

Characterising carotid atherosclerotic plaques by creating computational and phantom ultrasound models

Neuphoria Group

Department of Bioengineering
Imperial College London

*A Project Report Submitted in
Partial Fulfilment of the*

MEng — Degree

Authors: Yangqi Zhu, Victoria Walker, Stefano Elias, Yiran Sun,
Yangkun Chen, Laetitia Bacha, Ricardo Zhang

Supervisor:
Professor Tang
mengxing.tang@imperial.ac.uk
Department of Bioengineering

Word count: 4987

April, 2025

Contents

1. INTRODUCTION	4
1.1. Background	4
1.2. Main Objectives	5
2. METHODS	6
2.1. Computational Method	6
2.1.1. Computational method introduction	6
2.1.2. Scatterer Positions and Amplitudes (Appendix A.4)	7
2.1.3. Ultrasound Processing (Appendix A.5)	7
2.1.4. Image Plotting (Appendix A.6)	8
2.1.5. Microvessel Visualisation Using Microbubbles	8
2.1.6. Ultrasound Image Alignment and 3D Volume Construction (Appendix 7)	8
2.2. Experimental Method	9
2.2.1. Mould 3D Printing	9
2.2.2. Laboratory Casting Method	9
2.2.3. 3D Ultrasound Testing	10
3. RESULTS	11
3.1. Computational Results	11
3.1.1. Static 2D Image Simulation	11
3.1.2. Dynamic 2D Simulation of a microbubble flowing through a microvessel	12
3.1.3. The 3D Model of the Carotid Artery Simulation	12
3.1.4. Qualitative Assessment	13
3.2. Experimental Results	13
4. DISCUSSION	14
4.1. Discussion of this study's computational research and findings	14
4.1.1. Quantifying the relationship between image quality and the number of scatterers	14
4.1.2. Quantifying the relationship between image quality and the number of lines	14
4.1.3. The usage and limitation of 2D moving simulation	15
4.1.4. Limitations of 3D simulation	15
4.1.5. Qualitative assessment	15
4.2. Discussion of this study's experimental research and findings	15
4.2.1. Initial Phantom Results	15
4.2.2. Limitations of the phantom fabrication process	16
5. CONCLUSION	16
6. Acknowledgments	16
A. Appendix	20
A.1. Project Management Assessment	20
A.2. Key Project Management Lessons	20
A.2.1. Efficient Communication	20
A.2.2. Positive Teamwork Environment, Collaboration and Adaptability	20
A.2.3. Defined Milestones and Continuous Performance Tracking	21
A.3. Execution.m MATLAB code	21
A.4. Scatterer Positions and Amplitudes MATLAB code	23
A.5. Ultrasound Processing MATLAB code	24
A.6. Image plotting MATLAB code	25
A.7. Ultrasound Image Alignment and 3D Volume Construction MATLAB code	26

A.8. processed_bubble MATLAB code 29

Abstract

Ultrasound Localisation Microscopy (ULM) is as a promising emerging solution for the imaging of atherosclerotic plaques. By precisely tracking and localising individual microbubbles to achieve micron-scale spatial resolution of plaque microvasculature, ULM can visualise microvascular features that are indiscernible with conventional modalities. However, presently ULM remains in the early stages of clinical integration, as transitioning from research to widespread clinical settings requires standardised protocols. Thus, we sought to establish a ground truth dataset for ULM which would aid the aforementioned shift. By devising a bilateral strategy that consists of developing, separately, computational simulations and phantom models of vascularised atherosclerotic plaques in carotid arteries, we obtained validated results that can be used for algorithm development. Computationally, we used Field II to simulate 2D ultrasound images of microbubbles flowing through microvessels within a carotid plaque and investigated how scatterer density and scan line count affect image quality and reliability. These dynamic simulations were then used to construct a static 3D ultrasound image of the vascular region for enhanced visualisation. Experimentally, a tissue-mimicking PVA carotid phantom with an embedded plaque and small vessel was fabricated and tested under ultrasound with microbubble contrast agents. This validated the phantom's ability to replicate acoustic features of vasculature and flow dynamics. Together, the results provide a framework for optimising ULM imaging parameters and serve as reference models to guide future algorithm development and phantom-based calibration.

1. INTRODUCTION

1.1. Background

Cardiovascular diseases (CVD), of which coronary heart disease (CHD) is most prevalent, are the leading cause of mortality globally — responsible for an estimated 18 million annual deaths [1]. CVD is characterised by chronic arterial narrowing due to the buildup of lipid plaques within the intimal wall which predispose acute ischaemic events, including strokes and myocardial infarctions [2]. To diagnose the risk of acute ischaemic events occurring, assessing morphological plaque components, especially the degree of neovascularisation, is of paramount importance. Carotid atherosclerotic plaques are often selected to model this because they are considered a surrogate marker for CHD, as well as lying more superficially which facilitates clearer ultrasound imaging [3].

Currently, the most common tools for imaging plaque components are conventional contrast-enhanced ultrasound imaging techniques, such as Brightness-mode (B-mode) ultrasounds with

added contrast agents. B-mode works by transmitting pulsed acoustic waves into tissue and reconstructing a two-dimensional grayscale image by mapping the amplitude of backscattered echoes. Although moderately effective at visualising plaques, ultrasound diffraction limits resolution to approximately half a wavelength [4]. The resolution limit of ultrasound transducers, calculated using the wave equation $\lambda = \frac{v}{f}$, depends on the speed of sound in artery walls (around 1600 m/s) [5], with vascular ultrasonography typically using linear array transducers operating at frequencies between 5 - 12 MHz[5, 6]. Thus, the resolvable range for two distinct point-sources using conventional ultrasound is approximately $133 - 320 \mu\text{m}$, insufficient for robustly visualising microvessels, which range from $20 - 300 \mu\text{m}$ [7].

Super Resolution Ultrasound (SRU), a relatively novel imaging technique, facilitates the distinction of objects at resolutions beyond the classic diffraction limit of half a wavelength [7]. By surpassing the theoretical resolution limit, key structural features of plaques, such as microvessels, can be imaged much more reliably,

hence overcoming the constraints of conventional ultrasound. Ultrasound Localisation Microscopy (ULM) - a type of SRU - precisely localises and tracks individual microbubbles flowing through the vascular networks [8]. By obtaining a temporal accumulation of the microbubbles' localisation through ULM, a high-resolution image of the tissue's vascular morphology is created [9].

Microbubbles are small gas-filled spheres, usually administered intravenously, that act as contrast agents by oscillating nonlinearly in response to emitted acoustic waves, consequently scattering significant amounts of energy and generating bright spots that enhance the optical clarity of the surrounding tissue or plaque [10, 11]. Unlike Contrast-Enhanced Ultrasound (CEUS), where microbubbles' primary purpose is simply to increase contrast, in ULM the microbubbles are treated as the scatterers. Hence, by accumulating numerous images of microbubbles flowing through the vasculature, the microbubbles can be localised –from which a super-resolved image can be generated [7]. The sequence of steps in ULM image processing is shown in Figure 1.

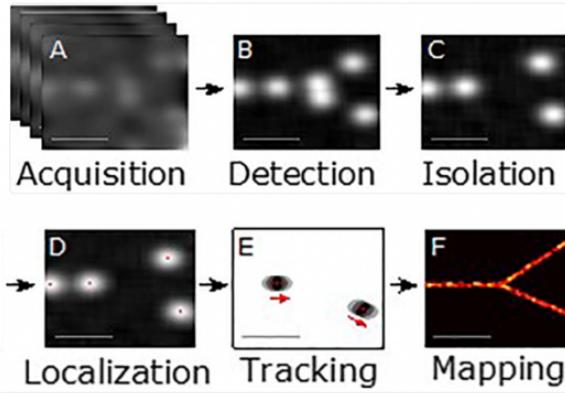


Figure 1: Flow-chart outlining the steps in ULM (SRU) image processing [4].

Despite its promising diagnostic potential in imaging tissues' structural components, ULM is not yet widely used in clinical settings. The main alternative is CEUS, which is quicker, less technically demanding, and provides real-time imaging. However, ULM offers significantly better resolution and reliable tissue perfusion quantification, making it valuable for establishing conclusive patient diagnoses [12, 13]. Therefore, overcoming the barriers to its clinical implementation is crucial.

To address ULM's technical complexity, establishing a ground truth for the video acquisition stage (step A in Figure 1) and microbubble location is essential. As the video acquisition stage involves acquiring 'classic' high-frame-rate ultrasound images – often B-mode – of microbubbles flowing in the vasculature, having a benchmark for this would guide researchers and clinicians in evaluating gathered data [7].

While ULM has improved microvasculature imaging, it still faces limitations, such as susceptibility to out-of-plane motion, long scan times, and the challenge of 2D projections of 3D structures, which can lead to plaque measurement errors [13]. 3D ultrasound imaging overcomes some of these barriers by enabling detailed volumetric mapping of plaque microvasculature. However, as *in vivo* ground truth vessel maps are generally unavailable [14], establishing a 3D ultrasound ground truth would aid in determining imaging parameters and provide a controlled testbed for evaluating ULM algorithms [14][15].

1.2. Main Objectives

Aim

The overarching aim of this study was to establish a ground truth set of images, in both 2D and 3D, that accurately represent ultrasound imaging of a microbubble flowing through the microvasculature of an atherosclerotic plaque within the carotid artery. By creating this controlled and well-defined dataset, where the presence and characteristics of the plaque are certain, the study aims to provide a ground truth for the development of ULM imaging and for evaluating the accuracy of the processing algorithms (such as localisation algorithm). To achieve this aim, a dual strategy was devised, comprising the following two objectives.

Objective 1

The first part of our bilateral strategy was to obtain accurate computational ultrasound simulations that can serve as benchmarks, generated in a controlled environment with easily adjustable parameters. Specifically, the goal was to secure a 2D ultrasound simulation (in

B-mode) tracking the microbubble's flow through the carotid artery's plaque microvasculature, and then to demonstrate a method for converting this 2D simulation into a 3D simulation for improved visualisation. The purpose of this ground truth model is to enable the fine-tuning of: the data acquisition method (interface between the ultrasound transducer (probe) and the computer system), and the data processing algorithms, to improve image accuracy and reliability.

Objective 2

The second facet of our strategy was to create an experimental ground truth ultrasound simulation by testing on a robust, accurate, and reusable carotid artery phantom. By designing our own phantoms, we acquired practical and realistic results that account for noise and imperfections present in non-ideal environments.

