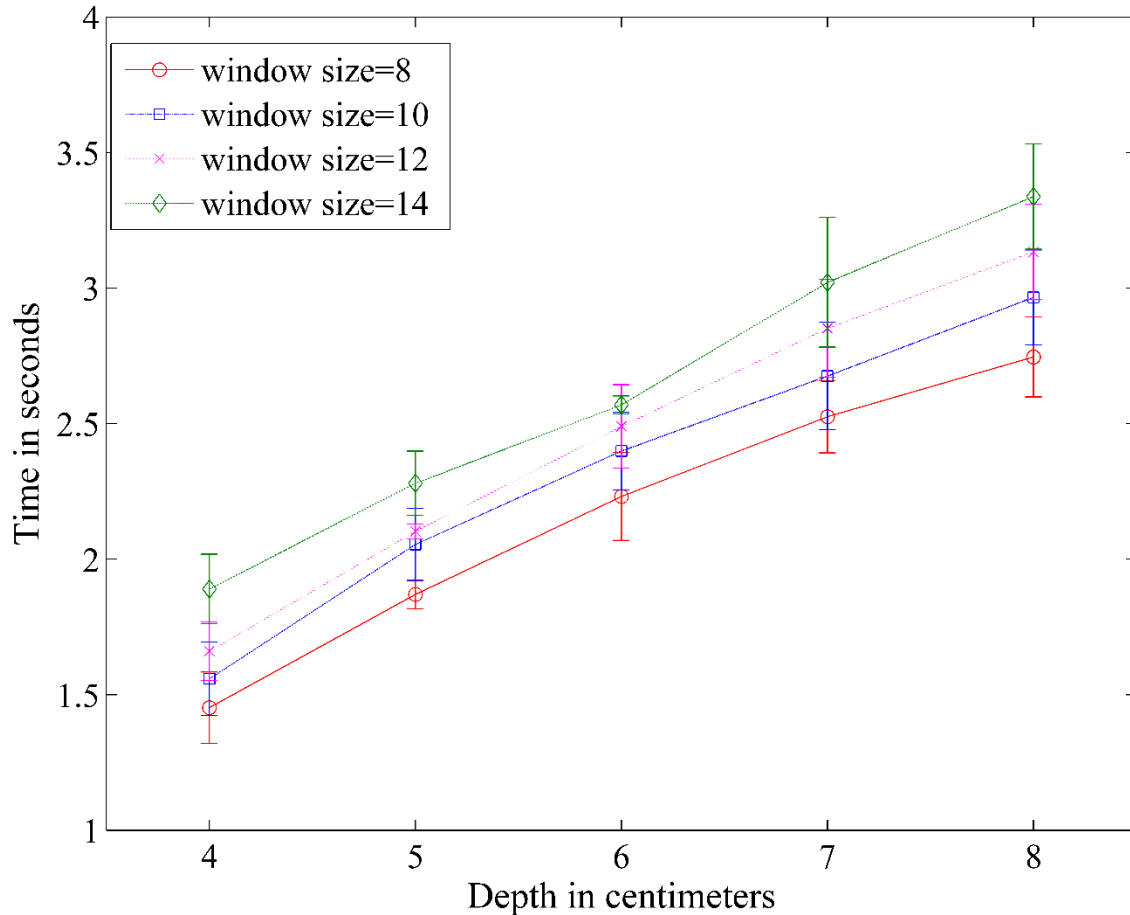


Figure 6.5 Performance of non-threaded 3D EI for different window sizes of NCC: this figure shows comparison between different window sizes of 3D NCC. The forward image search is restricted to maximum 2 mm, and window overlap is kept constant at 98 %. It has been observed that the runtime in general decreases with decreasing window size. The average standard deviation (0.13 s/volume) is lowest for window size 8 and 12 and highest (0.16 s/volume) for window size 10. This indicates that the speed for window size 8 is stable and faster than other window sizes. A difference in standard deviation of 0.03 s/volume is notable because the fastest runtime is 1.45 s/volume. The average sample size is 235 volumes. The maximum average time obtained is 3.337 s/volume for window size 8 and depth of 8 cm indicating slowest speed, and the maximum runtime is 4.45 s/volume. The minimum average runtime obtained is 1.45 s/volume for window size 8 and depth of 4 cm, while the minimum runtime is 1.347 s/volume.

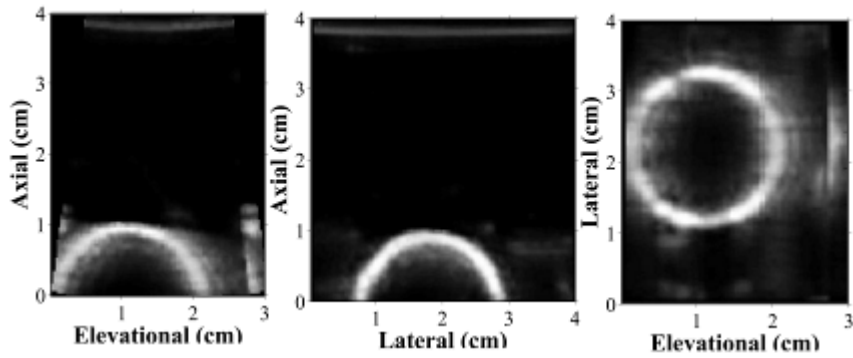


Figure 6.6 Validation of size after scan conversion: validation of size of an object after scan conversion is performed by imaging a 2.2-cm-diameter sphere inside a water tank. The scan-converted output is shown in the images with an approximate diameter of 2.2 cm in all the three views.

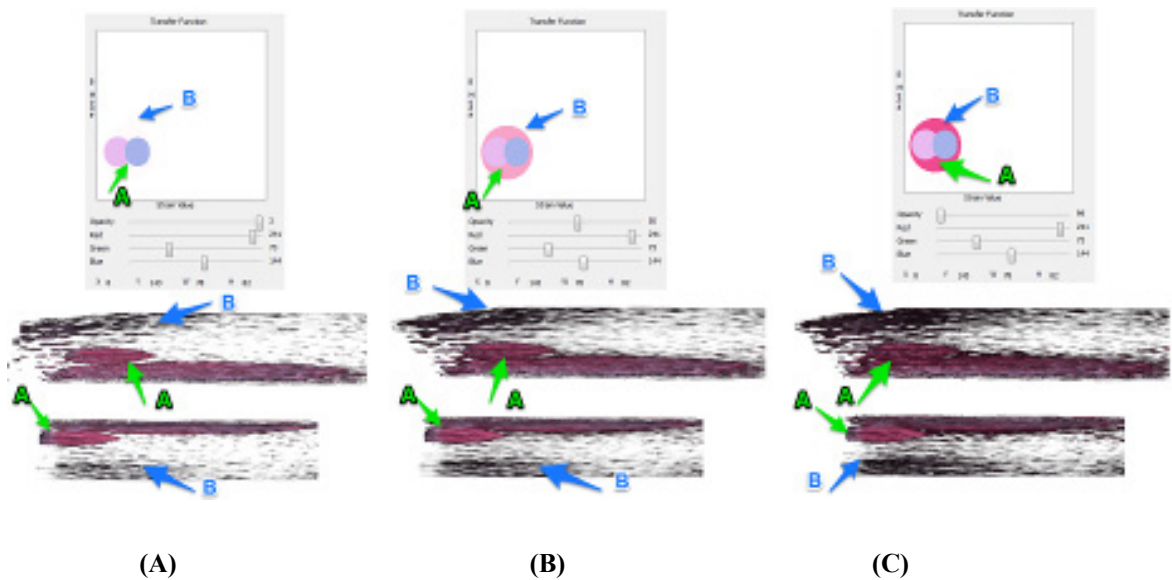


Figure 6.7 Impact of changing opacity values: *ellipse A* indicates the region of high strain value where the lesion is found; *ellipse B* indicates the region around *ellipse A*. The opacity value for (A) is set at 3; in (B), it is set at 50; in (C), it is set at 100. The *arrow* in the output section indicates the corresponding regions highlighted by each *ellipse*. As shown in (A), the low opacity value reduces unneeded noise and B-mode values while the lesion indicated by *arrow A* remains visible.

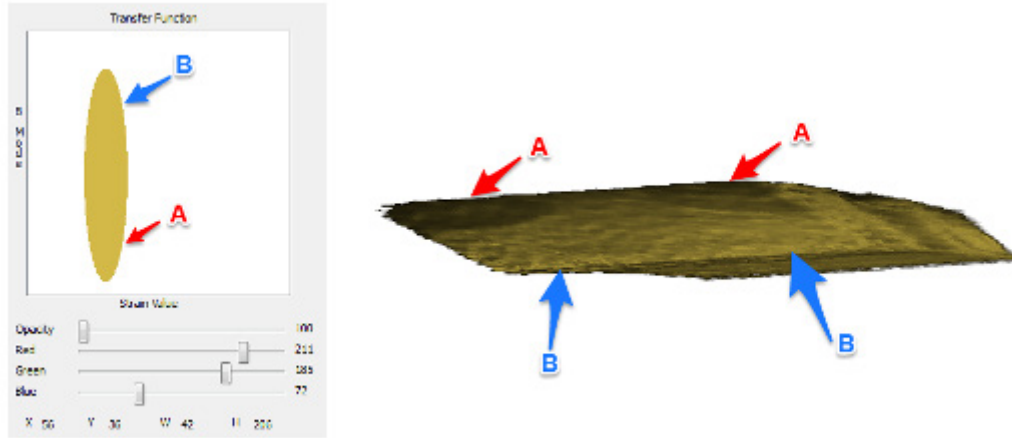


Figure 6.8 B-mode and strain volume fused together: region *A* on the transfer function map emphasizes the hyper-echoic region as a spherical region in the output. The rest of the B-mode values are in the surrounding region of the lesion.

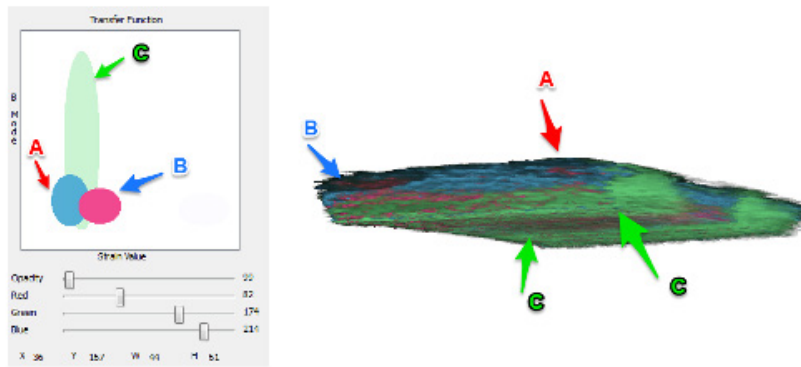
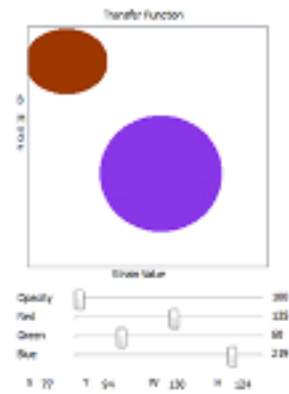
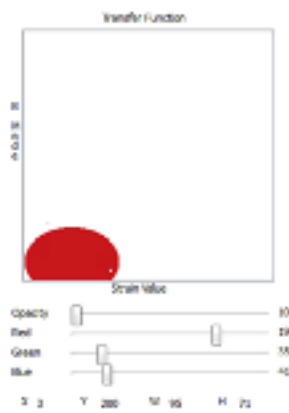
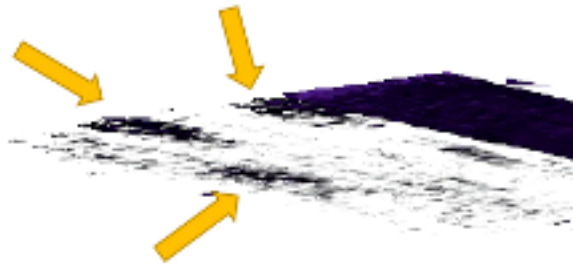


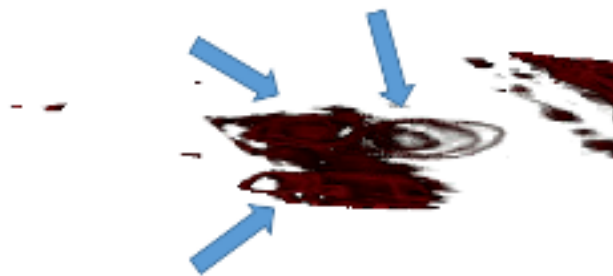
Figure 6.9 Differentiating hard and soft lesion: the soft lesion (elasticity 39 kPa) is highlighted with pink color (*ellipse B*) and hard lesion (elasticity 58 kPa) is highlighted with blue color (*ellipse A*). We can differentiate the soft lesion (label *B*) from the hard lesion (label *A*) with a subtle difference. There is an overlap of colors where the soft lesion is partially green due to the small difference in elasticity of the lesions.



(A) B-mode volume



(B) Elastography volume



(C) B-mode and Elastography fused volume

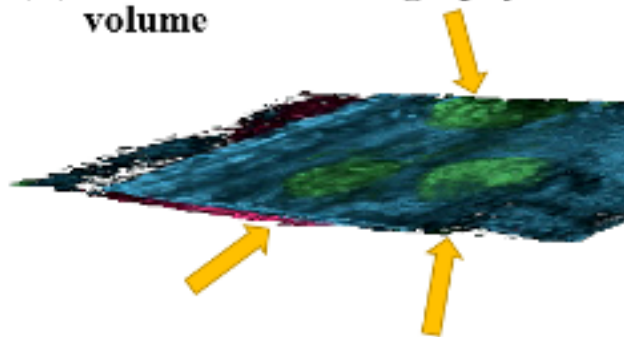


Figure 6.10 Differentiating B-mode, elastography and fused volume: this figure shows volume rendering with different input data types. (A) shows only B-mode volume to display contour of the lesion, (B) shows only elastography volume to show the ablated region, and (C) shows both B-mode and elastography fused volume. In (C) the lesion region is displayed as green region surrounded by background in blue and pink.