2. METHODS

2.1. Computational Method

To obtain the dynamic 2D ultrasound simulation (in B-mode) tracking the microbubble's flow through the carotid artery's plaque microvasculature, as well the static 3D ultrasound image of the microvasculature, the following method was pursued.

2.1.1. Computational method introduction

A MATLAB toolbox, Field II, was used to simulate ultrasound behaviour. Using linear systems theory, the program allows users to simulate both pulsed and continuous wave ultrasound fields for arbitrary transducer geometries, and any apodization scheme can be simulated [16, 17, 18]. A program, `Execution.m` (see Appendix A.3), was developed to automatically execute the entire simulation. It contains all global variables, which can be adjusted as needed. The variables are based on the parameters of the GE L3-12-D Linear Array ultrasound transducer [19] reported in Table 1, which is used in Professor MengXing Tang's laboratory at Imperial College London. The overall workflow is illustrated in Figure 2.

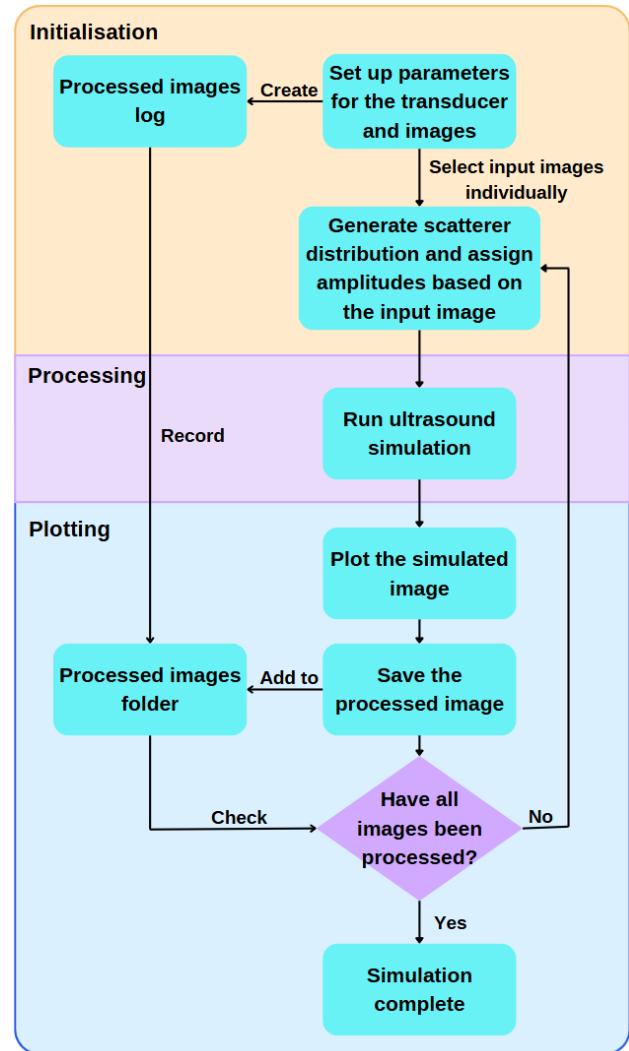


Figure 2: Flow chart explaining the MATLAB code `Execution.m`.

Parameters	Simulation Value
Transducer centre frequency f_0 (Hz)	6.5×10^6
Sampling frequency f_s (Hz)	100×10^6
Speed of sound c (m/s)	1540
Wavelength λ (m)	2.3692×10^{-4}
Element pitch (m)	2.4×10^{-4}
Kerf (gap between elements) (m)	0
Element width (m)	1.5×10^{-4}
Element height (m)	10^{-2}
Number of elements	256
Number of active elements	128
Focus [x, y, z] (m)	[0, 0, 0.035]
Phantom width (m)	0.027
Phantom depth (m)	0.002
Phantom thickness (m)	0.003
Phantom start depth (m)	0.001

Table 1: Summary of the simulation parameters used for the generation of the computational phantom images.

2.1.2. Scatterer Positions and Amplitudes (Appendix A.4)

2.12a Source Image

A carotid plaque ultrasound image [20] was used to generate the scatterer positions and amplitudes. In real applications, users can design scatterers according to their own requirements, rather than generating them from an existing ultrasound image.

2.12b Scatterer Position

Each scatterer contains information about its position and the reflection intensity of the tissue it represents. In this project, a 27×20 mm 2 scatterer map was used to match the input image, and 20,000 scatterers were placed arbitrarily to simulate the natural randomness of ultrasound reflections.

2.12c Scatterer Amplitude

Since the input image is a grayscale ultrasound image, pixel intensity corresponds to the tissue's ultrasound reflection strength—higher pixel values indicate stronger reflection. The amplitude of each scatterer was based on the normalized grayscale value of its corresponding pixel intensity in the input image. Additionally, random noise was introduced to simulate real-world ultrasound reflection variability.

2.1.3. Ultrasound Processing (Appendix A.5)

2.13a Transducer Setting

Based on the GE L3-12-D model, a 6.5 MHz centre frequency was used. The transmit and receive apertures were configured with 256 elements, each with a width of 0.15 mm, a height of 5 mm, and no subdivision. The kerf between elements and the elevation were set to 0.05 mm and 22 mm respectively.

2.13b Signal Generation

The transducer's excitation signal and impulse response were defined as sinusoidal waveforms modulated by a Hanning window [21, 22]. The

centre frequency ($f_0 = 6.5$ MHz) and sampling frequency ($f_s = 100$ MHz) [23] were used to control the temporal resolution and bandwidth of the system. These signals were assigned to both the transmit and receive apertures using Field II's `xdc_excitation()` and `xdc_impulse()` functions [24]. A loop was executed across all scan lines (denoted `no_lines`) for the subsequent processing.

2.13c Dynamic Focusing

The transducer elements were located on the surface of the tissue, with the centre focus defined at $z = 0$. Multiple focal zones were defined along the axial direction, evenly distributed across the tissue region covered in the input image. For transmit focusing, a fixed focal point was used at the deepest zone to avoid repeated transmissions at each depth, which would be time-consuming. For receive focusing, time delays for each focal depth were calculated and applied dynamically across all elements, improving lateral resolution throughout the image.

2.13d Apodization

To reduce sidelobes in the beam pattern and improve lateral resolution in the B-mode ultrasound image, dynamic apodization was applied to both the transmit and receive apertures using Field II's `xdc_apodization()` function. A Hanning window was used to taper the amplitude of the active transducer elements for each scan line. The active aperture was centred laterally with respect to the imaging direction. The number of inactive elements before (N_{pre}) and after (N_{post}) the active aperture was computed dynamically based on the lateral position x of the current scan line. The apodization vector was constructed as shown in Figure 3.

$$\text{apo_vector} = [\underbrace{0, \dots, 0}_{N_{\text{pre}}}, \underbrace{\text{Hanning window}}_{N_{\text{active}}}, \underbrace{0, \dots, 0}_{N_{\text{post}}}]$$

Figure 3: Illustration of the Apodization Vector.

The apodization vector was trimmed or padded as needed to match the total number of transducer elements.

2.13e Simulate Scattering

The `calc_scat()` function was used to simulate the backscattered Radio Frequency (RF) signals from the phantom for each scan line [25]. The resulting RF data and the corresponding starting time (t_{start}) were saved to individual files for each line to support future image reconstruction. To facilitate simulations on multiple computers, a check was implemented to skip scan lines that had already been processed.

2.1.4. Image Plotting (Appendix A.6)

The image was reconstructed from the previously acquired RF data. To align the signals in time, a zero-padding step was applied to account for any offset introduced by t_{start} . The signal was processed using the Hilbert transform, and the envelope was extracted by taking the magnitude of the resulting complex values [26].

To map the large dynamic range of the envelope data to the limited grayscale range of a display, logarithmic compression was applied [27]. The compressed image was then scaled using a range of 45 dB - 50 dB to match the expected signal variation within the phantom [28]. The resulting values were normalised to an 8-bit grayscale range (0–127).

To enhance lateral smoothness and visual quality, linear interpolation was applied laterally with a factor of 20. Axes were scaled to reflect physical dimensions in millimetres. The lateral axis was computed based on the scan line spacing (d_x) and the interpolation factor. The axial axis was derived from the adjusted sampling frequency and the known speed of sound in tissue (c), converting time to depth. The display was restricted to match the lateral and axial extents of the simulated tissue.

2.1.5. Microvessel Visualisation Using Microbubbles

To simulate a high-contrast individual microbubble flowing through the plaque's microvasculature, a high-amplitude scatterer ($\text{amp} = 3 \times 10^6$) was placed within the scatterer distribution and then processed.

By combining microbubbles at different positions into a video, the movement of the microbubble was visualised, thereby tracking the trajectory of the microvessel.

2.1.6. Ultrasound Image Alignment and 3D Volume Construction (Appendix 7)

A set of longitudinal ultrasound images and their corresponding binary expert masks were processed to construct a spatially aligned 3D ultrasound volume as shown in Figure 4 [29].

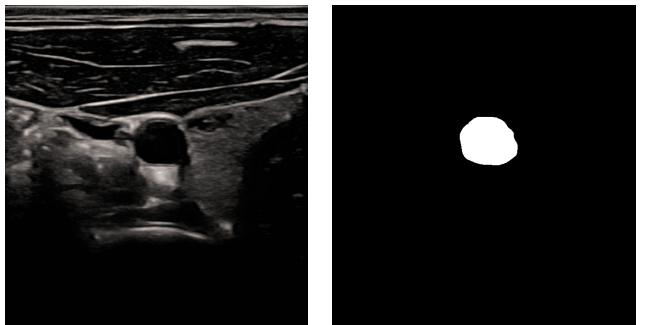


Figure 4: A longitudinal ultrasound images and its corresponding binary expert masks.

For each mask, the centroid of the region of interest was calculated. If multiple regions were present, the largest was selected; if no foreground pixels were found, the image centre was used. The average of all centroids defined a common reference point.

Each ultrasound image was then shifted to align its centroid with this reference, and these shifts determined the canvas size needed to fit all images without cropping. Each image was placed accordingly, producing a set of aligned 2D ultrasound slices.

The aligned slices were then stacked along the third dimension to form a 3D volume as shown in Figure 5 [30].

To focus on the anatomical region and eliminate empty space, the volume was centrally cropped using a configurable crop factor (e.g. 50%).