7 Conclusions and Future Work

7.1 Conclusions

Chapter 4, that was originally published separately [2], presents a multi-stream GPU-based implementation of elastography, specifically demonstrating how recent advancements in GPU hardware may be harnessed to achieve much higher frame rates than previously possible. Our system achieves an approximately 2.13X improvement over a conventional GPU-based NCC elastography implementation and produces a maximum frame rate of 78 fps, nearly matching the acquisition rate of typical ultrasound systems.

We demonstrated the versatility of our architecture by implementing an online version of tracked ultrasound elastography (O-TRuE), by performing *in vivo* animal experiments and by integrating the system with the da Vinci Surgical System. We devised a method to evaluate the quality of an output elastography stream based on the maximum correlation of a windowed region defined in consecutive elastography frames. Using this metric, we demonstrated that O-TRuE (with tracked free-hand elastography) produces a more stable output stream than untracked free-hand elastography. A comparative study was performed to assess the effect of the NCC window size on the elastography frame rate and the image quality. With the *in vivo* pig data, the optimal NCC window size of 10 provided a speed of 48 fps with a CNR of 3.57 and an SNR of 0.94. The *in vivo* animal experiment using untracked elastography demonstrated better contrast for ablated regions in the elastography images in comparison to the corresponding B-mode US images. Integration with the da

Vinci system investigated the effect of the palpation frequency and amplitude on the elastography image quality with an elastography phantom and, found a stable output stream using values of 5Hz and 3mm, respectively. These experiments demonstrate the practical feasibility of using GPUs for intra-operative real-time navigation and monitoring. In this chapter, I have developed the multi-stream GPU based elastography algorithm, integrated tracked ultrasound elastography in real-time elastography to give Online tracked ultrasound elastography (O-TRuE), and worked on integrating the real-time EI with the da Vinci surgical systems. Additionally, I developed a technique to analyze and quantify real-time elastography algorithms.

In Chapter 5, we presented a method for detecting a high-quality elastography image without the need to perform elastography on the entire RF data image. The results indicate that our method performs better than unfiltered or freehand elastography. The CNR and SNR values of the resulting elastography images indicate that filtered elastography using our method gives a higher frequency of occurrence of images in the higher spectrum of the respective values. The result was found to be extremely statistically significant with $p\text{-value} < 0.0001$ for both t-Test and ANOVA test. Experiments that investigated different sizes for the region of interest indicate that as the width (w) and height (h) approach 25 and 15 pixels, respectively, the resulting quality of the elastography images stabilizes. We prefer the width and height of 20 and 15 pixels since the results are similar to that of $w=25$ and $h=15$ pixels. This was found from the mean and standard deviation results and pairwise group $p\text{-value}$ (< 0.05) statistical significance results. From the different weight experiments, we found that the weight of the central region of interest should be 8 and the weight of the rest of the region of interest should be 1 to generate high-quality elastography

images. This was found to be true with the histogram analysis after observing the frequency distribution of various CNR and SNR values. The mean and standard deviation results along with group pairwise p-value (<0.05) for CNR indicated that the central weight of 8 is a better choice. Although the test failed for SNR analysis. The possible reason is low sampling size and mean of all groups very close to each other. We also performed visual classification of the unfiltered and filtered data. We found that images above CNR and SNR value 1 are predominantly good quality images. I have designed and implemented this algorithm for real-time determination of the quality of future elastography image by calculating elastography on RF lines encapsulating selected regions of interest using GPU. I also analyzed the data and validated it with animal experiments and statistical analysis.

In Chapter 6, that was originally published separately [3], we demonstrated the first known implementation of a five-dimensional ultrasound system that involves an end-to-end system for the acquisition of 3D RF and B-mode data. We transferred the data over the network, calculated the GPU-based 3D elastography, performed aGPU-based scan conversion of volumes and visualized the images using multi-dimensional transfer functions. The GPU-based multi-threaded 3D elastography gave us 4.45-fold better performance versus single-threaded GPU-based 3D elastography. We achieved a maximum speed of 79 B-mode volumes/s for 3D scan conversion. We validated the size and shape of the 3D B-mode scan conversion output for a 2.2-cm spherical ball—in 3D elastography, we were able to visualize the 1- and 2-cm-diameter phantom lesion. We then distinguished between the lesion and the surrounding tissue in the phantom using transfer functions. I have designed and implemented using GPUs the world's first known

implementation of the five-dimensional ultrasound system. I have collected and analyzed the data.

7.2 Future Work

The multistream real-time elastography method presented in Chapter 4 is essentially a 1D correlation search. This 1D correlation search can be extended to a 2D correlation search to obtain a more accurate displacement estimation. With this 2D correlation search, the NCC-based elastography on the GPU can be used to estimate displacement in the axial and lateral directions. This displacement estimation can be used to initialize the analytical minimization and dynamic programming (AM/DP) elastography [57] method's initial search range. The limited area of interest method from Chapter 5 can also be used to decide whether a complete AM/DP elastography method must be calculated.

The current da Vinci robotic palpation software does not communicate directly with the elastography system. A feedback loop can be established, where the elastography system receives the frequency and amplitude of palpation information from the palpation software. At the same time, the elastography system can also send information about the current strain information to the palpation software. Experiments can be performed to determine how this feedback mechanism enhances the quality of the elastography images.

The method outlined in Chapter 5 for determining elastography quality via an image-based search uses a simple averaging technique. We can extend this method to a machine learning-based approach, where the information from the different regions of interest acts as the input for the learning algorithms. Several high-quality strain data sets can be

evaluated. The variables in the form of the average correlation and SNR of the region of interest and the final SNR, CNR and correlation average can be given as an input to the learning algorithm. This method currently compares consecutive RF images that may not give the best results because two in-plane RF image pairs may be three to four images apart. We must extend this work to select high-quality images from a pool of images.

The 5D US system must be validated with human data. The implementation of tracked ultrasound 3D USE and the corresponding 3D B-mode systems will increase the accuracy and stability of the 5D US. The method described in Chapter 5 can also be extended to obtain a stable 3D elastography volume. With the advent of virtual reality (VR) systems, the 5D US can be easily extended to the VR systems to allow the surgeon can see the layout of the patient's body and organs along with the 5D data.

8 Bibliography

- [1] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: A quantitative method for imaging the elasticity of biological tissues,” *Ultrason. Imaging*, vol. 13, no. 2, pp. 111–134, Apr. 1991.
- [2] N. P. Deshmukh, H. J. Kang, S. D. Billings, R. H. Taylor, G. D. Hager, and E. M. Boctor, “Elastography using Multi-Stream GPU: An Application to Online Tracked Ultrasound Elastography, In-vivo and the da Vinci Surgical System,” *PLoS One*, 2014.
- [3] N. P. Deshmukh, J. J. Caban, R. H. Taylor, G. D. Hager, and E. M. Boctor, “Five-dimensional ultrasound system for soft tissue visualization.,” *Int. J. Comput. Assist. Radiol. Surg.*, Aug. 2015.
- [4] G. Sadigh, R. C. Carlos, C. H. Neal, and B. A. Dwamena, “Accuracy of quantitative ultrasound elastography for differentiation of malignant and benign breast abnormalities: a meta-analysis,” *Breast Cancer Res. Treat.*, vol. 134, no. 3, pp. 923–931, Aug. 2012.
- [5] R. Masuzaki, R. Tateishi, H. Yoshida, T. Sato, T. Ohki, T. Goto, H. Yoshida, S. Sato, Y. Sugioka, H. Ikeda, S. Shiina, T. Kawabe, and M. Omata, “Assessing liver tumor stiffness by transient elastography,” *Hepatol. Int.*, vol. 1, no. 3, pp. 394–397, Oct. 2007.
- [6] M. Dhyani, A. Anvari, and A. E. Samir, “Ultrasound elastography: liver,” *Abdom. Imaging*, vol. 40, no. 4, pp. 698–708, Apr. 2015.

- [7] I. Cespedes, J. Ophir, H. Ponnekanti, and N. Maklad, "Elastography: Elasticity Imaging Using Ultrasound with Application to Muscle and Breast in Vivo," *Ultrason. Imaging*, vol. 15, no. 2, pp. 73–88, Apr. 1993.
- [8] T. A. Krouskop, T. M. Wheeler, F. Kallel, B. S. Garra, and T. Hall, "Elastic Moduli of Breast and Prostate Tissues under Compression," *Ultrason. Imaging*, vol. 20, no. 4, pp. 260–274, Oct. 1998.
- [9] F. Kallel, R. E. Price, E. Konofagou, and J. Ophir, "Elastographic Imaging of the Normal Canine Prostate In Vitro," *Ultrason. Imaging*, vol. 21, no. 3, pp. 201–215, Jul. 1999.
- [10] D. L. Cochlin, R. . Ganatra, and D. F. . Griffiths, "Elastography in the Detection of Prostatic Cancer," *Clin. Radiol.*, vol. 57, no. 11, pp. 1014–1020, Nov. 2002.
- [11] J.-L. Gennisson, N. Grenier, C. Combe, and M. Tanter, "Supersonic Shear Wave Elastography of In Vivo Pig Kidney: Influence of Blood Pressure, Urinary Pressure and Tissue Anisotropy," *Ultrasound Med. Biol.*, vol. 38, no. 9, pp. 1559–1567, 2012.
- [12] N. Grenier, J.-L. Gennisson, F. Cornelis, Y. Le Bras, and L. Couzi, "Renal ultrasound elastography," *Diagn. Interv. Imaging*, vol. 94, no. 5, pp. 545–550, 2013.
- [13] A. Holdsworth, K. Bradley, S. Birch, W. J. Browne, and V. Barberet, "ELASTOGRAPHY OF THE NORMAL CANINE LIVER, SPLEEN AND KIDNEYS," *Vet. Radiol. Ultrasound*, vol. 55, no. 6, pp. 620–627, Nov. 2014.
- [14] T.-M. Mak, Y.-P. Huang, and Y.-P. Zheng, "Liver fibrosis assessment using transient elastography guided with real-time B-mode ultrasound imaging: a feasibility study.," *Ultrasound Med. Biol.*, vol. 39, no. 6, pp. 956–66, Jun. 2013.

- [15] M. Ziol, A. Handra-Luca, A. Kettaneh, C. Christidis, F. Mal, F. Kazemi, V. de Lédinghen, P. Marcellin, D. Dhumeaux, J.-C. Trinchet, and M. Beaugrand, “Noninvasive assessment of liver fibrosis by measurement of stiffness in patients with chronic hepatitis C,” *Hepatology*, vol. 41, no. 1, pp. 48–54, Jan. 2005.
- [16] S. B. Andersen, C. Ewertsen, J. F. Carlsen, B. M. Henriksen, and M. B. Nielsen, “Ultrasound Elastography is Useful for Evaluation of Liver Fibrosis in Children - A Systematic Review.,” *J. Pediatr. Gastroenterol. Nutr.*, Feb. 2016.
- [17] L. Castera, X. Forns, and A. Alberti, “Non-invasive evaluation of liver fibrosis using transient elastography,” *J. Hepatol.*, vol. 48, no. 5, pp. 835–847, 2008.
- [18] L. Sandrin, B. Fourquet, J.-M. Hasquenoph, S. Yon, C. Fournier, F. Mal, C. Christidis, M. Ziol, B. Poulet, F. Kazemi, M. Beaugrand, and R. Palau, “Transient elastography: a new noninvasive method for assessment of hepatic fibrosis,” *Ultrasound Med. Biol.*, vol. 29, no. 12, pp. 1705–1713, 2003.
- [19] I. Sporea, R. Sirli, A. Deleanu, A. Popescu, M. Focsa, M. Danila, and A. Tudora, “Acoustic Radiation Force Impulse Elastography as Compared to Transient Elastography and Liver Biopsy in Patients with Chronic Hepatopathies,” *Ultraschall der Medizin - Eur. J. Ultrasound*, vol. 32, no. S 01, pp. 46–52, Jan. 2011.
- [20] M. Friedrich-Rust, K. Wunder, S. Kriener, F. Sotoudeh, S. Richter, J. Bojunga, E. Herrmann, T. Poynard, C. F. Dietrich, J. Vermehren, S. Zeuzem, and C. Sarrazin, “Liver Fibrosis in Viral Hepatitis: Noninvasive Assessment with Acoustic Radiation Force Impulse Imaging versus Transient Elastography ¹,” *Radiology*, vol. 252, no. 2, pp. 595–604, Aug. 2009.

- [21] T. Varghese, J. . Zagzebski, and F. . Lee, “Elastographic imaging of thermal lesions in the liver in vivo following radiofrequency ablation: preliminary results,” *Ultrasound Med. Biol.*, vol. 28, no. 11, pp. 1467–1473, 2002.
- [22] C. J. Diederich, “Thermal ablation and high-temperature thermal therapy: Overview of technology and clinical implementation,” *Int. J. Hyperth.*, vol. 21, no. 8, pp. 745–753, Dec. 2005.
- [23] H. J. Kang, A. . Cheng, and E. Boctor, “MIDAS Journal - The MUSiiC Toolkit: Modular Real-Time Toolkit for Advanced Ultrasound Research,” *Midas Journal*, 2012. [Online]. Available: <http://www.midasjournal.org/browse/publication/730>.
- [24] P. F. A. U. Foroughi, H.-J. Kang, D. A. Carnegie, M. G. van Vledder, M. A. Choti, G. D. Hager, E. M. Boctor, K. H. J. FAU, C. D. A. C. M. A. FAU, M. van Vledder MG FAU - Choti A., and H. G. D. FAU, “A freehand ultrasound elastography system with tracking for in vivo applications.,” *Ultrasound Med. Biol.*, vol. 39, no. 2, pp. 211–25, Feb. 2013.
- [25] D. Mann, J. J. Caban, P. J. Stolka, E. M. Boctor, and T. S. Yoo, “Multi-dimensional transfer functions for effective visualization of streaming ultrasound and elasticity images,” *SPIE Med. Imaging*, p. 796439, Mar. 2011.
- [26] W. R. Hedrick, D. L. Hykes, and D. E. Starchman, *Ultrasound Physics and Instrumentation*, 3rd ed. 1995.
- [27] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics Of Medical Imaging*. 2002.
- [28] J. L. Prince and J. M. Links, *Medical Imaging Signals and Systems*, 2nd ed. Pearson.

- [29] A. ROHR, S. SPECTOR, S. SIEGEL, R. KATZ, and G. RACHELEFSKY, "Correlation between A-mode ultrasound and radiography in the diagnosis of maxillary sinusitis," *J. Allergy Clin. Immunol.*, vol. 78, no. 1, pp. 58–61, Jul. 1986.
- [30] C. Adam, F. Eckstein, S. Milz, E. Schulte, C. Becker, and R. Putz, "The distribution of cartilage thickness in the knee-joints of old-aged individuals — measurement by A-mode ultrasound," *Clin. Biomech.*, vol. 13, no. 1, pp. 1–10, Jan. 1998.
- [31] S. P. Schiffer, N. W. Rantanen, G. A. Leary, and G. M. Bryan, "Biometric study of the canine eye, using A-mode ultrasonography.," *Am. J. Vet. Res.*, vol. 43, no. 5, pp. 826–30, May 1982.
- [32] J. T. Salonen and R. Salonen, "Ultrasound B-mode imaging in observational studies of atherosclerotic progression.," *Circulation*, vol. 87, no. 3 Suppl, p. II56-65, Mar. 1993.
- [33] A. W. A. Lensing, P. Prandoni, D. Brandjes, P. M. Huisman, M. Vigo, G. Tomasella, J. Krekt, J. W. ten Cate, M. V Huisman, and H. R. Büller, "Detection of deep-vein thrombosis by real-time B-mode ultrasonography," *N. Engl. J. Med.*, vol. 320, no. 6, pp. 342–345, 1989.
- [34] S. L. Myers, J. S. Cohen, P. W. Sheets, and J. R. Bies, "B-mode ultrasound evaluation of skin thickness in progressive systemic sclerosis.," *J. Rheumatol.*, vol. 13, no. 3, pp. 577–80, Jun. 1986.
- [35] H. S. Teixidor and E. Kazam, "Combined mammographic-sonographic evaluation of breast masses.," *AJR. Am. J. Roentgenol.*, vol. 128, no. 3, pp. 409–17, Mar. 1977.
- [36] S. M. Bunce, A. P. Moore, and A. D. Hough, "M-mode ultrasound: a reliable

- measure of transversus abdominis thickness?,” *Clin. Biomech.*, vol. 17, no. 4, pp. 315–317, May 2002.
- [37] G. R. DeVore, R. L. Donnerstein, C. S. Kleinman, L. D. Platt, and J. C. Hobbins, “Fetal echocardiography. I. Normal anatomy as determined by real-time--directed M-mode ultrasound,” *Am. J. Obstet. Gynecol.*, vol. 144, no. 3, pp. 249–60, Oct. 1982.
- [38] G. R. DeVore, B. Siassi, and L. D. Platt, “Fetal echocardiography,” *Am. J. Obstet. Gynecol.*, vol. 150, no. 8, pp. 981–988, Dec. 1984.
- [39] G. Gamble, J. Zorn, G. Sanders, S. MacMahon, and N. Sharpe, “Estimation of arterial stiffness, compliance, and distensibility from M- mode ultrasound measurements of the common carotid artery,” *Stroke*, vol. 25, no. 1, pp. 11–16, Jan. 1994.
- [40] J. M. McMeeken, I. D. Beith, D. J. Newham, P. Milligan, and D. J. Critchley, “The relationship between EMG and change in thickness of transversus abdominis,” *Clin. Biomech. (Bristol, Avon)*, vol. 19, no. 4, pp. 337–42, May 2004.
- [41] A. G. Siqueira-Filho, C. L. Cunha, A. J. Tajik, J. B. Seward, T. T. Schattenberg, and E. R. Giuliani, “M-mode and two-dimensional echocardiographic features in cardiac amyloidosis,” *Circulation*, vol. 63, no. 1, pp. 188–196, Jan. 1981.
- [42] P. R. Hoskins, K. Martin, and A. Thrush, Eds., *Diagnostic Ultrasound Physics and Equipment*, 2nd ed. Cambridge University Press, 2010.
- [43] C. M. Rumack, S. R. Wilson, J. W. Charboneau, and D. Levine., *Diagnostic Ultrasound*, 4th ed. Elsevier Mosby, 2011.

- [44] F. A. Duck, A. . Backer, and H. . Starritt, Eds., *Ultrasound in Medicine*. 1998.
- [45] N. Rao, S. Mehra, and H. Zhu, “Ultrasound speckle statistics variations with imaging system impulse response.” The Institute of Electrical and Electronics Engineers (IEEE).
- [46] C. B. Burckhardt, “Speckle in ultrasound B-mode scans,” *IEEE Trans. Sonics Ultrason.*, vol. 25, no. 1, pp. 1–6, Jan. 1978.
- [47] A. S. Saada, “Elasticity, Theory and Applications,” 2nd ed., New York: Pergamon Press, Inc., 1974.
- [48] P. Chaturvedi, M. F. Insana, and T. J. Hall, “Testing the limitations of 2-D companding for strain imaging using phantoms,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 45, no. 4, pp. 1022–1031, Jul. 1998.
- [49] L. N. Bohs and G. E. Trahey, “A novel method for angle independent ultrasonic imaging of blood flow and tissue motion.,” *IEEE Trans. Biomed. Eng.*, vol. 38, no. 3, pp. 280–6, Mar. 1991.
- [50] F. Yeung, S. F. Levinson, and K. J. Parker, “Multilevel and Motion Model-Based Ultrasonic Speckle Tracking Algorithms,” *Ultrasound Med. Biol.*, vol. 24, no. 3, pp. 427–441, Mar. 1998.
- [51] K. Hoyt, F. Forsberg, and J. Ophir, “Comparison of shift estimation strategies in spectral elastography.,” *Ultrasonics*, vol. 44, no. 1, pp. 99–108, Jan. 2006.
- [52] X. Chen, H. Xie, R. Erkamp, K. Kim, C. Jia, J. M. Rubin, and M. O’Donnell, “3-D Correlation-Based Speckle Tracking,” *Ultrason. Imaging*, vol. 27, no. 1, pp. 21–36, Jan. 2005.

- [53] Y. Zhang, R. T. Brodell, E. N. Mostow, C. J. Vinyard, and H. Marie, "In vivo skin elastography with high-definition optical videos.," *Skin Res. Technol.*, vol. 15, no. 3, pp. 271–82, Aug. 2009.
- [54] M. F. Insana, M. Biegen, P. Chaturvedi, T. J. Hall, and M. Bertrand, "Signal processing strategies in acoustic elastography," in *1996 IEEE Ultrasonics Symposium. Proceedings*, 1996, vol. 2, pp. 1139–1142.
- [55] J. E. Lindop, G. M. Treece, A. H. Gee, and R. W. Prager, "3D elastography using freehand ultrasound.," *Ultrasound Med. Biol.*, vol. 32, no. 4, pp. 529–45, Apr. 2006.
- [56] H. Rivaz, E. Boctor, P. Foroughi, R. Zellars, G. Fichtinger, and G. Hager, "Ultrasound elastography: a dynamic programming approach," *IEEE Trans. Med. imaging JID - 8310780*, vol. 27, no. 1558–254, pp. 1373–7, Oct. 2008.
- [57] H. Rivaz, E. M. Boctor, M. A. Choti, and G. D. Hager, "Real-time regularized ultrasound elastography.," *IEEE Trans. Med. Imaging*, vol. 30, no. 4, pp. 928–45, Apr. 2011.
- [58] P. Foroughi, H.-J. Kang, D. A. Carnegie, M. G. van Vledder, M. A. Choti, G. D. Hager, and E. M. Boctor, "A freehand ultrasound elastography system with tracking for in vivo applications.," *Ultrasound Med. Biol.*, vol. 39, no. 2, pp. 211–25, Feb. 2013.
- [59] D. Parkin Maxwell and L. Fernandez M., G., "Use of Statistics to Assess the Global Burden of Breast Cancer," *Breast J.*, vol. 12, no. s1, pp. S70-80, 2006.
- [60] S. Wojcinski, E. Boehme, A. Farrokh, P. Soergel, F. Degenhardt, and P. Hillemanns, "Ultrasound real-time elastography can predict malignancy in BI-RADS®-US 3

- lesions.,” *BMC Cancer*, vol. 13, no. 1, p. 159, Jan. 2013.
- [61] A. Thomas, T. Fischer, H. Frey, R. Ohlinger, S. Grunwald, J.-U. Blohmer, K.-J. Winzer, S. Weber, G. Kristiansen, B. Ebert, and S. Kümmel, “Real-time elastography--an advanced method of ultrasound: First results in 108 patients with breast lesions.,” *Ultrasound Obstet. Gynecol.*, vol. 28, no. 3, pp. 335–40, Sep. 2006.
- [62] M. Brock, C. von Bodman, R. J. Palisaar, B. Löppenber, F. Sommerer, T. Deix, J. Noldus, and T. Eggert, “The impact of real-time elastography guiding a systematic prostate biopsy to improve cancer detection rate: a prospective study of 353 patients.,” *J. Urol.*, vol. 187, no. 6, pp. 2039–43, Jun. 2012.
- [63] M. Sumura, K. Shigeno, T. Hyuga, T. Yoneda, H. Shiina, and M. Igawa, “Initial evaluation of prostate cancer with real-time elastography based on step-section pathologic analysis after radical prostatectomy: a preliminary study.,” *Int. J. Urol.*, vol. 14, no. 9, pp. 811–6, Sep. 2007.
- [64] G. Ferraioli, C. Tinelli, A. Malfitano, B. D. Bello, G. Filice, C. Filice, E. Above, G. Barbarini, E. Brunetti, W. Calderon, M. Di Gregorio, R. Lissandrini, S. Ludovisi, L. Maiocchi, G. Michelone, M. Mondelli, S. F. A. Patruno, A. Perretti, G. Poma, P. Sacchi, M. Zaramella, and M. Zicchetti, “Performance of Real-Time Strain Elastography, Transient Elastography, and Aspartate-to-Platelet Ratio Index in the Assessment of Fibrosis in Chronic Hepatitis C,” *Am. J. Roentgenol.*, vol. 199, no. 1, pp. 19–25, Jul. 2012.
- [65] M. Xie, X. Zhang, J. Zhan, and K. Hua, “Application of real-time ultrasound elastography for discrimination of low- and high-grade serous ovarian carcinoma,”

J. ultrasound Med. Off. J. Am. Inst. Ultrasound Med. JID - 8211547, vol. 32, no. 2, pp. 257–62, Feb. 618.

- [66] T. Hinz, T. Hoeller, J. Wenzel, T. Bieber, and M.-H. Schmid-Wendtner, “Real-Time Tissue Elastography as Promising Diagnostic Tool for Diagnosis of Lymph Node Metastases in Patients with Malignant Melanoma: A Prospective Single-Center Experience,” *Dermatology*, vol. 226, no. 1, pp. 81–90, 2013.
- [67] C. Cappelli, I. Pirola, E. Gandossi, B. Agosti, E. Cimino, C. Casella, A. Formenti, and M. Castellano, “Real-time Elastography: A Useful Tool for Predicting Malignancy in Thyroid Nodules With Nondiagnostic Cytologic Findings ,” *J. Ultrasound Med.* , vol. 31, no. 11, pp. 1777–1782, Nov. 2012.
- [68] P. Trimboli, R. Guglielmi, S. Monti, I. Misischi, F. Graziano, N. Nasrollah, S. Amendola, S. N. Morgante, M. G. Deiana, S. Valabrega, V. Toscano, and E. Papini, “Ultrasound sensitivity for thyroid malignancy is increased by real-time elastography: a prospective multicenter study.,” *J. Clin. Endocrinol. Metab.*, vol. 97, no. 12, pp. 4524–30, Dec. 2012.
- [69] H. Rivaz, I. Fleming, L. Assumpcao, G. Fichtinger, U. Hamper, M. Choti, G. Hager, and E. Boctor, “Ablation Monitoring with Elastography: 2D In-vivo and 3D Ex-vivo Studies,” in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008*, vol. 5242, no. Pt 2, Springer Berlin Heidelberg, 2008, pp. 458–466.
- [70] J. Bercoff, M. Tanter, and M. Fink, “Supersonic shear imaging: a new technique for soft tissue elasticity mapping.,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 51, no. 4, pp. 396–409, Apr. 2004.

- [71] E. Turgay, S. Salcudean, and R. Rohling, "Identifying the mechanical properties of tissue by ultrasound strain imaging.," *Ultrasound Med. Biol.*, vol. 32, no. 2, pp. 221–35, Feb. 2006.
- [72] E. M. Boctor, N. Deshmukh, M. S. Ayad, C. Clarke, K. Dickie, M. A. Choti, and E. C. Burdette, "Three-dimensional heat-induced echo-strain imaging for monitoring high-intensity acoustic ablation," in *SPIE Medical Imaging*, 2009, p. 72650R–72650R.
- [73] X. Yang, S. Deka, and R. Righetti, "A hybrid CPU-GPGPU approach for real-time elastography," *Ultrason. Ferroelectr. Freq. Control. IEEE Trans.*, vol. 58, no. 12, pp. 2631–2645, Dec. 2011.
- [74] S. P. Nair, X. Yang, T. A. Krouskop, and R. Righetti, "Performance analysis of a new real-time elastographic time constant estimator," *IEEE Trans. Med. imaging JID - 8310780*, vol. 30, no. 1558–254, pp. 497–511, Feb. 524.
- [75] NVIDIA Corporation, "NVIDIA CUDA C Programming Guide." Jun-2011.
- [76] E. Boni, L. Bassi, A. Dallai, F. Guidi, A. Ramalli, S. Ricci, J. Housden, and P. Tortoli, "A reconfigurable and programmable FPGA-based system for nonstandard ultrasound methods.," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 59, no. 7, pp. 1378–85, Jul. 2012.
- [77] J. J. R. Raj, S. M. K. Rahman, and A. Sneh, "Electronic hardware design for ultrasound Transient Elastography," *Int. J. Eng. Sci. Technol.*, vol. 4, no. 8, 2012.
- [78] F. K. Schneider, A. Agarwal, Y. M. Yoo, T. Fukuoka, and Y. Kim, "A fully programmable computing architecture for medical ultrasound machines.," *IEEE*

Trans. Inf. Technol. Biomed., vol. 14, no. 2, pp. 538–40, Mar. 2010.