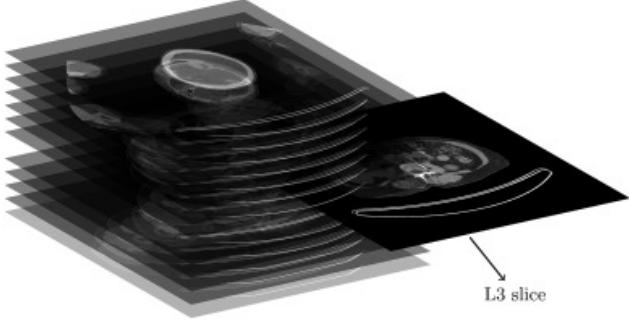


Figure 5: Demonstration of stacking 2D ultrasound images to form a 3D model.

2.2. Experimental Method

To achieve the second objective, a method was developed to create a realistic and practical phantom model.

2.2.1. Mould 3D Printing

The arterial phantom mould was designed in Blender for 3D printing (Figure 6).

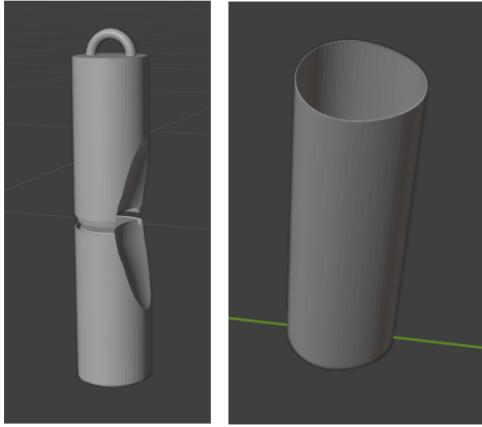


Figure 6: 3D model of the inner (left) and outer (right) shells of the mould in Blender.

It consists of an outer shell and a two-part inner shell, split for easy removal after silicone moulding. The outer shell is elliptical with a raised section simulating the skin barrier between the ultrasound probe and the artery. The inner shell includes a notch to mimic arterial plaque. A tenon-and-mortise joint ensured precise alignment, stability, and easy assembly/disassembly during fabrication.

Component	Parameter	Size (mm)
Inner Shell	Diameter	38.00
	Length	157.78
Outer Shell	Inner Diameter	67.26
	Outer Diameter	77.28
	Length	152.63
Plaque	Skin Layer Length	12.88
	Diameter	8.63

Table 2: Dimensions of the final 3D printed mould.

The final dimensions of the mould are shown in Table 2. The material was sliced using Orcaslicer software, and then printed into the mould shape on a Bambu 3D printer using PLA [31].

2.2.2. Laboratory Casting Method

The phantom was cast and extracted using the moulds described in Section 2.2.1. The chosen materials included polyvinyl alcohol (PVA) for its acoustic resemblance to arterial tissue and adjustable mechanical properties [31], potassium sorbate as a preservative and antimicrobial agent in PVA-based tissue phantoms [32], and silicon carbide for its ultrasound scattering properties, mimicking the echogenicity of biological tissues [33]. The phantom formulation was prepared using 870 g (86.74 wt%) deionised (DI) water, 100 g (9.97 wt%) polyvinyl alcohol (PVA), 30 g (2.99 wt%) silicon carbide, and 3 g (0.30 wt%) potassium sorbate.

The DI water was first heated to 90°C to enhance solubility, particularly of PVA, which dissolves more effectively at elevated temperatures. Once the target temperature was reached, PVA, silicon carbide, and potassium sorbate were added separately and stirred continuously for one hour to ensure complete dissolution.

Then, the solution was allowed to cool and degas at room temperature for approximately 12 hours. This step helped minimise the risk of mould deformation during the casting process by eliminating air bubbles and reducing internal pressure changes [34].

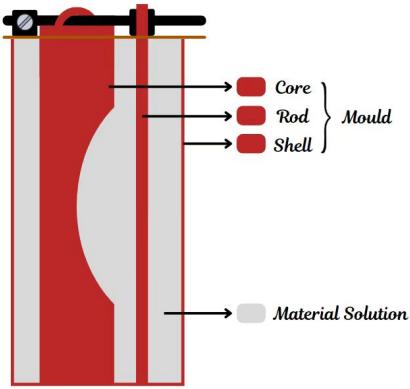


Figure 7: Plastic rod used as a placeholder for microvessel simulation.

Before casting, a plastic rod was suspended within the mould using an external frame, serving as a placeholder for a vessel structure in the final phantom. While ideally this would represent a microvessel, the current channel size was significantly larger due to fabrication constraints [31].

After the solution cooled and degassed, it was poured into the 3D-printed mould. The mould was then placed in a standard freezer for 60 hours to undergo a continuous freezing phase without thawing. This process is incorrect and will be further addressed in the Discussion part.

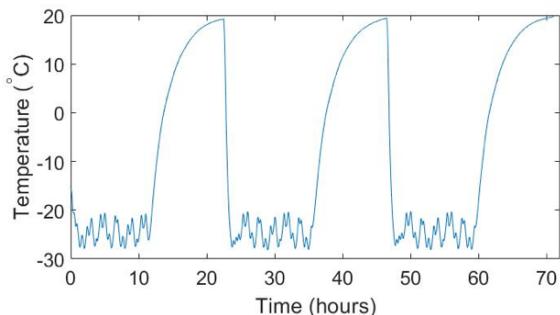


Figure 8: Temperature monitoring during freeze-thaw cycles. Figure created by Joe Hansen-Shearer from Professor Tang's group at Imperial College London.

Following this, the mould was moved to a specialised freeze-thaw system that underwent four 12-hour cycles of freezing at -20°C and then thawing at 20°C (see Figure 8) [35]. These cycles were critical for improving the mechanical strength of the PVA gel by promoting physical cross-linking and reinforcing the structure [36].

2.2.3. 3D Ultrasound Testing

Gloria Lee, a research postgraduate in Professor Tang's laboratory at Imperial College London, performed the 3D ultrasound acquisition and conducted the beamforming of the data used in this section.

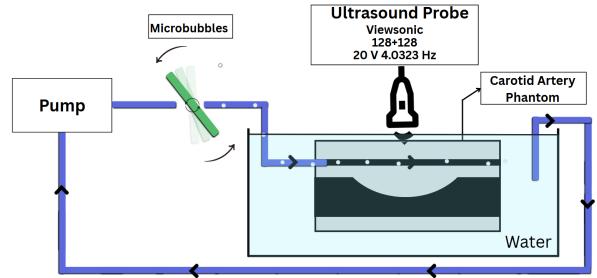


Figure 9: 3D ultrasonic testing procedure.

Ultrasound imaging of the fabricated phantom was performed using a row-column array probe integrated with a Versonic system. The probe featured 128 rows and 128 columns, enabling high-resolution three-dimensional imaging. The system operated with an input voltage of 20 V and a central frequency of 4.0323 MHz.

To simulate clinical conditions, the phantom was submerged in a water tank to serve as the acoustic coupling medium as shown in Figure 9. The ultrasound probe was positioned vertically above the phantom. Microbubbles were introduced into the main lumen using a magnetic stirrer-based setup. A microbubble contrast agent was used to enhance the images. As there was no attachment large enough to administer this directly into the vessel, the microbubbles were introduced at the opening of the main lumen and propagated through with a magnetic stirrer.

A peristaltic pump maintained a steady flow rate of 1 ml/s to replicate physiological conditions. Ultrasound images were captured at three distinct regions within the phantom to assess structure visibility and evaluate the effectiveness of microbubble contrast enhancement across the entire volume.

3. RESULTS

3.1. Computational Results

The simulated images were acquired using the method from section 2.1 and the parameters outlined in Table 1.

3.1.1. Static 2D Image Simulation

3.1.1a Varying the Number of Scatterers — Comparing Intensity Differences

Using the original clinical image (Figure 10) as a reference, images were simulated with a varying number of scatterers ranging from 10,000 to 100,000. For each scatterer count, the output from Field II is illustrated in Figure 11.

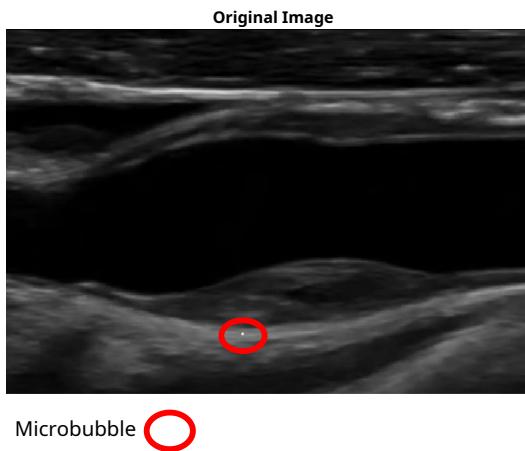


Figure 10: Original clinical image obtained from ultrasound.

As can be seen in Figure 11, the 10,000-scatterer case (Processed 1) shows some vessel and plaque detail with low intensity differences from the original, mainly around the plaque edges. With 20,000 scatterers (Processed 2), the speckle pattern becomes more pronounced, showing greater variability compared to the 10,000-case. The 50,000-scatterer image (Processed 3) displays clearer vessel and plaque details, while the corresponding difference image shows only minor discrepancies, again primarily around the plaque. Finally, the 100,000-scatterer image (Processed 4) closely matches the original vessel structure, with the dif-

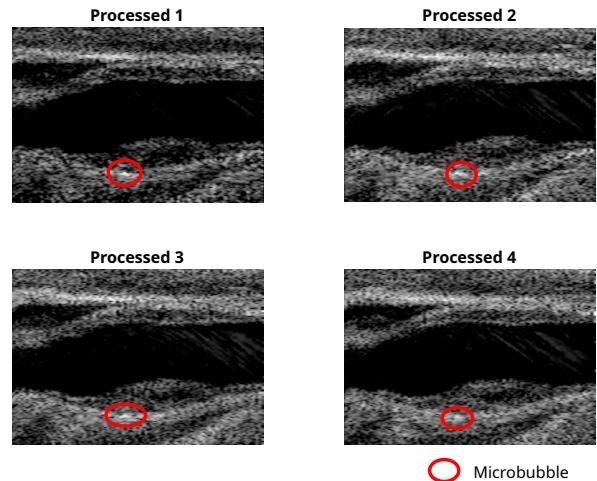


Figure 11: Simulated/processed images with fixed number of lines (100) and varying scatterer counts: Processed 1 (10,000), Processed 2 (20,000), Processed 3 (50,000), and Processed 4 (100,000).

ference image revealing even fewer discrepancies, particularly around the plaque and speckle areas.