- [79] A. Popescu, I. Sporea, R. Sirli, S. Bota, M. Focşa, M. Dănilă, D. Nicolîţă, A. Martie, M. Sendroiu, and A. Juchiş, “The mean values of liver stiffness assessed by Acoustic Radiation Force Impulse elastography in normal subjects.,” *Med. Ultrason.*, vol. 13, no. 1, pp. 33–7, Mar. 2011.
- [80] X. Xiao, N. Le, G. Corner, G. Nabi, A. Melzer, and Z. Huang, “Organ motion tracking in USgFUS - A feasibility study using sonoelastography,” in *12TH INTERNATIONAL SYMPOSIUM ON THERAPEUTIC ULTRASOUND*, 2012, vol. 1503, no. 1, pp. 135–139.
- [81] K. Nakashima, T. Shiina, M. Sakurai, K. Enokido, T. Endo, H. Tsunoda, E. Takada, T. Umemoto, and E. Ueno, “JSUM ultrasound elastography practice guidelines: breast,” *J. Med. Ultrason.*, vol. 40, no. 4, pp. 359–391, Jul. 2013.
- [82] S. Billings, N. Deshmukh, H. J. Kang, R. Taylor, and E. M. Boctor, “System for robot-assisted real-time laparoscopic ultrasound elastography,” in *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*, 2012, p. 83161W–83161W–8.
- [83] H. T. Sen, N. Deshmukh, R. Goldman, P. Kazanzides, R. H. Taylor, E. Boctor, and N. Simaan, “Enabling technologies for natural orifice transluminal endoscopic surgery (N.O.T.E.S) using robotically guided elasticity imaging,” *SPIE Med. Imaging*, p. 83161Y–83161Y, Feb. 2012.
- [84] J. P. Lewis, “Fast Normalized Cross-Correlation,” *Vis. interface*, vol. 10, no. 1, pp. 120–123, 1995.

- [85] H.-J. Kang, N. P. Deshmukh, P. Stolka, E. C. Burdette, and E. M. Boctor, "Ultrasound imaging software framework for real-time monitoring of acoustic ablation therapy," *SPIE Med. Imaging*, p. 83201E–83201E, Feb. 2012.
- [86] H. F. A. U. Rivaz, P. F. A. U. Foroughi, I. F. A. U. Fleming, R. F. A. U. Zellars, E. F. A. U. Boctor, and G. Hager, "Tracked regularized ultrasound elastography for targeting breast radiotherapy," *Med. image Comput. Comput. Interv. MICCAI ...International Conf. Med. Image Comput. Comput. Interv. JID - 101249582*, vol. 12, no. Pt 1, pp. 507–15, Jan. 607.
- [87] J. Jiang, T. J. Hall, and A. M. Sommer, "A novel performance descriptor for ultrasonic strain imaging: a preliminary study.," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 53, no. 6, pp. 1088–102, Jun. 2006.
- [88] T. J. Hall, Y. Zhu, and C. S. Spalding, "In vivo real-time freehand palpation imaging.," *Ultrasound Med. Biol.*, vol. 29, no. 3, pp. 427–35, Mar. 2003.
- [89] N. Deshmukh, H. Rivaz, and E. Boctor, "GPU-based elasticity imaging algorithms," in *Proc. Int. Conf. Med. Imag. Comp. & Comp. Assist. Interven*, 2009.
- [90] N. P. Deshmukh, J. Caban, R. H. Taylor, G. D. Hager, and E. M. Boctor, "Five dimensional Ultrasound Visualization Framework to Guide and Monitor Ablative Therapy," *IPCAI/CARS/IJCARS*, 2015.
- [91] K. G. Shin and P. Ramanathan, "Real-time computing: a new discipline of computer science and engineering," *Proc. IEEE*, vol. 82, no. 1, pp. 6–24, 1994.
- [92] A. Heinecke, M. Klemm, and H.-J. Bungartz, "From GPGPU to Many-Core: Nvidia Fermi and Intel Many Integrated Core Architecture," *Comput. Sci. Eng.*, vol. 14, no.

2, pp. 78–83, Mar. 2012.

- [93] E. Pospisil, R. Rohling, R. Azar, and S. Salcudean, “4-D x 3-D ultrasound: real-time scan conversion, filtering, and display of displacement vectors with a motorized curvilinear transducer.,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 57, no. 10, pp. 2271–83, Oct. 2010.
- [94] E. M. Boctor, M. Matinfar, O. Ahmad, H. Rivaz, M. Choti, and R. H. Taylor, “Elasticity-based three dimensional ultrasound real-time volume rendering,” in *SPIE Medical Imaging*, 2009, p. 72612V–72612V–9.
- [95] J. A. Weis, T. E. Yankeelov, S. A. Munoz, R. A. Sastry, S. L. Barnes, L. R. Arlinghaus, X. Li, and M. I. Miga, “A consistent pre-clinical/clinical elastography approach for assessing tumor mechanical properties in therapeutic systems,” in *SPIE Medical Imaging*, 2013, p. 86721F.
- [96] V. Heid, H. Evers, C. Henn, G. Glombitza, and H.-P. Meinzer, “5D interactive real time Doppler ultrasound visualization of the heart,” in *Medical Imaging 2000*, 2000, pp. 494–500.
- [97] R. Siegel, D. Naishadham, and A. Jemal, “Cancer statistics, 2013.,” *CA. Cancer J. Clin.*, vol. 63, no. 1, pp. 11–30, Jan. 2013.
- [98] O. Falou, A. Sadeghi-Naini, S. Prematilake, E. Sofroni, N. Papanicolau, S. Iradji, Z. Jahedmotlagh, S. Lemon-Wong, J.-P. Pignol, E. Rakovitch, J. Zubovits, J. Spayne, R. Dent, M. Trudeau, J. F. Boileau, F. C. Wright, M. J. Yaffe, and G. J. Czarnota, “Evaluation of neoadjuvant chemotherapy response in women with locally advanced breast cancer using ultrasound elastography.,” *Transl. Oncol.*, vol. 6, no.

- 1, pp. 17–24, Feb. 2013.
- [99] C. Weismann, C. Mayr, H. Egger, and A. Auer, “Breast Sonography - 2D, 3D, 4D Ultrasound or Elastography?,” *Breast Care (Basel)*, vol. 6, no. 2, pp. 98–103, Jan. 2011.
- [100] E. M. Boctor, M. Choti, G. Fictinger, R. Taylor, and J. L. Prince, “Robotic 5-dimensional ultrasound,” Patent US7901357 B2, 08-Mar-2011.
- [101] F. Torres, Z. Fanti, and F. Arambula Cosío, “3D freehand ultrasound for medical assistance in diagnosis and treatment of breast cancer: preliminary results,” in *IX International Seminar on Medical Information Processing and Analysis*, 2013, p. 89220K.
- [102] A. Sayed, G. Layne, J. Abraham, and O. Mukdadi, “Nonlinear characterization of breast cancer using multi-compression 3D ultrasound elastography in vivo,” *Ultrasonics*, vol. 53, no. 5, pp. 979–991, 2013.
- [103] G. Jedrzejewski, I. Ben-Skowronek, M. M. Wozniak, A. Brodzisz, E. Budzynska, and A. P. Wieczorek, “Testicular adrenal rest tumors in boys with congenital adrenal hyperplasia: 3D US and elastography – Do we get more information for diagnosis and monitoring?,” *J. Pediatr. Urol.*, vol. 9, no. 6, pp. 1032–1037, 2013.
- [104] M. Ying, Y.-P. Zheng, B. C.-W. Kot, J. C.-W. Cheung, S. C.-H. Cheng, and D. L.-W. Kwong, “Three-Dimensional Elastography for Cervical Lymph Node Volume Measurements: A Study to Investigate Feasibility, Accuracy and Reliability,” *Ultrasound Med. Biol.*, vol. 39, no. 3, pp. 396–406, 2013.
- [105] P. Foroughi, J. Burgner, M. A. Choti, R. J. Webster III, G. D. Hager, and E. M.

- Boctor, "Towards intra-operative monitoring of ablation using tracked 3D ultrasound elastography and internal palpation," in *SPIE Medical Imaging*, 2012, p. 83200T.
- [106] K. Kanenishi, U. Hanaoka, J. Noguchi, G. Marumo, and T. Hata, "4D ultrasound evaluation of fetal facial expressions during the latter stages of the second trimester," *Int. J. Gynecol. Obstet.*, vol. 121, no. 3, pp. 257–260, 2013.
- [107] A. Wittek, K. Karatolios, P. Bihari, T. Schmitz-Rixen, R. Moosdorf, S. Vogt, and C. Blase, "In vivo determination of elastic properties of the human aorta based on 4D ultrasound data," *J. Mech. Behav. Biomed. Mater.*, vol. 27, pp. 167–183, 2013.
- [108] G. Hilde, J. Staer-Jensen, F. Siafarikas, K. Gjestland, M. Ellström Engh, and K. Bø, "How well can pelvic floor muscles with major defects contract? A cross-sectional comparative study 6 weeks after delivery using transperineal 3D/4D ultrasound and manometer.," *BJOG*, vol. 120, no. 11, pp. 1423–9, Oct. 2013.
- [109] I. H. Braekken, M. Majida, M. E. Engh, and K. Bø, "Test-retest reliability of pelvic floor muscle contraction measured by 4D ultrasound.," *Neurourol. Urodyn.*, vol. 28, no. 1, pp. 68–73, Jan. 2009.
- [110] S. Yagel, S. M. Cohen, I. Shapiro, and D. V Valsky, "3D and 4D ultrasound in fetal cardiac scanning: a new look at the fetal heart.," *Ultrasound Obstet. Gynecol.*, vol. 29, no. 1, pp. 81–95, Jan. 2007.
- [111] S. Vijayan, S. Klein, E. F. Hofstad, F. Lindseth, B. Ystgaard, and T. Lango, "Validation of a non-rigid registration method for motion compensation in 4D ultrasound of the liver," in *2013 IEEE 10th International Symposium on Biomedical*

Imaging, 2013, pp. 792–795.

- [112] N. Gritzmann, C. F. Weismann, and L. Datz, “Diagnostic algorithm: How to make use of new 2D, 3D and 4D ultrasound technologies in breast imaging,” *Eur. J. Radiol.*, vol. 64, no. 2, pp. 250–257, 2007.
- [113] R. Achiron, L. Gindes, Y. Zalel, S. Lipitz, and B. Weisz, “Three- and four-dimensional ultrasound: new methods for evaluating fetal thoracic anomalies,” *Ultrasound Obstet. Gynecol.*, vol. 32, no. 1, pp. 36–43, Jul. 2008.
- [114] S. S. Mahdavi, M. Moradi, W. J. Morris, S. L. Goldenberg, and S. E. Salcudean, “Fusion of ultrasound B-mode and vibro-elastography images for automatic 3D segmentation of the prostate,” *IEEE Trans. Med. Imaging*, vol. 31, no. 11, pp. 2073–82, Nov. 2012.
- [115] M. Moradi, P. Abolmaesumi, and P. Mousavi, “Tissue typing using ultrasound RF time series: Experiments with animal tissue samples,” *Med. Phys.*, vol. 37, no. 8, p. 4401, Jul. 2010.
- [116] N. Uniyal, H. Eskandari, P. Abolmaesumi, S. Sojoudi, P. Gordon, L. Warren, R. N. Rohling, S. E. Salcudean, and M. Moradi, “Ultrasound RF time series for classification of breast lesions,” *IEEE Trans. Med. Imaging*, vol. 34, no. 2, pp. 652–61, Feb. 2015.
- [117] A. Itoh, E. Ueno, E. Tohno, H. Kamma, H. Takahashi, T. Shiina, M. Yamakawa, and T. Matsumura, “Breast Disease: Clinical Application of US Elastography for Diagnosis,” *Radiology*, vol. 239, no. 2, pp. 341–350, 2006.
- [118] E. S. Burnside, T. J. Hall, A. M. Sommer, G. K. Hesley, G. A. Sisney, W. E.

Svensson, J. P. Fine, J. Jiang, and N. J. Hangiandreou, "Differentiating Benign from Malignant Solid Breast Masses with US Strain Imaging," *Radiology*, vol. 245, no. 2, pp. 401–410, 2007.

[119] H. Takahashi, H. Hasegawa, and H. Kanai, "Improvement of Automated Identification of the Heart Wall in Echocardiography by Suppressing Clutter Component," *Jpn. J. Appl. Phys.*, vol. 52, no. 7S, p. 07HF17, Jul. 2013.