A MATLAB code `processed_bubble`(see Appendix A.8) was written to calculate the mean squared error (MSE) [37] between the processed images and the original clinical image. Table 3 details the MSE values for each scatterer count.

Number of Scatterers	MSE
10,000	1195.7186
20,000	2127.4978
50,000	1833.7047
100,000	1769.1526

Table 3: Mean Squared Error (MSE) for varying number of scatterers.

3.1.1b Varying the Number of Lines — Comparing Intensity Differences

To assess the effect of scan line density, further simulations were performed with a fixed scatterer count of 20,000 while varying the number of scan lines from 100 to 200.

In Figure 12, the image with 100 scan lines (Processed 1) exhibits pronounced graininess and lower lateral resolution, particularly along the arterial wall and plaque boundaries. With 150 scan lines

(Processed 2), the graininess is reduced and finer details in the plaque region emerge more clearly. At 200 scan lines (Processed 3), the image is the smoothest and clearest, with well-defined boundaries and minimal artefacts; the corresponding intensity difference image also shows smaller, more uniform discrepancies.

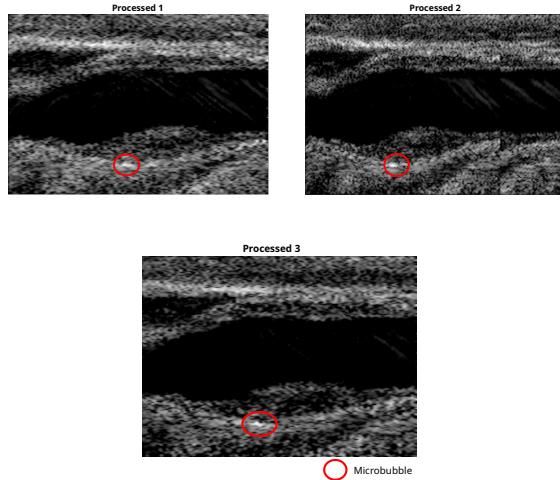


Figure 12: Simulated/processed images with a fixed scatterer count (20,000) and varying number of scan lines: Processed 1 (100 lines), Processed 2 (150 lines), and Processed 3 (200 lines).

Table 4 presents the MSE values associated with each scan line configuration.

Number of Lines	MSE
100	2118.6891
150	1719.5105
200	1188.4925

Table 4: Mean Squared Error (MSE) for varying number of scan lines.

Overall, the MSE values summarised in Tables 3 and 4 illustrate visual improvements in resolution and speckle as the number of scatterers and scan lines increases, while maintaining the phantom structure, including the plaque.

3.1.2. Dynamic 2D Simulation of a microbubble flowing through a microvessel

After generating the 2D images, we carried out a motion simulation of a microbubble flowing

through a microvessel within the plaque. We used a frame rate of 30 fps [38] to produce a 2-second video, resulting in a total of 60 frames processed using Field II.

The microbubble was modelled as moving vertically through the microvessel across all frames.

Based on previous parameter analyses and to reduce computational load, the simulation was performed with 100 scan lines and 200,000 scatterers, even though 200 scan lines demonstrated better performance.

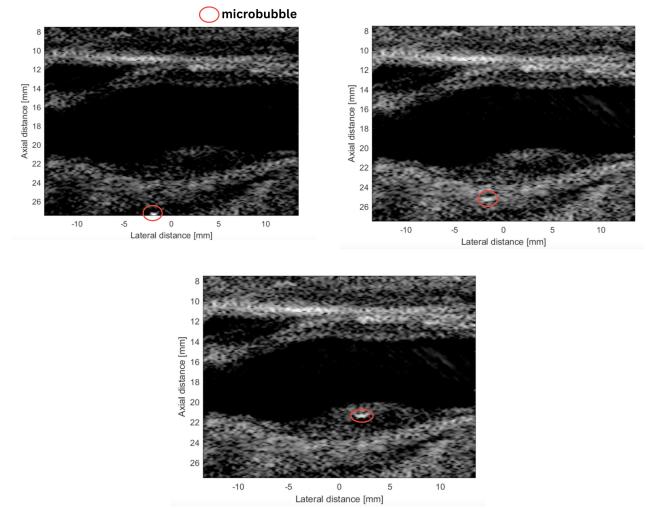


Figure 13: 2D moving simulation with a microbubble flowing through a microvessel within the plaque. Frames are shown at 0 s, 1 s, and 2 s (The bubble moves from bottom to top) and taken directly from the video simulation.

3.1.3. The 3D Model of the Carotid Artery Simulation

The ultrasound image alignment and 3D volume reconstruction successfully created a clear and coherent anatomical representation.

The final 3D volume showed good spatial continuity, with minimal displacement between slices and clearly defined target structures. This method effectively handled differences in image position and size, successfully combining the 2D ultrasound slices into a single 3D view.

The final cropped volume was visualised interactively using MATLAB's `volumeViewer`.

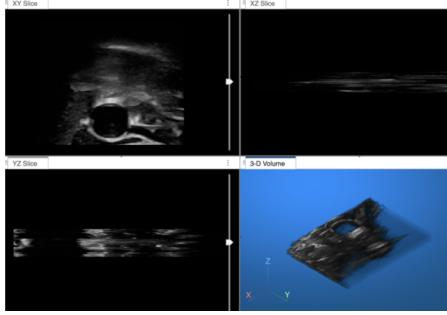


Figure 14: VolumeViewer for the ultrasound image alignment, illustrating the final 3D reconstructed volume.

3.1.4. Qualitative Assessment

A Vascular Surgery Specialist Registrar and doctoral fellow at Imperial College London, Dr David Hakim, was consulted to qualitatively assess our static 2D image simulation. The feedback confirmed that the model is sufficiently realistic for clinical interpretation. Dr Hakim noted that vascular surgeons would appreciate access to such models, as they provide enough detail to enable accurate measurements and support clinical decision-making.

3.2. Experimental Results

The physical phantom mould was successfully produced using 3D printing, and its three components were assembled seamlessly. The casting and demoulding of the PVA-based solution from the mould were completed without any deformation. Specifically, the notch feature within the inner shell, representing the atherosclerotic plaque, was accurately retained in the final cast phantom.

The phantom was then tested with ultrasound. The following figures present the ultrasound imaging results.

The three scans in Figures 15, 16, and 17 successfully showed the main vessel, secondary vessel, and plaque, each distinguishable based on their spatial location and acoustic response. Microbubble contrast agents were clearly visualised flowing through both vessels, confirming the effectiveness of the stirring-based perfusion. The different aspects of the scans are compared in Table 5.

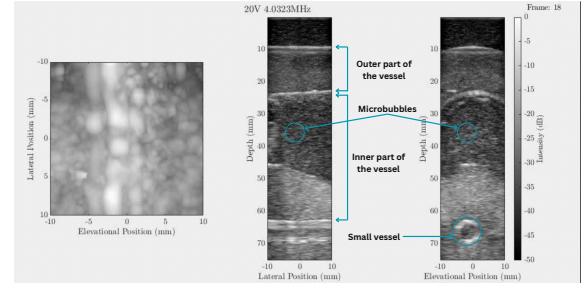


Figure 15: Annotated ultrasound imaging of the left side of the phantom (frame 18 from the first 50-frame segment). B-mode image of the elevation position vs. lateral position, B-mode image along the lateral axis (depth vs. lateral position), and B-mode image along the elevational axis (depth vs. elevational position). The image is directly acquired from moving video.

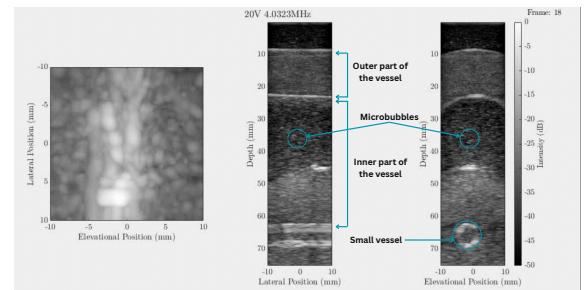


Figure 16: Annotated ultrasound imaging of the middle of the phantom (frame 18 from the first 50-frame segment). The image displays the same modalities as the left scan. The image is directly acquired from moving video.

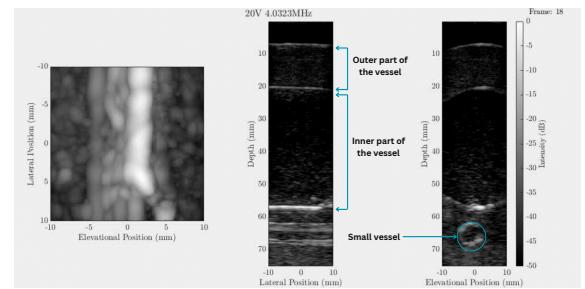


Figure 17: Annotated ultrasound imaging of the right side of the phantom (frame 18 from the first 50-frame segment), showing similar imaging modalities. The image is directly acquired from moving video.

Aspect	Left	Middle	Right
Overall Image Quality	Diffuse, low contrast, plaque near top edge of field of view	Better than left but slightly lower contrast than right	Highest contrast overall, better visibility
Microbubble Visibility in Large Vessel	Visible near plaque across a ~ 30–40 mm depth range	Clearly visible across a ~ 30–40 mm depth range	Not visible, possibly outside imaging zone
Large Vessel Structure	Seen but with blurred boundaries	Not prominently visible (only partial vessel seen)	Visible with sharper boundaries but lower contrast
Small Vessel Inclusion	Difficult to distinguish smaller branches (~ 70–80% of large vessel diameter)	Strong definition of smaller vessels	Slightly reduced clarity but moderate signal-to-noise ratio

Table 5: Summary of the Results Comparing Left, Middle, and Right Scans

4. DISCUSSION

4.1. Discussion of this study's computational research and findings

The aim of this study was to accurately simulate 2D ultrasound images of microbubble flow through plaque microvasculature in the carotid artery, develop a method to extend these to 3D, and provide a ground truth model to refine ULM algorithms for improved imaging of plaque microvessels.

4.1.1. Quantifying the relationship between image quality and the number of scatterers

Our results show that increasing the number of scatterers produces a more detailed, grainy speckle pattern from echo interference, making vessel and plaque areas appear more realistic.