9 Appendix A

9.1 3D scan conversion on GPU

Scan conversion helps to convert the data collected on the surface of 4D probe that depicts spherical sector, which is an array of images, to the appropriate location in space. This section details the equations used for scan conversion on GPU. We calculate φ , the angle of field of view by

$$\varphi = N \times \delta \quad (1)$$

where N is the number of slices per volume and δ is the step angle of the motor. We then determine the dimension of the scan-converted volume by

$$y' = \frac{x' \times \left(\tau + y - \tau \times \cos\left(\frac{\varphi}{2}\right) \right)}{\left(2 \times (\tau + y) \times \tan\left(\frac{\varphi}{2}\right) \right)} \quad (2)$$

where (x', y') is the new (elevation, axial) size in pixels. In our case, x' is user defined, τ is the radius of the curvature in pixels and y is the number of pixels in the axial direction in the original volume. We then calculate the step size in the axial direction by

$$r_i = \tau + i + 1 \quad (3)$$

where r_i is the step size in the axial direction (pixels) from the origin of the probe for all $i = 0$ to y . We then calculate the angle θ_j from the center slice by

$$\theta_j = -\frac{\varphi}{2} + j \times \delta \quad (4)$$

where θ_j is negative or positive on the opposite sides of the center slice, $j = 0$ to x , where x is the number of pixels in the elevational direction. Typically $x = N$ when slice thickness is 1 pixel. We now calculate the forward index i'' into the final volume along the axial direction by

$$i'' = \frac{x' \times \left(\text{round}(r_i \times \cos(\theta_j)) - \tau \times \cos\left(\frac{\varphi}{2}\right) \right)}{\text{round}\left(2 + (\tau + y) \times \tan\left(\frac{\varphi}{2}\right)\right)} \quad (5).$$

We now calculate the forward index j'' in the final volume along the elevation direction by

$$j'' = \frac{y' \times \left((\tau + y) \times \tan\left(\frac{\varphi}{2}\right) + \text{round}(r_i \times \sin(\theta_j)) \right)}{\left(\tau + y - \tau \times \cos\left(\frac{\varphi}{2}\right) \right)} \quad (6).$$

The following equation gives final scan conversion after substituting values of y' , i'' , and j'' from Eqs. (3), (6), and (7) respectively:

$$\text{output}[k \times x' \times y' + i'' \times x' + j''] = \text{input}[k \times x \times y + i \times x + j] \quad (7)$$

where *output* is the output volume buffer which will contain the scan-converted volume, *input* is the input volume buffer which contains the US machine acquired volume, $k = 0 - z$, z is the number of pixels in the lateral direction.

GPU kernels execute Eqs. (1)-(7) that are highly parallel since index mappings are independent of each other. In the case of conflict, where multiple pixels from the different locations in the original buffer get mapped to the same pixel in the output buffer, we simply consider the maximum pixel value. There are holes created into the output slices, which are recovered using interpolation by a simple GPU-based averaging filter.

For this scan conversion to work and to keep the equations simple, we need to rotate the volume around an axis at the center of the volume and axis-align to elevation-axial-lateral directions. The rotation depicts swapping of elevation-axial and lateral-axial slices. The following equations give us an easy way to swap the values in the 3D buffers:

$$postswap[j \times y \times x + k \times x + i] = preswap[i \times y \times z + k \times z + (z - j - 1)] \quad (8)$$

$$preswap[j \times y \times x + k \times x + (x - i - 1)] = postswap[i \times y \times z + k \times z + j] \quad (9)$$

where pre- and post-swap are the buffers before and after swapping of the pixels. A GPU kernel executes Eq. 8 just before the scan conversion and Eq. 9 after the scan conversion.

10 Appendix B

10.1 Data for multi-stream GPU elastography

The data for experiments in Chapter 4, that is earlier published as [2], can be found at:

Source code: <https://sourceforge.net/projects/multistreamelastography/>

Data: <http://dx.doi.org/10.7910/DVN/28010>

10.2 Data for five-dimensional ultrasound system

The data for experiments in Chapter 6, that is earlier published as [3], can be found at:

Data: <http://dx.doi.org/10.7910/DVN/KZDHYE>

10.3 Data for elastography image pair selection system by computing elastography on limited regions of interest

The data for experiments in Chapter 5, is presented in this section. The data can be found at:

Data: <http://dx.doi.org/10.7910/DVN/28010>

Table 10.1 Unfiltered vs Filtered elastography dataset (Sr. no. 1 – 75) for CNR values and used to generate the results in the section 5.3.2.1.

Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered
1	0.0157	0.2618	26	4.5279	0.0771	51	4.3686	1.7027
2	0.9464	0.2184	27	4.5245	2.4621	52	1.6034	2.0861
3	0.0592	0.1878	28	0.9829	2.3756	53	0.4356	0.7284
4	2.0861	0.3510	29	2.8279	3.1224	54	1.5652	0.7091
5	0.2742	0.3193	30	2.6896	0.1052	55	0.0612	0.6997
6	2.3915	0.1650	31	0.3616	0.3030	56	0.1221	1.9605
7	2.1633	0.0308	32	2.1416	0.1332	57	5.3959	0.2713
8	2.1194	0.1073	33	2.6025	0.1786	58	0.2418	0.0336
9	1.5700	0.1785	34	0.1627	0.0188	59	3.8199	0.0516
10	2.0082	0.3992	35	0.0572	0.0764	60	4.5812	0.1172
11	2.9888	0.3457	36	0.1196	0.3226	61	4.5680	2.6734
12	0.0949	0.4072	37	0.1964	2.3448	62	0.3912	1.8292
13	3.9217	0.2406	38	2.3416	1.3703	63	0.1614	1.4713
14	0.6862	0.1604	39	3.8025	0.0319	64	1.7983	0.0853
15	1.7299	0.4284	40	0.2541	0.3431	65	1.4623	0.5075
16	1.6597	0.1124	41	3.6489	0.0157	66	3.9524	0.3753
17	1.8349	0.2572	42	0.3413	0.2383	67	0.2725	0.0153
18	1.4302	0.0866	43	3.0854	0.6978	68	3.1971	0.3759
19	0.2933	0.2514	44	3.2571	1.6268	69	3.2868	1.0701
20	2.4581	0.0774	45	3.3038	0.9464	70	1.0217	2.0011
21	1.9506	0.5932	46	3.1165	1.6797	71	1.1306	2.9321
22	1.5164	0.1219	47	2.6262	0.2848	72	0.1773	0.2742
23	1.6887	0.1307	48	0.6730	0.8689	73	0.8203	1.7992
24	0.2231	0.1948	49	4.7457	0.0592	74	2.6227	2.3915
25	1.7900	0.1461	50	3.2294	0.3561	75	2.7102	2.4422

Table 10.2 Unfiltered vs Filtered elastography dataset (Sr. no. 76 – 106) for CNR values and used to generate the results in the section 5.3.2.1.

Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered
76	0.2150	1.8736	101	1.0180	0.4876
77	0.1782	2.2093	102	3.0642	1.6658
78	2.3811	1.7947	103	3.6253	0.9563
79	3.2566	1.0608	104	3.6583	1.4360
80	3.6077	1.6875	105	3.2873	1.7299
81	3.8252	2.1633	106	0.5087	1.6597
82	0.3724	2.1194			
83	3.2844	1.5700			
84	2.3671	2.0015			
85	2.1410	2.0082			
86	2.9623	1.4874			
87	0.2647	0.5378			
88	0.0006	0.7058			
89	0.4206	2.8579			
90	0.2031	2.9888			
91	0.1244	0.0949			
92	3.1522	0.1283			
93	0.6124	3.7848			
94	2.6108	3.9217			
95	0.6954	3.0546			
96	3.2973	0.6862			
97	3.4205	0.8640			
98	2.9129	0.6544			
99	3.3271	0.9548			
100	0.6528	0.2649			

Table 10.3 Unfiltered vs Filtered elastography dataset (Sr. no. 1 – 75) for SNR values and used to generate the results in the section 5.3.2.1.

Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered
1	1.0057	0.5458	26	0.8872	0.5360	51	1.1247	1.1076
2	1.1439	0.4851	27	0.9450	1.0261	52	0.8352	1.1296
3	1.0521	0.4593	28	0.9195	1.0097	53	0.4986	0.7588
4	1.1296	0.5526	29	1.1353	1.1018	54	1.1207	0.7636
5	0.9585	0.4485	30	1.1061	0.6653	55	0.9125	0.6915
6	1.1587	0.4372	31	1.0361	0.5360	56	1.2551	0.6794
7	1.1552	0.4518	32	1.1391	0.4957	57	1.0132	0.4494
8	1.1399	0.4434	33	1.1140	0.7194	58	0.6337	0.4597
9	1.0414	0.4475	34	1.3502	0.5014	59	1.1055	0.4134
10	1.1584	0.4157	35	0.9645	0.6161	60	0.9819	0.4511
11	0.8663	0.4566	36	0.5608	0.7613	61	1.1145	1.1348
12	0.5932	0.4048	37	1.2962	1.0502	62	1.0031	1.0847
13	1.0031	0.4005	38	1.2974	1.0076	63	1.6765	1.1118
14	1.2463	0.4173	39	1.1370	0.5046	64	1.0010	0.6699
15	1.0956	0.4702	40	0.8554	0.8634	65	0.8967	0.6533
16	1.1385	0.4595	41	1.0923	1.0057	66	1.0229	0.6274
17	1.1156	0.4545	42	1.1871	0.9729	67	0.9448	0.5738
18	1.0778	0.3888	43	1.0615	1.0059	68	0.9504	0.6303
19	3.3994	0.4936	44	1.1892	1.1380	69	1.0389	0.8544
20	1.1459	0.4696	45	1.1729	1.1439	70	1.0468	1.1626
21	1.1246	0.5772	46	1.1558	1.1127	71	0.7996	1.1364
22	1.0332	0.4639	47	1.0710	0.8840	72	0.9437	0.9585
23	1.1976	0.5316	48	0.6800	0.9295	73	0.9278	0.9784
24	1.0666	0.4512	49	1.1652	1.0521	74	1.0079	1.1587
25	1.0629	0.5099	50	1.0326	0.9295	75	1.0187	1.1175

Table 10.4 Unfiltered vs Filtered elastography dataset (Sr. no. 76 – 106) for SNR values and used to generate the results in the section 5.3.2.1.

Sr. no.	Filtered	Unfiltered	Sr. no.	Filtered	Unfiltered
76	1.3654	0.9602	101	0.9158	0.6319
77	1.0025	1.0930	102	1.0793	0.9176
78	1.0075	1.1335	103	1.1534	0.9265
79	1.1459	1.0397	104	1.1101	1.0684
80	1.1212	1.1461	105	1.1553	1.0956
81	1.1334	1.1552	106	1.1849	1.1385
82	1.2238	1.1399			
83	1.1673	1.0414			
84	1.1077	1.1526			
85	1.0229	1.1584			
86	0.9938	1.0336			
87	0.8267	0.5107			
88	1.2904	0.4513			
89	1.2712	0.9016			
90	1.0095	0.8663			
91	1.3077	0.5932			
92	1.1192	0.4215			
93	1.0535	0.9860			
94	1.0727	1.0031			
95	0.9178	1.1115			
96	1.0882	1.2463			
97	1.1332	0.8563			
98	1.1661	0.4826			
99	1.0305	0.8444			
100	1.0043	0.7000			

Table 10.5 Varying size of region of interest to calculate elastography dataset (Sr. no. 1 – 25) for CNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
1	0.2618	0.2618	0.2618	0.1029
2	0.2184	0.1604	0.0866	0.1461
3	0.3193	0.2572	0.1307	2.4621
4	0.4072	0.1461	0.1461	0.7023
5	0.1604	2.4621	2.4621	0.1052
6	0.2572	2.3756	2.3756	0.3226
7	0.0866	3.1224	3.1224	1.3703
8	0.5932	0.3226	0.1052	0.0157
9	0.1307	2.3448	0.3226	0.2383
10	0.1461	1.3703	2.3448	0.6978
11	2.4621	0.0157	0.0492	1.6268
12	2.3756	0.2383	0.0157	0.9464
13	3.1224	0.6978	0.6978	1.6797
14	0.0188	1.6268	1.6268	0.0592
15	0.3226	0.9464	0.9464	0.3561
16	2.3448	0.0592	0.0498	1.7027
17	1.3703	0.3561	0.3561	2.0861
18	0.3431	1.7027	1.7027	0.7284
19	0.0157	2.0861	2.0861	0.6997
20	0.2383	0.7284	0.7284	1.9605
21	0.6978	0.6997	0.6997	0.1172
22	1.6268	1.9605	1.9605	2.6734
23	0.9464	0.0516	0.1172	1.4713
24	1.6797	0.1172	1.8292	1.0701
25	0.2848	2.6734	1.4713	2.0011

Table 10.6 Varying size of region of interest to calculate elastography dataset (Sr. no. 26 – 50) for CNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
26	0.0592	1.8292	1.0701	2.9321
27	0.3561	1.4713	2.0011	0.2742
28	1.7027	2.9321	2.9321	2.3915
29	2.0861	0.2742	0.2742	2.4422
30	0.7284	2.3915	2.3915	1.8736
31	0.7091	2.4422	2.4422	2.2093
32	0.6997	1.8736	1.8736	1.7947
33	1.9605	2.2093	1.7947	1.0608
34	0.2713	1.7947	1.0608	1.6875
35	0.0516	1.0608	2.1633	2.1633
36	0.1172	1.6875	2.1194	2.1194
37	2.6734	2.1633	1.5700	1.5700
38	1.8292	2.1194	2.0082	2.0015
39	1.8292	1.5700	1.4874	2.0082
40	0.0853	2.0015	2.8579	1.4874
41	0.5075	2.0082	0.0949	2.8579
42	0.3753	1.4874	3.7848	2.9888
43	2.0011	2.8579	3.9217	0.0949
44	2.9321	2.9888	3.0546	0.1283
45	0.2742	0.0949	0.6862	3.7848
46	2.3915	0.1283	0.8640	3.9217
47	2.4422	3.7848	0.9548	0.6862
48	0.0549	3.9217	0.2649	0.0730
49	1.7947	3.0546	0.4876	0.2649
50	1.0608	0.6862	1.6658	1.6658