However, this detailed speckle can partially mask the microbubble's high-amplitude signal, reducing its relative contrast and making it less obvious. This effect can enhance realism, such as in vivo, where microbubble contrast is influenced by inherent tissue scattering. The slight contrast reduction in simulations with more scatterers may better mimic clinical conditions where the microbubble signal competes with complex speckle patterns.

MSE values do not consistently decrease with more scatterers. The 10,000-scatterer case yields a lower MSE, while 20,000, 50,000, and 100,000 show higher MSEs, reflecting finer speckle detail. A slightly higher MSE is acceptable if the image better mimics clinical scans. However, more scatterers also increase MATLAB processing time: for example, with $N = 20,000$, one line takes about 4 seconds, but doubles for $N = 50,000$. Therefore, there is a trade-off between image quality and simulation time. For video simulations, such as vessel dilation or contraction, 10,000 scatterers are sufficient to produce realistic speckle patterns without excessive computational load.

4.1.2. Quantifying the relationship between image quality and the number of lines

With fewer scan lines ($N = 100$), lateral resolution is lower, making the image grainier and the edges of the artery wall and plaque less distinct, resulting in a higher MSE compared to the reference image. Increasing the number of scan lines, such as $N = 150$ or 200 , improves lateral resolution, reduces graininess, and lowers MSE, making the plaque region more defined. While more scan lines enhance image quality, they also increase computational complexity, as more subdivisions require heavier and longer processing times. In practice, balancing image quality and efficiency is essential. For tracking microbubbles in tiny plaque vessels, using more scan lines better reveals plaque shape, as these vessels often lie along the lateral sides of the main vessel. Detailed imaging of this region is crucial for accurate diagnosis.

4.1.3. The usage and limitation of 2D moving simulation

With a predefined path, the simulation offers a reliable benchmark for assessing microbubble detection, localisation, and velocity estimation, helping to identify tracking errors without real-world complexities. Its deterministic design ensures reproducibility, making it useful for comparing tracking methods and refining algorithms. However, using static phantom images and simplified microbubble motion inadequately captures the complex tissue dynamics and flow variations seen *in vivo*, where bubbles may change speed, rotate, or fragment due to environmental interactions.

4.1.4. Limitations of 3D simulation

Ultrasound images and their binary masks from an online dataset were stacked to create a 3D volumetric model of the carotid artery. Since the images were captured in real time, tissue motion and pulsatility caused misalignment, which was corrected using a common centroid and central cropping. However, discontinuity in the slices stems from how the dataset was acquired. Instead of a full sequence spanning from shallow to deep anatomical layers, the dataset contains multiple images taken at a single depth. For any given depth, there isn't a continuous progression covering the full volume. Despite this, the method shows promise for 3D reconstruction from 2D images and, with a better dataset, could produce a more accurate and refined model.

4.1.5. Qualitative assessment

Based on qualitative evaluation by Dr David Hakim, the images are of good quality for assessing structural details but are less effective for analysing flow dynamics. This study focused on B-mode imaging to highlight structural features of the carotid artery and surrounding vasculature. For deeper characterisation of blood flow, Doppler imaging would be more appropriate.

4.2. Discussion of this study's experimental research and findings

Our aim was to create a robust, accurate, and reusable carotid artery phantom to model and characterise atherosclerotic disease for ultrasound imaging.

4.2.1. Initial Phantom Results

Overall, the phantom was able to mimic the tissue's echogenic properties, largely due to the choice of PVA, resulting in a simplified ultrasound image-set of the (artificial) arterial plaque and microvessel. The principles of creating vascular channels in phantoms, proposed by Adusei et al., were demonstrated. Adjusting the number of freeze-thaw cycles facilitated control over the phantom's strength and stiffness, while incorporating scattering particles into the PVA ameliorated its echogenicity under B-mode ultrasound.

The phantom's ultrasound images varied with probe positioning. At the central position, the clearest, highest contrast images were achieved. Both vessels were distinctly visible, with the smaller vessel (microvessel in practice) appearing as a well-defined ring at 60–70 mm depth, and microbubbles clearly observed in the lumen (30 mm depth). On the other hand, positioning the probe at the left or right led to inferior image quality than at the center. The variation in these images are likely due to fluctuations in the concentration of the microbubble contrast flowing into the phantom during testing. Use of the stirrer limited control of the flow of the microbubbles into the vessels which may account for some of the variability in the resultant images.

The results of this experimental study imply that a framework for a ground-truth set of ultrasound images obtained from phantom models, with the end goal of improving ULM algorithms, can be established based on the method and materials outlined, as well as through ensuring that a controlled and adequate concentration of microbubble contrast is administered for optimal visual clarity.

4.2.2. Limitations of the phantom fabrication process

The 3D-printed mould had limitations. The inner shell and rod were unsecured, exposing them to shifting during PVA pouring and introducing variability between future phantom reproductions and ultrasound comparisons. For future improvements, the rod and shell will be secured within the mould and its walls reinforced using alternative materials, preventing cracking from PVA expansion.

The significant scaled-up dimensions of our phantom compared to an *in vivo* neovascularised plaque-laden carotid artery introduced difficulties in identifying and tracking microbubbles. This limits the viability of our images in being used as a ground truth for ULM. Therefore, future phantom iterations should be refined to match the physiological *in vivo* dimensions of carotid arteries, and to include sub-millimeter micro-vessels, in order to accurately validate computationally simulated images [39, 40]. In addition, further experimental testing can be done to determine the optimal concentration of microbubble contrast that permits effective visualisation of the vessels.

Future phantom iterations should mimic diseased tissue's varied structure, unlike the current phantom's uniform density[41]. Reiterations should also include a carotid bifurcation - a common site for atherosclerosis due to turbulent flow [42]. Material testing would be needed to help better understand PVA's properties, and compare it to human tissue and existing models.

Although the main vessel was visible on ultrasound, its large size exceeded the image frame and limited microbubble control. Despite this, we managed to propagate some bubbles into the lumen. However, testing with blood-mimicking fluid is still pending and would provide a more physiologically accurate model.

5. CONCLUSION

For the first part of our bilateral strategy, we successfully simulated a static 2D carotid artery

with a plaque, which was used to simulate an accurate 2D video of a microbubble flowing through a microvessel within the plaque. These models, created computationally, have easily adjustable parameters, which can be chosen according to the trade-off between computational simulation time and accuracy. They serve as a ground truth for the fine-tuning of the ultrasound system and hence the development of ULM.

For the second part of our strategy, we created an anatomically and acoustically accurate, reusable phantom for ultrasound imaging of atherosclerosis. This will support reliable plaque characterisation in a controlled environment, helping to develop a risk stratification tool to assess plaque vulnerability and improve patient management.

6. Acknowledgments

We would like to thank: Professor MengXing Tang, Gloria Lee, HaoXuan Yao – this project could not be realised without you.

References

- [1] World Health Organization. Cardiovascular diseases, 2025. Available from: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1.
- [2] University of Ottawa Heart Institute. Coronary artery disease (atherosclerosis) - university of ottawa heart institute, 2018. Available from: <https://www.ottawaheart.ca/heart-condition/coronary-artery-disease-atherosclerosis>.
- [3] David.J Spence. Ultrasound measurement of carotid plaque as a surrogate outcome for coronary artery disease. *The American Journal of Cardiology*, 89, 2002 Feb. Available from: [https://www.ajconline.org/article/S0002-9149\(01\)02327-X/fulltext](https://www.ajconline.org/article/S0002-9149(01)02327-X/fulltext).
- [4] JL Miller. Ultrasound resolution beats the diffraction limit. *Physics Today*, 69(2):14–16, 2016 Feb. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8515632/>.
- [5] GR Lockwood, LK Ryan, JW Hunt, and FS Foster. Measurement of the ultrasonic properties of vascular tissues and blood from 35–65 mhz. *Ultrasound in Medicine & Biology*, 17(7):653–666, 1991 Jan.
- [6] M Gerhard-Herman, JM Gardin, MR Jaff, ER Mohler, MJ Roman, and TZ Naqvi. Guidelines for noninvasive vascular laboratory testing. *Journal of The American Society of Echocardiography*, 19(8):955–972, 2006 Nov.
- [7] K Christensen-Jeffries, O Couture, PA Dayton, YC Eldar, K Hynynen, F Kiessling, et al. Super-resolution ultrasound imaging. *Ultrasound in Medicine & Biology*, 46(4):865–891, 2020 Apr.
- [8] P Song, JM Rubin, and MR Lowerison. Super-resolution ultrasound microvascular imaging: Is it ready for clinical use? *Zeitschrift für Medizinische Physik*, 33(3), 2023 May. Available from: <https://www.sciencedirect.com/science/article/pii/S0939388923000430>.
- [9] J Kim, MR Lowerison, C Sekaran, Z Kou, Z Dong, ML Oelze, et al. Improved ultrasound localization microscopy based on microbubble uncoupling via transmit excitation. *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 69(3):1041–1052, 2022 Mar.
- [10] H Lee, H Kim, H Han, M Lee, S Lee, H Yoo, et al. Microbubbles used for contrast enhanced ultrasound and theragnosis: a review of principles to applications. *Biomedical Engineering Letters*, 7(2):59–69, 2017 Feb. Available from: <https://link.springer.com/article/10.1007%2Fs13534-017-0016-5>.
- [11] E Stride and N Sa. Microbubble ultrasound contrast agents: a review, 2003. Available from: <https://www.ucl.ac.uk/~zcapc43/Microbubble%20ultrasound%20contrast%20agents%20review.pdf>.
- [12] C. Errico, J. Pierre, S. Pezet, et al. Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging. *Nature*, 527:499–502, 2015 Nov.
- [13] H Yi, MR Lowerison, P Song, and W Zhang. A review of clinical applications for super-resolution ultrasound localization microscopy. *Current Medical Science*, 42(1):1–16, 2022 Feb.
- [14] T Mano, T Grutman, and T Illovitsh. Versatile ultrasound-compatible microfluidic platform for in vitro microvasculature flow research and imaging optimization. *ACS Omega*, 8(50):47667–47777, 2023 Dec. Available from: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10734021/>.
- [15] L Denis, G Chabouh, B Heiles, and O Couture. Volumetric ultrasound localization microscopy. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 71(12):1643–1656, 2024 Dec. Available from: <https://ieeexplore.ieee.org/document/10735230>.

- [16] J Arendt Jensen. Field ii ultrasound simulation program. Available from: <https://field-ii.dk//?background.html>.
- [17] JA Jensen and P Munk. Computer phantoms for simulating ultrasound B-mode and CFM images. In *23rd International Symposium on Acoustical Imaging*, Boston, Massachusetts, USA, 1997 Apr.
- [18] JA Jensen. Simulation of advanced ultrasound systems using field ii. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano*, 2005. [IEEE Cat No 04EX821]. Available from: <https://field-ii.dk//?background.html>.
- [19] M Vega. Ge transducers for vantage systems - verasonics, 2023. Available from: <https://verasonics.com/ge-transducers-for-vantage-systems/>.
- [20] Bird Ultrasound. Carotid plaque burden case of the week, 2022 Apr. Available from: <https://www.youtube.com/watch?v=BvHKfHvLfk4>.
- [21] PR Stepanishen. Pulsed transmit/receive response of ultrasonic piezoelectric transducers. *Journal of the Acoustical Society of America*, 69:1815–1827, 1981 June.
- [22] JA Jensen and NB Svendsen. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 39:262–267, 1992 March.
- [23] JA Jensen. Speed-accuracy trade-offs in computing spatial impulse responses for simulating medical ultrasound imaging. *Journal of Computational Acoustics*, 9(3):731–744, 2001.
- [24] JA Jensen. A new calculation procedure for spatial impulse responses in ultrasound. *Journal of the Acoustical Society of America*, 105:3266–3274, 1999.
- [25] JA Jensen. A model for the propagation and scattering of ultrasound in tissue. *Journal of the Acoustical Society of America*, 89:182–191, 1991 Jan.
- [26] JA Jensen, M Schlaikjer, and S Torp Pedersen. Simulation of RF data with tissue motion for optimizing stationary echo canceling filters. *Ultrasonics*, 41:415–419, 2003 Aug.
- [27] José Seabra and João M. Sanches. Modeling log-compressed ultrasound images for radio frequency signal recovery. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, Canada, 2008 Aug.
- [28] Yeonhwa Lee, Jinbum Kang, and Yangmo Yoo. Automatic dynamic range adjustment for ultrasound b-mode imaging. *Ultrasonics*, 56:435–443, 2015 Feb.
- [29] A Momot. Common carotid artery ultrasound images, 2022 Nov. Available from: <https://data.mendeley.com/datasets/d4xt63mgjm/1>.
- [30] Ole Vegard Solberg, Frank Lindseth, Hans Torp, Robert E Blake, and Toril A Nagelhus Hernes. Freehand 3D ultrasound reconstruction algorithms—a review. *Ultrasound in Medicine & Biology*, 33:991–1009, 2007 Jul.
- [31] Y Guendouz, NAM Razif, F Bernasconi, G O'Brien, RD Johnston, and C Lally. Simulating atherosclerotic plaque mechanics using polyvinyl alcohol (pva) cryogel artery phantoms, ultrasound imaging and inverse finite element analysis. *Physics in Medicine & Biology*, 69(24):245020, 2024 Dec.
- [32] A Jawli, W Aldehani, G Nabi, and Z Huang. Tissue-mimicking material fabrication and properties for multiparametric ultrasound phantoms: A systematic review. *Bioengineering*, 11(6):620, 2024.
- [33] R Schatz. Development of a vascularized phantom for carotid ultrasound examination, 2022.
- [34] M. Divandari and J. Campbell. The mechanism of bubble damage in castings. In *1st International Conference on Gating, Filling and Feeding of Alu-*

- minum Castings*, pages 49–63, Opryland Hotel, Nashville, TN, 1999 Oct. Available from: https://www.researchgate.net/publication/233987664_The_Mechanism_of_Bubble_Damage_in_Castings.
- [35] H Adelnia, R Ensandoost, SS Moonshi, JN Gavgani, EI Vasafi, and HT Ta. Freeze/thawed polyvinyl alcohol hydrogels: Present, past and future. *European Polymer Journal*, 164:110974, 2022. Available from: <https://www.sciencedirect.com/science/article/pii/S0014305721007084#b1030>.
- [36] A. J. Malone, S. Cournane, I. G. Naydenova, A. J. Fagan, and J. E. Browne. Polyvinyl alcohol cryogel based vessel mimicking material for modelling the progression of atherosclerosis. *Physica Medica*, 69:1–8, 2020. Available from: <https://doi.org/10.1016/j.ejmp.2019.11.012>.
- [37] K Yang, P Hoskins, G Corner, C Xia, and Z Huang. Wall shear stress measurement in carotid artery phantoms with variation in degree of stenosis using plane wave vector doppler. *Applied Sciences*, 13(1):617–627, 2023 Jan. Available from: <https://www.mdpi.com/2076-3417/13/1/617>.
- [38] J Stoitsis, S Golemati, V Koropouli, and KS Nikita. Simulating dynamic b-mode ultrasound image data of the common carotid artery, 2008 Sept. Available from: <https://ieeexplore.ieee.org/document/4659958>.
- [39] AC Dudley and AW Griffioen. Pathological angiogenesis: mechanisms and therapeutic strategies. *Angiogenesis*, 26, 2023 Apr.
- [40] K Dakok, M Matjafri, N Suardi, A Oglat, and S Nabasu. A review of carotid artery phantoms for doppler ultrasound applications. *PubMed*, 29(3):157–166, 2021 Nov. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8515632/>.
- [41] S Zhang, X Yu, H Gu, B Kang, N Guo, and X Wang. Identification of high-risk carotid plaque by using carotid perivascular fat density on computed tomography angiography. *European Journal of Radiology*, 150:110269, 2022 May.
- [42] P Jiang, Z Chen, DS Hippe, H Watase, B Sun, R Lin, et al. Association between carotid bifurcation geometry and atherosclerotic plaque vulnerability. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 40(5):1383–1391, 2020 May.

A. Appendix

A.1. Project Management Assessment

According to our pitched timeline, the remaining computational task was to develop a 3D simulation of a microbubble flowing within a carotid artery, as by then we had successfully completed the 2D dynamic simulation. We began the 3D simulation around mid-March, which involved stacking static 2D ultrasound images according to a depth gradient. However, we found that simulating every image was too computationally intensive, especially without access to high-performance computers. Furthermore, the dataset had no depth variation and, as it consisted of ultrasound images, the slices were not continuous. Given these barriers, we focused on demonstrating the stacking method as a concept rather than producing a full reconstruction of a flowing microbubble through microvasculature.

Regarding the experimental objectives of the timeline, after designing and producing the mould and phantom, we planned to reiterate the design following ultrasound testing to make it more realistic. However, the first phantom was completed in early April, leaving us insufficient time to reiterate the design with more realistic scaling dimensions or to test its mechanical properties in detail. Nonetheless, we still carried out ultrasound testing and assessed the initial phantom. Our objectives were achieved following the original timeline with results adjusted to overcome the challenges.

A.2. Key Project Management Lessons

A.2.1. Efficient Communication

Clear and consistent communication was crucial to the project's success. For better communication efficiency, we split into computational and experimental sub-teams. Our group held regular meetings and had weekly check-ins with Professor Tang's laboratory group to ensure that everyone stayed updated on progress, while also gaining feedback to improve our models. Each sub-team held additional meetings to share updates on their individual tasks, depending on the progress of the project. One of the group members recorded detailed notes on OneNote during each meeting and outlined upcoming tasks for each member in the WhatsApp group, with specified key deadlines. We also used several communication channels, such as Microsoft Teams, WhatsApp, and WeChat, to maintain productivity and stay on schedule during the holidays.

A.2.2. Positive Teamwork Environment, Collaboration and Adaptability

Good team spirit and collaborative teamwork were also important in achieving our goals. Our willingness to adapt, collaborate, and remain flexible played a major role in driving the project's success. We began by clearly defining roles and responsibilities, ensuring that each team member's strengths were matched to appropriate tasks. This approach optimised our work efficiency, helped prioritise assignments, and enabled us to meet deadlines effectively. A positive working environment was also created, which significantly increased overall productivity and fostered mutual enthusiasm and commitment. As team members were taking different modules, some were very busy at times. During these periods, other members willingly took on additional tasks to keep the project moving smoothly. Reflecting on our process, we recognise the importance of adaptability, as refining our goals and adjusting our methods as new challenges emerged – such as the infeasibility of simulating a flowing microbubble in 3D – was essential to our success.

A.2.3. Defined Milestones and Continuous Performance Tracking

Another key lesson was the importance of setting clear milestones to ensure progress. By breaking the project into manageable objectives, we created a clear roadmap. Completing each step before moving on to the next kept everyone focused and aligned. Consistent progress monitoring and feedback from Professor Tang's laboratory group helped us identify challenges early, allowing us to adjust our strategy and reallocate efforts as needed. This proactive tracking enabled us to assess our progress objectively and created a sense of responsibility within the team. In hindsight, this was vital for maintaining momentum and ensuring that each phase of the project contributed directly to our main goals. Our rigorous approach to milestone-setting and performance evaluation helped us remain adaptable and deliver high-quality work throughout the project.