Table 10.7 Varying size of region of interest to calculate elastography dataset (Sr. no. 51 – 75) for CNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
51	2.1194	0.8640	0.9563	1.4360
52	1.5700	0.2649	1.4360	1.7299
53	2.0082	0.4876	1.7299	1.6597
54	1.4874	0.1348	1.3641	1.8349
55	0.7058	0.9563	1.8349	2.4035
56	2.8579	1.4360	2.4035	0.1774
57	2.9888	1.7299	1.4302	2.7782
58	0.0949	1.6597	2.7782	0.9135
59	0.1283	0.0891	0.9135	0.2065
60	3.7848	1.8349	2.4581	1.9506
61	3.9217	2.4035	0.1661	1.5164
62	3.0546	1.4302	1.9506	0.1767
63	0.6862	0.2935	1.5164	0.2231
64	0.8640	0.9135	1.6887	1.7900
65	0.2210	0.2029	0.2231	2.6018
66	0.2649	1.9506	1.7900	2.3245
67	0.4876	0.1470	2.6018	4.5279
68	1.6658	0.2231	2.3245	4.5245
69	0.3232	1.7900	4.5279	3.5382
70	1.4360	2.6018	4.5245	2.6064
71	1.7299	2.3245	3.5382	3.1633
72	1.8349	4.5279	2.6064	0.9829
73	2.4035	4.5245	3.1633	2.8279
74	0.1789	3.5382	0.9829	0.1440
75	2.7782	2.6064	2.8279	0.1453

Table 10.8 Varying size of region of interest to calculate elastography dataset (Sr. no. 76 – 106) for CNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
76	2.4581	0.9829	0.1440	0.0678
77	1.9506	2.8279	0.0678	4.1361
78	1.5164	0.1453	0.0975	0.3554
79	1.6887	0.0678	0.3554	0.7026
80	1.0591	0.1078	0.7026	2.6896
81	0.0381	0.7026	2.6896	2.9270
82	1.7947	2.6896	1.3264	2.0489
83	2.3245	2.9270	1.2976	1.3264
84	4.5245	2.0489	1.7443	1.2976
85	3.5382	0.1061	0.3616	1.7443
86	3.1633	1.2976	2.1307	0.3616
87	0.9829	1.7443	1.7247	0.0129
88	2.8279	0.3616	2.1416	2.1416
89	0.1440	2.1307	2.6025	2.6025
90	0.1453	0.0016	0.1627	0.1627
91	0.0678	2.1416	2.3342	2.3342
92	2.0650	2.6025	0.3667	0.3667
93	0.3554	0.1627	0.6134	2.0874
94	0.7026	0.0443	0.1196	0.6134
95	2.6896	0.3667	0.1367	0.1196
96	2.9270	2.0874	1.0898	0.1367
97	2.0489	0.6134	3.5876	1.0898
98	1.2976	0.1196	5.0051	3.5876
99	1.7443	0.1367	5.2231	5.0051
100	0.0195	1.0898	0.6571	5.2231
101	2.1416	3.5876	0.1959	0.6571
102	2.6025	5.2231	2.3416	3.5137
103	0.1627	4.2633	3.8025	2.3416
104	0.0778	0.6571	0.1181	3.8025
105	0.1527	3.5137	3.6075	3.5946
106	0.3667	3.8025	3.6489	3.6075

Table 10.9 Varying size of region of interest to calculate elastography dataset (Sr. no. 1 – 25) for SNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
1	0.5458	0.5458	0.5458	0.5624
2	0.4851	0.4173	0.3888	0.5099
3	0.4485	0.4545	0.5316	1.0261
4	0.4048	0.5099	0.5099	0.6694
5	0.4173	1.0261	1.0261	0.6653
6	0.4545	1.0097	1.0097	0.7613
7	0.3888	1.1018	1.1018	1.0076
8	0.5772	0.7613	0.6653	1.0057
9	0.5316	1.0502	0.7613	0.9729
10	0.5099	1.0076	1.0502	1.0059
11	1.0261	1.0057	0.5115	1.138
12	1.0097	0.9729	1.0057	1.1439
13	1.1018	1.0059	1.0059	1.1127
14	0.5014	1.138	1.138	1.0521
15	0.7613	1.1439	1.1439	0.9295
16	1.0502	1.0521	1.8075	1.1076
17	1.0076	0.9295	0.9295	1.1296
18	0.8634	1.1076	1.1076	0.7588
19	1.0057	1.1296	1.1296	0.6915
20	0.9729	0.7588	0.7588	0.6794
21	1.0059	0.6915	0.6915	0.4511
22	1.138	0.6794	0.6794	1.1348
23	1.1439	0.4134	0.4511	1.1118
24	1.1127	0.4511	1.0847	0.8544
25	0.884	1.1348	1.1118	1.1626

Table 10.10 Varying size of region of interest to calculate elastography dataset (Sr. no. 26 – 50) for SNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
26	1.0521	1.0847	0.8544	1.1364
27	0.9295	1.1118	1.1626	0.9585
28	1.1076	1.1364	1.1364	1.1587
29	1.1296	0.9585	0.9585	1.1175
30	0.7588	1.1587	1.1587	0.9602
31	0.7636	1.1175	1.1175	1.093
32	0.6915	0.9602	0.9602	1.1335
33	0.6794	1.093	1.1335	1.0397
34	0.4494	1.1335	1.0397	1.1461
35	0.4134	1.0397	1.1552	1.1552
36	0.4511	1.1461	1.1399	1.1399
37	1.1348	1.1552	1.0414	1.0414
38	1.0847	1.1399	1.1584	1.1526
39	1.0847	1.0414	1.0336	1.1584
40	0.6699	1.1526	0.9016	1.0336
41	0.6533	1.1584	0.5932	0.9016
42	0.6274	1.0336	0.986	0.8663
43	1.1626	0.9016	1.0031	0.5932
44	1.1364	0.8663	1.1115	0.4215
45	0.9585	0.5932	1.2463	0.986
46	1.1587	0.4215	0.8563	1.0031
47	1.1175	0.986	0.8444	1.2463
48	1.5377	1.0031	0.7	0.5918
49	1.1335	1.1115	0.6319	0.7
50	1.0397	1.2463	0.9176	0.9176

Table 10.11 Varying size of region of interest to calculate elastography dataset (Sr. no. 51 – 75) for SNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
51	1.1399	0.8563	0.9265	1.0684
52	1.0414	0.7	1.0684	1.0956
53	1.1584	0.6319	1.0956	1.1385
54	1.0336	1.3407	1.0462	1.1156
55	0.4513	0.9265	1.1156	1.1117
56	0.9016	1.0684	1.1117	0.5289
57	0.8663	1.0956	1.0778	1.0943
58	0.5932	1.1385	1.0943	0.891
59	0.4215	0.6793	0.891	1.0614
60	0.986	1.1156	1.1459	1.1246
61	1.0031	1.1117	0.5669	1.0332
62	1.1115	1.0778	1.1246	1.4325
63	1.2463	3.9715	1.0332	1.0666
64	0.8563	0.891	1.1976	1.0629
65	0.4061	8.2975	1.0666	0.9573
66	0.7	1.1246	1.0629	0.8535
67	0.6319	0.3168	0.9573	0.8872
68	0.9176	1.0666	0.8535	0.945
69	0.4734	1.0629	0.8872	0.8631
70	1.0684	0.9573	0.945	0.9055
71	1.0956	0.8535	0.8631	1.1262
72	1.1156	0.8872	0.9055	0.9195
73	1.1117	0.945	1.1262	1.1353
74	0.622	0.8631	0.9195	0.5974
75	1.0943	0.9055	1.1353	0.5898

Table 10.12 Varying size of region of interest to calculate elastography dataset (Sr. no. 76 – 106) for SNR values and used to generate the results in the section 5.3.2.2.

Sr. no.	w=10 h=10	w=15 h=15	w=20 h=15	w=25 h=15
76	1.1459	0.9195	0.5974	0.5989
77	1.1246	1.1353	0.5989	1.0672
78	1.0332	0.5898	0.9002	1.0429
79	1.1976	0.5989	1.0429	0.8751
80	1.0058	0.9601	0.8751	1.1061
81	1.5123	0.8751	1.1061	1.1213
82	1.1335	1.1061	1.0092	1.1886
83	0.8535	1.1213	1.0758	1.0092
84	0.945	1.1886	1.0569	1.0758
85	0.8631	0.5432	1.0361	1.0569
86	1.1262	1.0758	1.1303	1.0361
87	0.9195	1.0569	1.0775	0.6545
88	1.1353	1.0361	1.1391	1.1391
89	0.5974	1.1303	1.114	1.114
90	0.5898	0.795	1.3502	1.3502
91	0.5989	1.1391	1.1071	1.1071
92	1.0048	1.114	0.5947	0.5947
93	1.0429	1.3502	0.6135	0.9975
94	0.8751	0.9878	0.5608	0.6135
95	1.1061	0.5947	0.508	0.5608
96	1.1213	0.9975	0.8086	0.508
97	1.1886	0.6135	0.8794	0.8086
98	1.0758	0.5608	1.1327	0.8794
99	1.0569	0.508	0.9528	1.1327
100	0.7192	0.8086	0.5524	0.9528
101	1.1391	0.8794	1.3005	0.5524
102	1.114	0.9528	1.2974	1.0775
103	1.3502	1.0168	1.137	1.2974
104	1.4927	0.5524	2.4012	1.137
105	0.5579	1.0775	1.0891	1.1217
106	0.5947	1.137	1.0923	1.0891

Table 10.13 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 1 – 50) for CNR values and used to generate the results in the section 5.3.2.3.

Sr. No.	2 4 2	1 8 1	1 12 1	Sr. No.	2 4 2	1 8 1	1 12 1
1	0.2618	3.1224	0.1029	26	0.2742	2.2093	2.9321
2	0.5932	0.1052	0.1461	27	2.3915	1.7947	0.2742
3	0.1461	0.3226	2.4621	28	2.4422	1.0608	2.3915
4	2.4621	2.3448	0.7023	29	1.8736	1.6875	2.4422
5	2.3756	0.0498	0.1052	30	1.7947	2.1633	1.8736
6	3.1224	0.2383	0.3226	31	1.0608	2.1194	2.2093
7	0.3226	0.6978	1.3703	32	2.1633	1.57	1.7947
8	2.3448	1.6268	0.0157	33	1.57	2.0015	1.0608
9	1.3703	0.9464	0.2383	34	2.0082	2.0082	1.6875
10	0.0157	1.6797	0.6978	35	1.4874	1.4874	2.1633
11	0.2383	0.0592	1.6268	36	2.8579	2.8579	2.1194
12	0.6978	1.7027	0.9464	37	2.9888	2.9888	1.57
13	0.9464	2.0861	1.6797	38	0.0949	3.7848	2.0015
14	0.3561	0.7284	0.0592	39	3.7848	3.9217	2.0082
15	2.0861	0.6997	0.3561	40	3.9217	3.0546	1.4874
16	0.7284	1.9605	1.7027	41	0.6862	0.6862	2.8579
17	0.6997	0.1172	2.0861	42	0.864	0.864	2.9888
18	0.1172	1.8292	0.7284	43	0.9548	0.2649	0.0949
19	2.6734	1.4713	0.6997	44	0.4876	0.4876	0.1283
20	1.8292	2.0011	1.9605	45	0.0773	1.6658	3.7848
21	1.4713	2.9321	0.1172	46	0.323	1.436	3.9217
22	0.0853	0.2742	2.6734	47	1.436	1.7299	0.6862
23	1.0701	2.3915	1.4713	48	1.7299	1.6597	0.073
24	2.0011	2.4422	1.0701	49	1.6597	1.8349	0.2649
25	2.9321	1.8736	2.0011	50	1.3641	2.4035	1.6658

Table 10.14 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 51 – 80) for CNR values and used to generate the results in the section 5.3.2.3.

Sr. No.	2 4 2	1 8 1	1 12 1
51	1.8349	1.4302	1.436
52	2.4035	2.7782	1.7299
53	1.4302	0.9135	1.6597
54	0.2937	2.4581	1.8349
55	0.9135	1.9506	2.4035
56	0.2039	1.5164	0.1774
57	0.165	1.6887	2.7782
58	1.9506	0.2231	0.9135
59	1.5164	1.79	0.2065
60	1.6887	2.6018	1.9506
61	1.79	2.3245	1.5164
62	2.6018	4.5279	0.1767
63	2.3245	3.5382	0.2231
64	4.5279	3.1633	1.79
65	4.5245	0.9829	2.6018
66	3.5382	0.144	2.3245
67	2.6064	0.1453	4.5279
68	1.5597	0.0678	4.5245
69	3.1633	4.1361	3.5382
70	0.9829	0.3554	2.6064
71	2.8279	0.7026	3.1633
72	0.144	2.6896	0.9829
73	0.0678	2.927	2.8279
74	4.1361	2.0489	0.144
75	0.3554	0.3616	0.1453
76	0.7026	1.7247	0.0678
77	2.6896	2.1416	4.1361
78	2.927	2.6025	0.3554
79	1.3264	0.1627	0.7026
80	1.2976	2.3342	2.6896

Table 10.15 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 1 – 50) for CNR values and used to generate the results in the section 5.3.2.3.