A.3. Execution.m MATLAB code

```
1 %% 1) Set Global Parameters
2 disp('Setting global parameters...');

3
4 % Field II Path (Update with your own)
5 addpath('F:\matlab\r2023b\field_2');

6
7 % Initialize Field II
8 field_init;

9
10 % Model: L3-12-D Linear Array
11 % Transducer Parameters
12 f0 = 6.5e6; % Transducer center frequency [Hz]
13 fs = 100e6; % Sampling frequency [Hz]
14 c = 1540; % Speed of sound [m/s]
15 lambda = c / f0; % Wavelength [m]

16
17 pitch = 0.20 / 1000; % Element pitch [m]
18 kerf = 0.05 / 1000; % Gap between elements [m]
19 width = pitch - kerf; % Width of transducer element [m]
20 element_height = 5 / 1000; % Height of transducer element [m]
21 focus = [0, 0, 0.022]; % Fixed focal point [m]
22 N_elements = 256; % Total number of physical elements
23 N_active = 128; % Number of active elements

24
25 % Phantom Dimensions
26 x_size = 27/1000; % Width [m]
27 z_size = 20/1000; % Depth [m]
28 y_size = 0.01/1000; % Thickness [m]
29 z_start = 2/1000; % Start of the phantom in z-direction [m]

30
31 % Image Parameters
32 no_lines = 100; % Number of lines in the image
33 image_width = x_size; % Width of the image [m]
34 d_x = image_width / no_lines; % Distance between adjacent lines [m]

35
36 N = 20000; % Number of scatterers

37
38 disp('All parameters are set!');

39
40 %% 2) Generate Scatterer Positions and Noise
41 disp('Generating scatterer positions and random noise...');

42
43 positions = zeros(N, 3);
44 positions(:,1) = (rand(N,1) - 0.5) * x_size; % x
45 positions(:,2) = (rand(N,1) - 0.5) * y_size; % y
```

```

46 positions(:,3) = rand(N,1) * z_size + z_start; % z
47
48 random_noise = randn(N,1);
49
50 save('scatterer_positions.mat', 'positions', 'random_noise');
51 disp('Scatterer positions and random noise generated and saved!');
52
53 %% 3) Process Each BMP File Sequentially
54 image_folder = 'BubbleImage';
55 output_folder = 'ProcessedImages';
56 if ~exist(output_folder, 'dir')
57     mkdir(output_folder);
58 end
59
60 log_file = 'processed_files.txt';
61
62 if isfile(log_file)
63     fid = fopen(log_file, 'r');
64     processed_files = textscan(fid, '%s', 'Delimiter', '\n');
65     fclose(fid);
66     processed_files = processed_files{1};
67 else
68     processed_files = {};
69 end
70
71 bmp_files = dir(fullfile(image_folder, '*.bmp'));
72 bmp_filenames = {bmp_files.name};
73
74 new_files = setdiff(bmp_filenames, processed_files);
75
76 for i = 1:length(new_files)
77     bmp_filename = new_files{i};
78     disp(['Processing: ', bmp_filename]);
79
80     bmp_path = fullfile(image_folder, bmp_filename);
81     [liv_kid, ~] = bmpread(bmp_path);
82
83     folderPath = 'rf_data';
84     delete(fullfile(folderPath, '*.*'));
85     disp('Folder cleaned: All files removed.');
86
87     filename = bmp_filename;
88
89     load('scatterer_positions.mat', 'positions');
90
91     [phantom_positions, phantom_amplitudes] = scatterers_phantom( ...
92         N, x_size, z_size, y_size, z_start, liv_kid, filename, positions,
93         random_noise);
94
95     save('pht_data.mat', 'positions', 'phantom_amplitudes');
96     disp('Phantom Data saved!');
97
98     disp('Running B-mode Imaging... ');
99     run('train_image.m');
100
101    figure();
102    disp('Processing RF Data and Displaying Image... ');
103    run('make_image.m');
104
105    output_filename = strrep(bmp_filename, '.bmp', '_processed.png');
106    save_path = fullfile(output_folder, output_filename);
107    saveas(gcf, save_path);
108    disp(['Image saved: ', save_path]);

```

```

108
109 close(gcf);
110
111 fid = fopen(log_file, 'a');
112 fprintf(fid, '%s\n', bmp_filename);
113 fclose(fid);
114 end
115
116 disp('All new BMP files processed and saved!');

```

A.4. Scatterer Positions and Amplitudes MATLAB code

```

1 function [positions, amp] = scatterers_phantom(N, x_size, z_size, y_size, z_start,
2     liv_kid, filename, cpositions, random_noise)
3
4 % Extract x_target_index and z_target_index from the filename
5 tokens = regexp(filename, '\((\d+),(\d+)\)', 'tokens');
6
7 if ~isempty(tokens)
8     x_target_index = str2double(tokens{1}{1}); % Extract x index
9     z_target_index = str2double(tokens{1}{2}); % Extract z index
10 else
11     error('Filename does not contain valid (x,z) pixel coordinates.');
12 end
13
14 % Define image coordinates
15 liv_kid = liv_kid';
16 [Nl, Ml] = size(liv_kid);
17
18 dx = x_size / Nl; % Sampling interval in x direction [m]
19 dz = z_size / Ml; % Sampling interval in z direction [m]
20
21 % Use precomputed positions
22 x = cpositions(:,1);
23 y = cpositions(:,2);
24 z = cpositions(:,3);
25
26 % Find the index for the amplitude value
27 xindex = round((x + 0.5 * x_size) / dx + 1); % x pixel index
28 zindex = round((z - z_start) / dz + 1); % z pixel index
29 inside = (xindex > 0) & (xindex <= Nl) & (zindex > 0) & (zindex <= Ml); % Valid
30     range check
31 index = (xindex + (zindex - 1) * Nl) .* inside + 1 * (1 - inside); % Pixel
32     position in matrix
33
34 % Assign amplitudes based on the image
35 amp = exp(liv_kid(index)/100);
36 amp = amp - min(amp);
37 amp = 1e6 * amp / max(amp);
38
39 % Use precomputed noise instead of generating new noise
40 amp = amp .* random_noise .* inside; % Apply randomness and ensure scatterers
41     inside the image
42
43 % Convert pixel index to real-world coordinates for the target scatterer
44 x_target = (x_target_index - 1) * dx - 0.5 * x_size;
45 z_target = (z_target_index - 1) * dz + z_start;
46 y_target = (rand - 0.5) * y_size; % Random y-coordinate within range
47
48 % Add one high-amplitude scatterer
49 x = [x; x_target];

```

```

46 y = [y; y_target];
47 z = [z; z_target];
48 amp = [amp; 3e6]; % Assign high amplitude
49
50 % Adjust z positions to maintain relative structure
51 z = z - min(z) + z_start;
52
53 % Store all positions
54 positions = [x y z];
55
56 end

```

A.5. Ultrasound Processing MATLAB code

```

1 % Linear Array B-mode Imaging with Adjusted Parameters
2 apodization_values = zeros(100, 2); % Columns: [N_pre, N_post]
3
4 % Set Sampling Frequency
5 set_sampling(fs);
6
7 % Create Transmit and Receive Apertures
8 xmit_aperture = xdc_linear_array(N_elements, width, element_height, kerf, 1, 1, focus
    );
9 receive_aperture = xdc_linear_array(N_elements, width, element_height, kerf, 1, 1,
    focus);
10
11 % Impulse Response and Excitation Signal
12 impulse_response = sin(2 * pi * f0 * (0:1/fs:2/f0));
13 impulse_response = impulse_response .* hanning(length(impulse_response))';
14 xdc_impulse(xmit_aperture, impulse_response);
15
16 excitation = sin(2 * pi * f0 * (0:1/fs:2/f0));
17 xdc_excitation(xmit_aperture, excitation);
18 xdc_impulse(receive_aperture, impulse_response);
19
20 % Load Scatterer Data
21 if ~exist('pht_data.mat', 'file')
22     error('Scatterer data file "pht_data.mat" not found.');
23 else
24     load pht_data; % Load 'phantom_positions' and 'phantom_amplitudes'
25 end
26
27 % Set Focal Zones for Reception
28 focal_zones = linspace(z_start, z_start + z_size, 5)';
29 Nf = length(focal_zones);
30 focus_times = (focal_zones - z_start) / c;
31
32 % Set Apodization
33 apo = hanning(N_active)';
34
35 % Linear Array Imaging
36 for i = 1:no_lines
37     % File for Storing RF Data
38     file_name = sprintf('rf_data/rf_ln%d.mat', i);
39
40     if ~exist(file_name, 'file')
41         % Reserve the Calculation by Creating an Empty File
42         save(file_name, 'i');
43
44         disp(['Now calculating line ', num2str(i)]);
45

```

```

46 % Calculate Imaging Direction
47 x = -image_width / 2 + (i - 1) * d_x;
48
49 % Set Focus for Current Direction
50 xdc_center_focus(xmit_aperture, [x, 0, 0]);
51 xdc_focus(xmit_aperture, 0, [x, 0, focal_zones(end)]);
52 xdc_center_focus(receive_aperture, [x, 0, 0]);
53 xdc_focus(receive_aperture, focus_times, [x * ones(Nf, 1), zeros(Nf, 1),
      focal_zones]);
54
55 % Calculate apodization vector for current line
56 N_post = 208;
57 N_pre = max(0, round(x / (width + kerf) + N_elements / 2 - N_active / 2));
58
59 if N_pre == 0
60     N_post = N_post - 3 * i; % Decrease by an additional 3 for each
       occurrence
61 else
62     N_post = max(0, N_elements - N_pre - N_active);
63 end
64
65 apodization_values(i, :) = [N_pre, N_post];
66
67 apo_vector = [zeros(1, N_pre), apo, zeros(1, N_post)];
68
69 % Ensure apodization vector matches number of elements
70 if length(apo_vector) > N_elements
71     apo_vector = apo_vector(1:N_elements); % Trim excess elements
72 elseif length(apo_vector) < N_elements
73     apo_vector = [apo_vector, zeros(1, N_elements - length(apo_vector))]; % Pad missing elements
74 end
75
76 % Set apodization for transmit and receive apertures
77 xdc_apodization(xmit_aperture, 0, apo_vector);
78 xdc_apodization(receive_aperture, 0, apo_vector);
79
80 % Simulate Scattering
81 [rf_data, tstart] = calc_scat(xmit_aperture, receive_aperture,
      phantom_positions, phantom_amplitudes);
82
83 % Save RF Data
84 save(file_name, 'rf_data', 'tstart');
85 else
86     disp(['Line ', num2str(i), ' is being processed by another machine.']);
87 end
88 end
89
90 % Free Apertures
91 xdc_free(xmit_aperture);
92 xdc_free(receive_aperture);
93
94 disp('Simulation completed. Run make_image to process and display the image.');

```

A.6. Image plotting MATLAB code

```

1 axis_x = [-x_size/2, x_size/2] * 1000; % Convert to mm
2 axis_z = [7.5/1000, 27.5/1000] * 1000; % Convert to mm
3
4 % Initialize variables
5 min_sample = 0;