Sr. No.	2 4 2	1 8 1	1 12 1	Sr. No.	2 4 2	1 8 1	1 12 1
1	0.5458	1.1018	0.5624	26	0.9585	1.093	1.1364
2	0.5772	0.6653	0.5099	27	1.1587	1.1335	0.9585
3	0.5099	0.7613	1.0261	28	1.1175	1.0397	1.1587
4	1.0261	1.0502	0.6694	29	0.9602	1.1461	1.1175
5	1.0097	0.4589	0.6653	30	1.1335	1.1552	0.9602
6	1.1018	0.9729	0.7613	31	1.0397	1.1399	1.093
7	0.7613	1.0059	1.0076	32	1.1552	1.0414	1.1335
8	1.0502	1.138	1.0057	33	1.0414	1.1526	1.0397
9	1.0076	1.1439	0.9729	34	1.1584	1.1584	1.1461
10	1.0057	1.1127	1.0059	35	1.0336	1.0336	1.1552
11	0.9729	1.0521	1.138	36	0.9016	0.9016	1.1399
12	1.0059	1.1076	1.1439	37	0.8663	0.8663	1.0414
13	1.1439	1.1296	1.1127	38	0.5932	0.986	1.1526
14	0.9295	0.7588	1.0521	39	0.986	1.0031	1.1584
15	1.1296	0.6915	0.9295	40	1.0031	1.1115	1.0336
16	0.7588	0.6794	1.1076	41	1.2463	1.2463	0.9016
17	0.6915	0.4511	1.1296	42	0.8563	0.8563	0.8663
18	0.4511	1.0847	0.7588	43	0.8444	0.7	0.5932
19	1.1348	1.1118	0.6915	44	0.6319	0.6319	0.4215
20	1.0847	1.1626	0.6794	45	1.194	0.9176	0.986
21	1.1118	1.1364	0.4511	46	0.4432	1.0684	1.0031
22	0.6699	0.9585	1.1348	47	1.0684	1.0956	1.2463
23	0.8544	1.1587	1.1118	48	1.0956	1.1385	0.5918
24	1.1626	1.1175	0.8544	49	1.1385	1.1156	0.7
25	1.1364	0.9602	1.1626	50	1.0462	1.1117	0.9176

Table 10.16 Varying weight for the region of interest to calculate elastography dataset (Sr. no. 51 – 80) for CNR values and used to generate the results in the section 5.3.2.3.

Sr. No.	2 4 2	1 8 1	1 12 1
51	1.1156	1.0778	1.0684
52	1.1117	1.0943	1.0956
53	1.0778	0.891	1.1385
54	3.4623	1.1459	1.1156
55	0.891	1.1246	1.1117
56	2.0939	1.0332	0.5289
57	0.5654	1.1976	1.0943
58	1.1246	1.0666	0.891
59	1.0332	1.0629	1.0614
60	1.1976	0.9573	1.1246
61	1.0629	0.8535	1.0332
62	0.9573	0.8872	1.4325
63	0.8535	0.8631	1.0666
64	0.8872	1.1262	1.0629
65	0.945	0.9195	0.9573
66	0.8631	0.5974	0.8535
67	0.9055	0.5898	0.8872
68	1.023	0.5989	0.945
69	1.1262	1.0672	0.8631
70	0.9195	1.0429	0.9055
71	1.1353	0.8751	1.1262
72	0.5974	1.1061	0.9195
73	0.5989	1.1213	1.1353
74	1.0672	1.1886	0.5974
75	1.0429	1.0361	0.5898
76	0.8751	1.0775	0.5989
77	1.1061	1.1391	1.0672
78	1.1213	1.114	1.0429
79	1.0092	1.3502	0.8751
80	1.0758	1.1071	1.1061

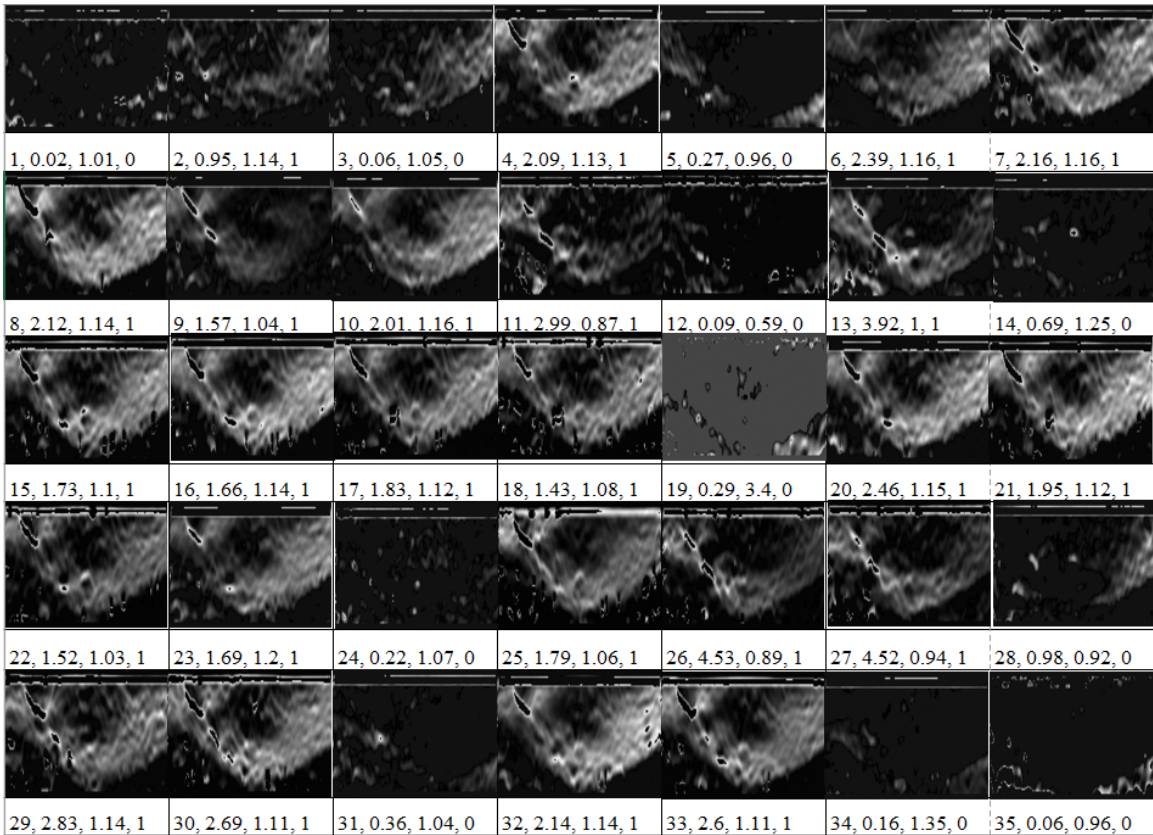


Figure 10.1 Raw dataset for visual classification of filtered elastography dataset (1-35): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

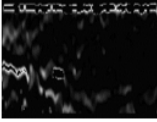
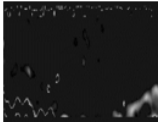
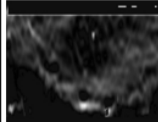
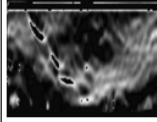
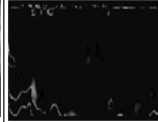
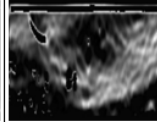
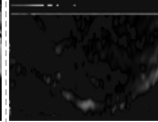
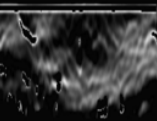
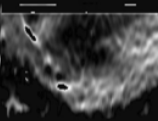
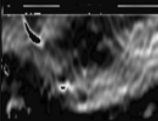
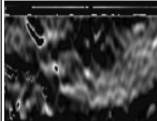
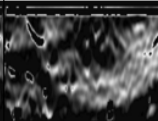
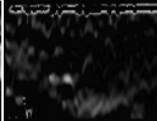
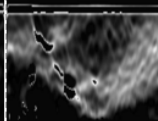
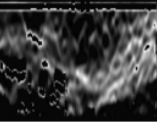
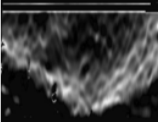
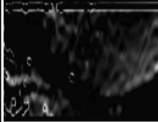
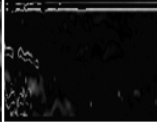
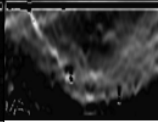
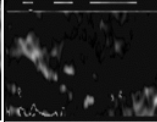
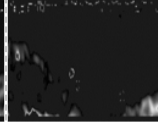
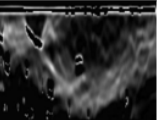
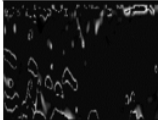
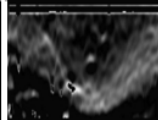
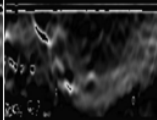
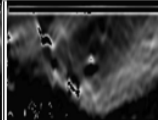
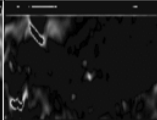
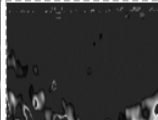
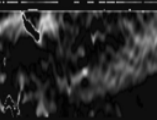
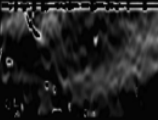
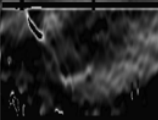
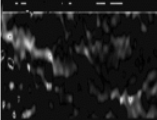
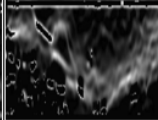
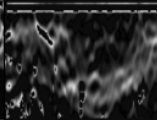
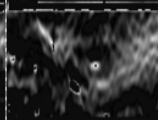
						
36, 0.12, 0.56, 0	37, 0.2, 1.3, 0	38, 2.34, 1.3, 1	39, 3.8, 1.14, 1	40, 0.25, 0.86, 0	41, 3.65, 1.09, 1	42, 0.34, 1.19, 0
						
43, 3.09, 1.06, 1	44, 3.26, 1.19, 1	45, 3.3, 1.17, 1	46, 3.12, 1.16, 1	47, 2.63, 1.07, 1	48, 0.67, 0.68, 0	49, 4.75, 1.17, 1
						
50, 3.23, 1.03, 1	51, 4.37, 1.12, 1	52, 1.6, 0.84, 0	53, 0.44, 0.5, 0	54, 1.57, 1.12, 1	55, 0.06, 0.91, 0	56, 0.12, 1.26, 0
						
57, 5.4, 1.01, 1	58, 0.24, 0.63, 0	59, 3.82, 1.11, 1	60, 4.58, 0.98, 1	61, 4.57, 1.11, 1	62, 0.39, 1, 0	63, 0.16, 1.68, 0
						
64, 1.8, 1, 1	65, 1.46, 0.9, 1	66, 3.95, 1.02, 1	67, 0.27, 0.94, 0	68, 3.2, 0.95, 1	69, 3.29, 1.04, 1	70, 1.02, 1.05, 1

Figure 10.2 Raw dataset for visual classification of filtered elastography dataset (36-70): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

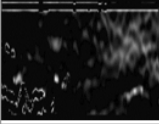
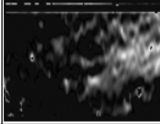
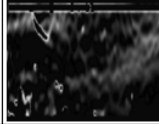
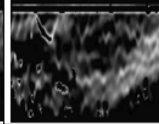
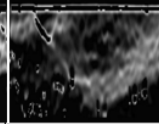
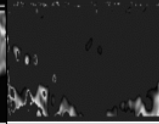
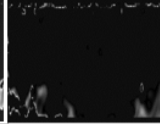
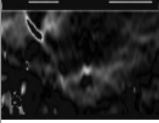
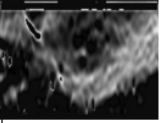
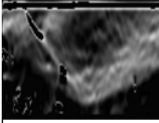
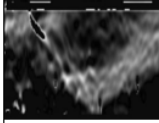
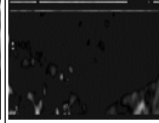
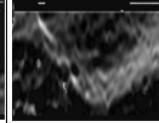
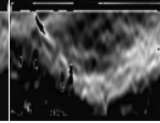
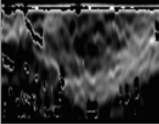
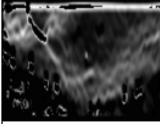
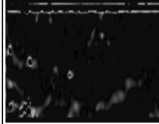
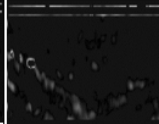
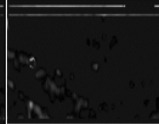
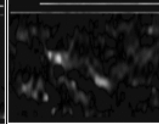

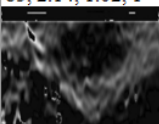
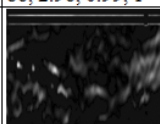
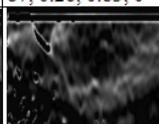
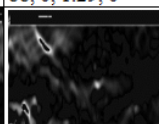
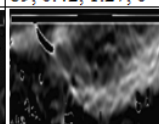
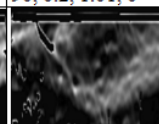
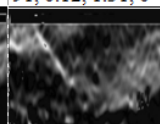
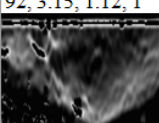
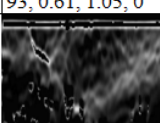
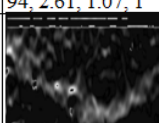
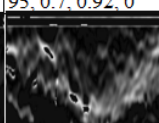
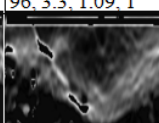
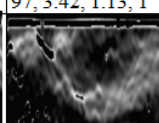
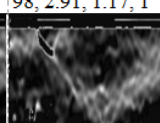
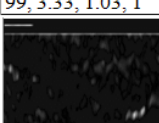
						
71, 1.13, 0.8, 0	72, 0.18, 0.94, 0	73, 0.82, 0.93, 0	74, 2.62, 1.01, 1	75, 2.71, 1.02, 1	76, 0.22, 1.37, 0	77, 0.18, 1, 0
						
78, 2.38, 1.01, 1	79, 3.26, 1.15, 1	80, 3.61, 1.12, 1	81, 3.83, 1.13, 1	82, 0.37, 1.22, 0	83, 3.28, 1.17, 1	84, 2.37, 1.11, 1
						
85, 2.14, 1.02, 1	86, 2.96, 0.99, 1	87, 0.26, 0.83, 0	88, 0, 1.29, 0	89, 0.42, 1.27, 0	90, 0.2, 1.01, 0	91, 0.12, 1.31, 0
						
92, 3.15, 1.12, 1	93, 0.61, 1.05, 0	94, 2.61, 1.07, 1	95, 0.7, 0.92, 0	96, 3.3, 1.09, 1	97, 3.42, 1.13, 1	98, 2.91, 1.17, 1
						
99, 3.33, 1.03, 1	100, 0.65, 1, 1	101, 1.02, 0.92, 1	102, 3.06, 1.08, 1	103, 3.63, 1.15, 1	104, 3.66, 1.11, 1	105, 3.29, 1.16, 1
						
106, 0.51, 1.18, 0						

Figure 10.3 Raw dataset for visual classification of filtered elastography dataset (71-106): This figure shows the dataset classified by visual inspection on the resulting output of filtered elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

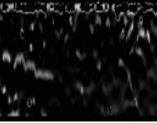
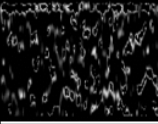
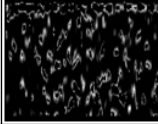
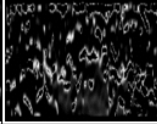
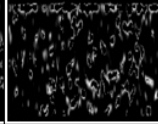
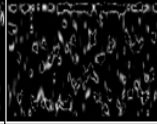
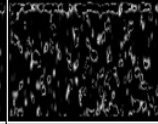
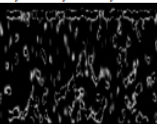
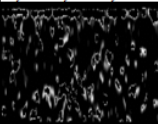
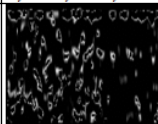

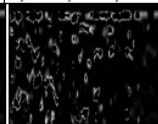
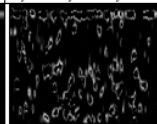
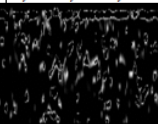
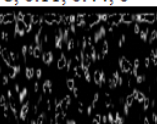
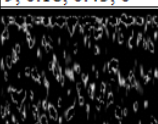
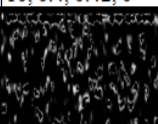
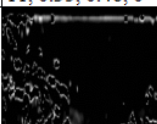
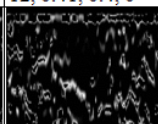
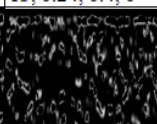
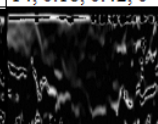
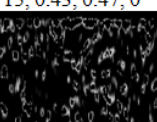
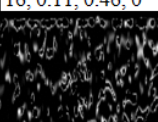
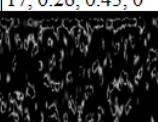
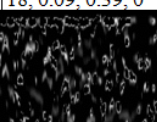
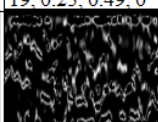
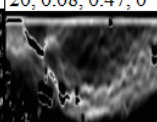
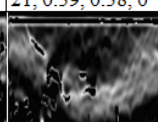
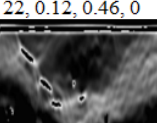
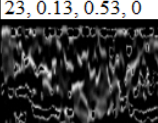
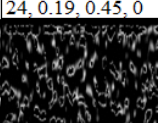
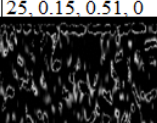
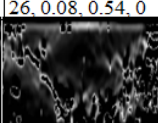
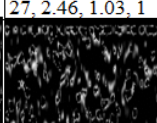
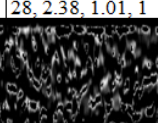
						
1, 0.26, 0.55, 0	2, 0.22, 0.49, 0	3, 0.19, 0.46, 0	4, 0.35, 0.55, 0	5, 0.32, 0.45, 0	6, 0.17, 0.44, 0	7, 0.03, 0.45, 0
						
8, 0.11, 0.44, 0	9, 0.18, 0.45, 0	10, 0.4, 0.42, 0	11, 0.35, 0.46, 0	12, 0.41, 0.4, 0	13, 0.24, 0.4, 0	14, 0.16, 0.42, 0
						
15, 0.43, 0.47, 0	16, 0.11, 0.46, 0	17, 0.26, 0.45, 0	18, 0.09, 0.39, 0	19, 0.25, 0.49, 0	20, 0.08, 0.47, 0	21, 0.59, 0.58, 0
						
22, 0.12, 0.46, 0	23, 0.13, 0.53, 0	24, 0.19, 0.45, 0	25, 0.15, 0.51, 0	26, 0.08, 0.54, 0	27, 2.46, 1.03, 1	28, 2.38, 1.01, 1
						
29, 3.12, 1.1, 1	30, 0.11, 0.67, 0	31, 0.3, 0.54, 0	32, 0.13, 0.5, 0	33, 0.18, 0.72, 0	34, 0.02, 0.5, 0	35, 0.08, 0.62, 0

Figure 10.4 Raw dataset for visual classification of freehand elastography dataset (1-35): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

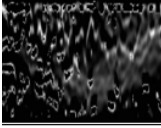
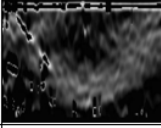
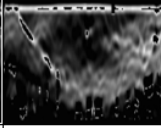
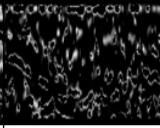
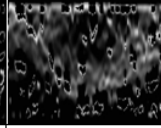
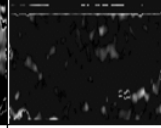
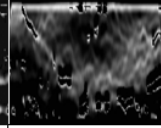
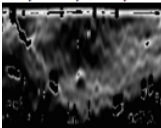
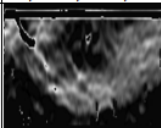
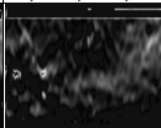
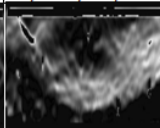
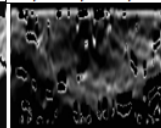
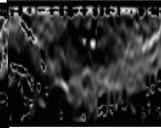
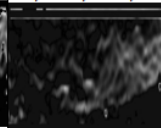
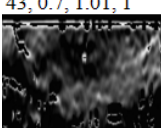
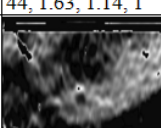
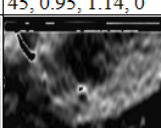
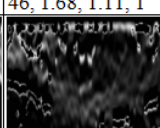
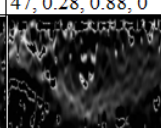
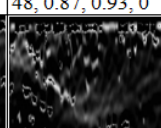
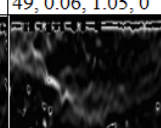
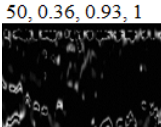
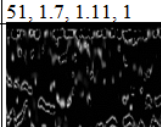
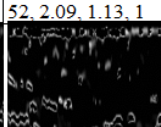
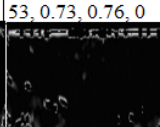
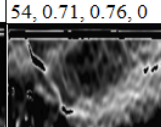
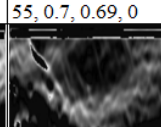
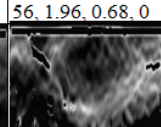
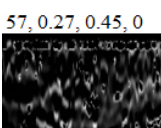
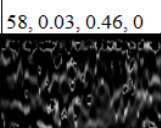
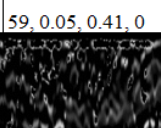
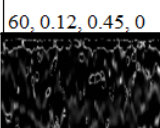
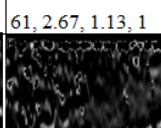
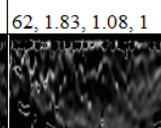
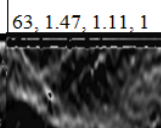
						
36, 0.32, 0.76, 0	37, 2.34, 1.05, 1	38, 1.37, 1.01, 1	39, 0.03, 0.5, 0	40, 0.34, 0.86, 0	41, 0.02, 1.01, 0	42, 0.24, 0.97, 1
						
43, 0.7, 1.01, 1	44, 1.63, 1.14, 1	45, 0.95, 1.14, 0	46, 1.68, 1.11, 1	47, 0.28, 0.88, 0	48, 0.87, 0.93, 0	49, 0.06, 1.05, 0
						
50, 0.36, 0.93, 1	51, 1.7, 1.11, 1	52, 2.09, 1.13, 1	53, 0.73, 0.76, 0	54, 0.71, 0.76, 0	55, 0.7, 0.69, 0	56, 1.96, 0.68, 0
						
57, 0.27, 0.45, 0	58, 0.03, 0.46, 0	59, 0.05, 0.41, 0	60, 0.12, 0.45, 0	61, 2.67, 1.13, 1	62, 1.83, 1.08, 1	63, 1.47, 1.11, 1
						
64, 0.09, 0.67, 0	65, 0.51, 0.65, 0	66, 0.38, 0.63, 0	67, 0.02, 0.57, 0	68, 0.38, 0.63, 0	69, 1.07, 0.85, 0	70, 2, 1.16, 1

Figure 10.5 Raw dataset for visual classification of freehand elastography dataset (36-70): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

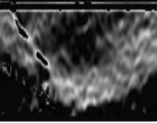
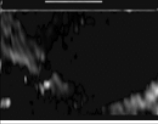
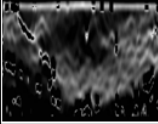
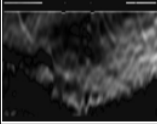
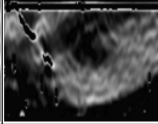
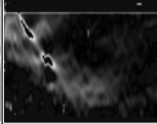
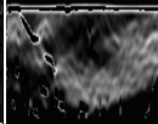
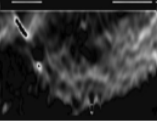
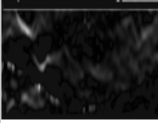
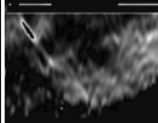
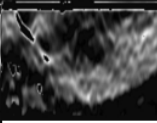

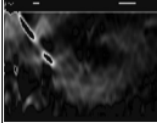
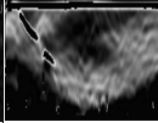
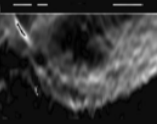
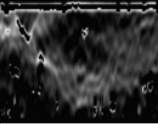
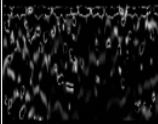
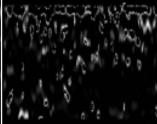
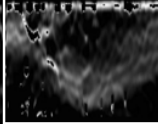
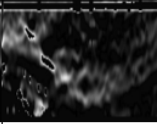
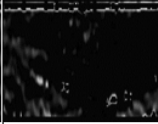
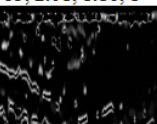
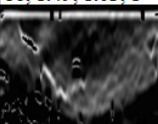
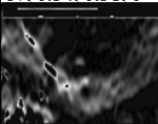
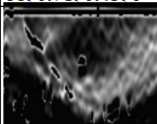

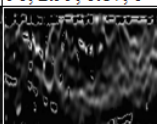
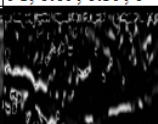
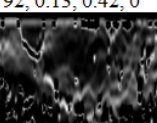
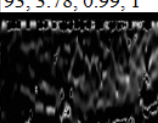
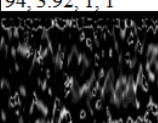
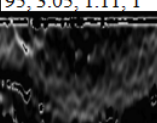
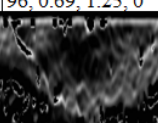
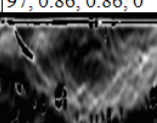
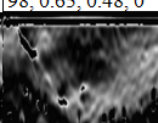
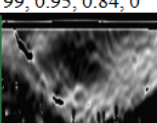
						
71, 2.93, 1.14, 1	72, 0.27, 0.96, 0	73, 1.8, 0.98, 1	74, 2.39, 1.16, 1	75, 2.44, 1.12, 1	76, 1.87, 0.96, 0	77, 2.21, 1.09, 1
						
78, 1.79, 1.13, 1	79, 1.06, 1.04, 0	80, 1.69, 1.15, 1	81, 2.16, 1.16, 1	82, 2.12, 1.14, 1	83, 1.57, 1.04, 0	84, 2, 1.15, 1
						
85, 2.01, 1.16, 1	86, 1.49, 1.03, 1	87, 0.54, 0.51, 0	88, 0.71, 0.45, 0	89, 2.86, 0.9, 0	90, 2.99, 0.87, 0	91, 0.09, 0.59, 0
						
92, 0.13, 0.42, 0	93, 3.78, 0.99, 1	94, 3.92, 1, 1	95, 3.05, 1.11, 1	96, 0.69, 1.25, 0	97, 0.86, 0.86, 0	98, 0.65, 0.48, 0
						
99, 0.95, 0.84, 0	100, 0.26, 0.7, 0	101, 0.49, 0.63, 0	102, 1.67, 0.92, 0	103, 0.96, 0.93, 0	104, 1.44, 1.07, 1	105, 1.73, 1.1, 1
						
106, 1.66, 1.14, 1						

Figure 10.6 Raw dataset for visual classification of freehand elastography dataset (71-106): This figure shows the dataset classified by visual inspection on the resulting output of freehand elastography algorithm. The tuples below each image are (Image sequence number, CNR value, SNR value, group). The group contains value 0 or 1, where 1 indicates that the image appears to be a good quality elastogram as per the reference user and 0 indicates that the image appears to be a bad elastogram.

Vita

Nishikant Deshmukh was born in Amravati, Maharashtra, India. He completed his Bachelor of Engineering in Computer Engineering from University of Pune, 2005. He worked at the National Stock Exchange of India as an Associate System Analyst from 2005-2008. In 2008, Nishikant enrolled in Master's program at the Department of Computer Science at The Johns Hopkins University. His background is in the field of Parallel and Distributed Systems. He got interested in the field of Computer Integrated Surgery and Ultrasound research. Noticing the potential advantages of integrating High Performance Computing and Medical Imaging, Nishikant started his Ph.D. studies in the Department of Computer Science at The Johns Hopkins University from 2010. Nishikant has implemented the world's first known Five-Dimensional Ultrasound System. Because of his research and academic excellence, Nishikant was selected in the Heidelberg Laureate Forum, Germany, 2015 as the top 80 Young Investigators world-wide. In 2015, Nishikant was selected as an INK Fellow.