```

```

6 % Read the RF data and adjust it in time
7 for i = 1:no_lines
8     % Load the RF data for each line
9     cmd = sprintf('load rf_data/rf_ln%d.mat', i);
10    disp(cmd);
11    eval(cmd);
12
13
14    % Find the envelope using the Hilbert transform
15    rf_env = abs(hilbert([zeros(round(tstart * fs - min_sample), 1); rf_data]));
16    env(1:max(size(rf_env)), i) = rf_env;
17 end
18
19 % Perform logarithmic compression
20 D = 10;           % Decimation factor
21 dB_range = 45;   % Dynamic range for display in dB
22
23 disp('Finding the envelope');
24 log_env = env(1:D:max(size(env)), :) / max(max(env));
25 log_env = 20 * log10(log_env);
26 log_env = 127 / dB_range * (log_env + dB_range);
27
28 % Perform interpolation to smooth the image
29 disp('Doing interpolation');
30 ID = 20; % Interpolation factor
31 [n, m] = size(log_env);
32 new_env = zeros(n, m * ID);
33
34 for i = 1:n
35     new_env(i, :) = interp(log_env(i, :), ID);
36 end
37
38 [n, m] = size(new_env);
39
40 % Display the image
41 fn = fs / D; % New sampling frequency after decimation
42 clf;
43 image(((1:(ID * no_lines - 1)) * d_x / ID - no_lines * d_x / 2) * 1000, ...
44     ((1:n) / fn + min_sample / fs) * c / 2 * 1000, new_env);
45 xlabel('Lateral distance [mm]');
46 ylabel('Axial distance [mm]');
47 colormap(gray(128));
48 axis('image');
49
50 % Adjust the axis range to match the kidney phantom
51 axis([axis_x(1), axis_x(2), axis_z(1), axis_z(2)]);

```

A.7. Ultrasound Image Alignment and 3D Volume Construction MATLAB code

```

1 % --- Settings: Adjust these paths as needed ---
2 binary_folder = 'C:\Users\EFL02\Downloads\Common Carotid Artery Ultrasound Images\
3     Expert mask images';
4 ultrasound_folder = 'C:\Users\EFL02\Downloads\Common Carotid Artery Ultrasound Images\
5     \US images';
6
7 % --- Step 1: Gather file names from both folders ---
8 valid_ext = {'png', '.jpg', '.jpeg', '.tif', '.tiff'};
9
10 binary_files_struct = dir(binary_folder);
11 binary_files = {};
12 for i = 1:length(binary_files_struct)

```

```

11 [~, ~, ext] = fileparts(binary_files_struct(i).name);
12 if ~binary_files_struct(i)..isdir && any(strcmpi(ext, valid_ext))
13     binary_files{end+1} = binary_files_struct(i).name;
14 end
15 end
16 binary_files = sort(binary_files);
17
18 ultrasound_files_struct = dir(ultrasound_folder);
19 ultrasound_files = {};
20 for i = 1:length(ultrasound_files_struct)
21     [~, ~, ext] = fileparts(ultrasound_files_struct(i).name);
22     if ~ultrasound_files_struct(i)..isdir && any(strcmpi(ext, valid_ext))
23         ultrasound_files{end+1} = ultrasound_files_struct(i).name;
24     end
25 end
26 ultrasound_files = sort(ultrasound_files);
27
28 if length(binary_files) ~= length(ultrasound_files)
29     error('The number of binary images and ultrasound images does not match.');
30 end
31
32 % --- Step 2: Compute centroids for each binary image ---
33 numImages = length(binary_files);
34 image_info = struct('fname', [], 'us_shape', [], 'centroid', []);
35
36 for i = 1:numImages
37     fname = binary_files{i};
38     binary_path = fullfile(binary_folder, fname);
39     binary_img = imread(binary_path);
40     if size(binary_img, 3) == 3
41         binary_img = rgb2gray(binary_img);
42     end
43     binary_thresh = imbinarize(binary_img, 127/255);
44
45     stats = regionprops(binary_thresh, 'Centroid', 'Area');
46     if ~isempty(stats)
47         [~, idx] = max([stats.Area]);
48         centroid = stats(idx).Centroid;
49     else
50         [h, w] = size(binary_thresh);
51         centroid = [w/2, h/2];
52     end
53
54     us_path = fullfile(ultrasound_folder, fname);
55     us_img = imread(us_path);
56     if size(us_img, 3) == 3
57         us_img = rgb2gray(us_img);
58     end
59     [h, w] = size(us_img);
60
61     image_info(i).fname = fname;
62     image_info(i).us_shape = [h, w];
63     image_info(i).centroid = centroid;
64 end
65
66 % --- Step 3: Determine reference centroid ---
67 all_centroids = zeros(numImages, 2);
68 for i = 1:numImages
69     all_centroids(i,:) = image_info(i).centroid;
70 end
71 ref_point = mean(all_centroids, 1);
72 fprintf('Reference centroid: (%.2f, %.2f)\n', ref_point(1), ref_point(2));
73

```

```

74 % --- Step 4: Compute shifts and canvas size ---
75 shifts = zeros(numImages, 2);
76 top_lefts = zeros(numImages, 2);
77 bottom_rights = zeros(numImages, 2);
78
79 for i = 1:numImages
80     cx = image_info(i).centroid(1);
81     cy = image_info(i).centroid(2);
82     dx = ref_point(1) - cx;
83     dy = ref_point(2) - cy;
84     shifts(i,:) = [dx, dy];
85
86     h = image_info(i).us_shape(1);
87     w = image_info(i).us_shape(2);
88
89     tl_x = dx;
90     tl_y = dy;
91     br_x = dx + w;
92     br_y = dy + h;
93
94     top_lefts(i,:) = [tl_x, tl_y];
95     bottom_rights(i,:) = [br_x, br_y];
96 end
97
98 min_x = min(top_lefts(:,1));
99 min_y = min(top_lefts(:,2));
100 max_x = max(bottom_rights(:,1));
101 max_y = max(bottom_rights(:,2));
102
103 canvas_width = ceil(max_x - min_x);
104 canvas_height = ceil(max_y - min_y);
105 fprintf('Canvas size: %d x %d\n', canvas_width, canvas_height);
106
107 offset_x = -min_x;
108 offset_y = -min_y;
109
110 % --- Step 5: Shift ultrasound images and place on canvas ---
111 aligned_images = cell(numImages, 1);
112
113 for i = 1:numImages
114     fname = image_info(i).fname;
115     us_path = fullfile.ultrasound_folder, fname);
116     us_img = imread(us_path);
117     if size(us_img, 3) == 3
118         us_img = rgb2gray(us_img);
119     end
120     [h, w] = size(us_img);
121
122     dx = shifts(i,1);
123     dy = shifts(i,2);
124     tx = round(dx + offset_x);
125     ty = round(dy + offset_y);
126
127     canvas = zeros(canvas_height, canvas_width, 'like', us_img);
128     row_range = (ty+1):(ty+h);
129     col_range = (tx+1):(tx+w);
130
131     if row_range(end) > canvas_height || col_range(end) > canvas_width || row_range
132         (1) < 1 || col_range(1) < 1
133         error('Image %s is out of canvas bounds.', fname);
134     end
135     canvas(row_range, col_range) = us_img;

```

```

136 aligned_images{i} = canvas;
137 end
138
139 % --- Step 6: Stack into 3D volume ---
140 volume = cat(3, aligned_images{:});
141 fprintf('3D volume size: %d x %d x %d\n', size(volume,1), size(volume,2), size(volume,3));
142
143 % --- Step 7: Crop the center area ---
144 crop_factor = 0.5;
145 [h, w, d] = size(volume);
146
147 crop_h = round(h * crop_factor);
148 crop_w = round(w * crop_factor);
149 start_row = round((h - crop_h) / 2) + 1;
150 start_col = round((w - crop_w) / 2) + 1;
151 end_row = start_row + crop_h - 1;
152 end_col = start_col + crop_w - 1;
153
154 cropped_volume = volume(start_row:end_row, start_col:end_col, :);
155 fprintf('Cropped 3D volume size: %d x %d x %d\n', ...
156     size(cropped_volume,1), size(cropped_volume,2), size(cropped_volume,3));
157
158 % --- Save volumes ---
159 save('aligned_volume.mat', 'volume');
160 fprintf('Aligned volume saved as "aligned_volume.mat"\n');
161
162 save('cropped_aligned_volume.mat', 'cropped_volume');
163 fprintf('Cropped aligned volume saved as "cropped_aligned_volume.mat"\n');
164
165 % --- Display cropped volume ---
166 figure;
167 volumeViewer(cropped_volume, 'ScaleFactors', [1, 1, 1]);

```

A.8. processed_bubble MATLAB code

```

1 % Read the original image
2 img1 = imread('no_processed_img.png');
3
4 % Convert to grayscale if RGB
5 if size(img1, 3) == 3
6     img1_gray = rgb2gray(img1);
7 else
8     img1_gray = img1;
9 end
10
11 % List of processed image filenames
12 processed_images = {'100_lines.png', '150_lines.png', '200_lines.png'};
13
14 % Number of processed images
15 num_images = length(processed_images);
16
17 % Figure 1: Original image
18 figure;
19 imshow(img1_gray);
20 title('Original Image');
21 saveas(gcf, 'Original_Image.png');
22
23 % Figure 2: Processed images
24 figure;
25 for idx = 1:num_images

```

```

26 % Read processed image
27 img2 = imread(processed_images{idx});
28
29 % Convert to grayscale if RGB
30 if size(img2, 3) == 3
31     img2_gray = rgb2gray(img2);
32 else
33     img2_gray = img2;
34 end
35
36 % Resize if needed
37 if ~isequal(size(img1_gray), size(img2_gray))
38     img2_gray = imresize(img2_gray, size(img1_gray));
39 end
40
41 % Plot processed image
42 subplot(num_images, 1, idx);
43 imshow(img2_gray);
44 title(sprintf('Processed %d', idx));
45 end
46 saveas(gcf, 'Processed_Images.png');
47
48 % Figure 3: Difference maps
49 figure;
50 for idx = 1:num_images
51     % Read processed image
52     img2 = imread(processed_images{idx});
53
54     % Convert to grayscale if RGB
55     if size(img2, 3) == 3
56         img2_gray = rgb2gray(img2);
57     else
58         img2_gray = img2;
59     end
60
61     % Resize if needed
62     if ~isequal(size(img1_gray), size(img2_gray))
63         img2_gray = imresize(img2_gray, size(img1_gray));
64     end
65
66     % Compute difference map
67     diff_map = imabsdiff(img1_gray, img2_gray);
68
69     % Plot difference map
70     subplot(num_images, 1, idx);
71     imshow(diff_map, []);
72     title(sprintf('Difference %d', idx));
73
74     % Optionally compute and display MSE
75     mse_val = immse(img1_gray, img2_gray);
76     fprintf('MSE for image %d: %0.4f\n', idx, mse_val);
77 end
78 saveas(gcf, 'Difference_Maps.png');

```