

FERNANDO MITSUYAMA CARDOSO

Estimation of elastic properties for tissue characterization based on ultrasound images

Sao Paulo

2015

FERNANDO MITSUYAMA CARDOSO

Estimation of elastic properties for tissue characterization based on ultrasound images

Thesis submitted to the School of Engineering
of the University of Sao Paulo in partial
fulfilment of the requirements for the degree of
Doctor of Science in Electric Engineering.

Supervisor: Prof. Dr. Sérgio Shiguemi Furuie

Sao Paulo

2015

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de março de 2015.

Assinatura do autor _____

Assinatura do orientador _____

Catlogação-na-publicação

Cardoso, Fernando Mitsuyama

Estimation of elastic properties for tissue characterization based on ultrasound imagens / F.M. Cardoso. -- versão corr. -- São Paulo, 2015.

148 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Simulação numérica 2.Ultrassonografia 3.Intravascular 4.Rastreamento 5.Elastografia 6.Elasticidade I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

À minha família

Agradecimentos

Agradeço ao professor Sérgio, que vem me orientando desde a iniciação científica até o doutorado. Sou grato pelo estímulo, sabedoria, presença, confiança e liberdade.

Agradeço também aos meus companheiros Mônica, Matheus, Maria Luiza, Edward, Diego, Verônica e Fernando Rangel.

À minha família, uma fonte de felicidade sempre ao meu lado.

Finalmente, agradeço à Fapesp pelo apoio financeiro.

A good professor impresses his students,
a great professor is impressed by them.

(Fernando Mitsuyama Cardoso)

CONTENT

List of Figures.....	i
List of Tables.....	vi
List of Abbreviations.....	vii
Abstract.....	viii
Resumo.....	x
1. Introduction.....	1
1.1 Ultrasonography.....	1
1.2 Cardiovascular Diseases.....	2
2. Objective.....	7
3. Fundaments and Bibliographic Review.....	9
3.1 Speckle Noise.....	9
3.2 Elastography.....	10
3.3 Speckle Noise Simulation.....	11
3.4 Speckle Filtering.....	11
3.5 Movement Estimation.....	20
4. Speckle Noise Simulation and Filtering.....	26
4.1 Speckle Noise Simulation.....	26
4.2 Speckle Noise Filters Performances.....	30
4.3 SNOL performance.....	37
5. Creation of US Images Using 3D phantoms.....	40
5.1 User Inputs.....	41

5.2	Phantom Deformation	48
5.3	Scatterers Movement	51
5.4	Ultrasound Simulation - RF data acquisition.....	54
5.5	Ultrasound Imaging – Attenuation.....	55
5.6	Ultrasound Imaging – B-mode image formation.....	56
5.7	Deformation Maps	59
5.8	Phantom Validation	60
5.9	Intravascular Ultrasound Phantoms	67
6.	Structures Movement Estimation in Transcutaneous US.....	85
6.1	Performance on Numeric Simulations	85
6.2	Performance on Physical Phantoms.....	89
7.	Structures Movement Estimation in Intravascular Ultrasound	93
8.	Conclusions and Discussions	97
8.1	Main contributions	98
8.2	Future works	99
9.	Publications	100
10.	References	102
11.	Appendix	106
11.1	Finite Elements Method Formulation	106
11.2	IVUS Simulation: 2D phantoms	112
11.3	IVUS Simulation: 3D phantoms	121

List of Figures

Figure 1: Left, illustration of a diagnostic ultrasound procedure. Right, B-mode ultrasound image of a lesionade bladder as visualized by the physician	1
Figure 2: Illustration of a lesioned coronary (atherosclerosis) [7].	2
Figure 3: Artery anatomy [13] (left) and IVUS image (right).	3
Figure 4: Catheter schematic representation	4
Figure 5: Cross sectional view of the catheter tip and its transducers. The transducers are sequentially activated in order to obtain information all around	5
Figure 6: IVUS segmented image. 1- adventitia, 2- media, 3- intima and 4- lumen	5
Figure 7: A is an example of calcified plaque; B, mostly fibrous plaque; C, thrombotic prominence closing the lumen; D, mostly fibrous plaque; E, mostly lipid plaque. F, lipid pool covered by a tissue with greater acoustic impedance	6
Figure 8: Blood vessel, whose lumen is reduced due to presence of atherosclerotic plaque (yellow)	6
Figure 9: Left, fibroadenoma ultrasonography. Right, respective elastography, in which cold colors represent hard tissue; and hot colors, soft tissue.	10
Figure 10: A) Numeric Speckled Image, B) Median Filtered Image, C) Constructive Interference only Image, D) Resulting Image.....	12
Figure 11: Explanatory diagram of SNOL.	20
Figure 12: Two-dimensional similarity search algorithm geometry. The kernel, with center x, y and size $2w + 1 \times 2h + 1$, from one image (a) is swept across the search region in the successive image (b) and the block matching algorithm calculates the position $x + dx, y + dy$ with the best similarity inside the search region.....	23
Figure 13: a) and b) US image before and after compression, respectively. c) and d) displacement map in the vertical direction before and after sub-pixel linear interpolation, respectively. e) Plot of the ‘step’ displacements along the vertical in c) (green) and the ‘ramp’ displacements along the vertical line in d) (red).	25
Figure 14: Speckle noise simulation using convolutional approach. Top – cross sectional slice of the simulated vessel. Bottom left – Speckle noise inclusion using 64 emitted beams and 40 MHz. Bottom right, Speckle noise inclusion using 128 emitted beams and 50 MHz.	28
Figure 15: Speckle noise simulation using Field II. Top – cross sectional slice of the simulated vessel. bottom left – Speckle noise inclusion using the solid-state transducer at 20 MHz. Bottom right, speckle noise inclusion using the mechanical transducer at 40 MHz.	29

Figure 16: a) Sim1 gold-standard image, b) Sim1 speckled image, c) Sim2 gold-standard image, and d) Sim2 speckled image. The dashed rectangles depict the homogeneous regions that were used to calculate mean and standard deviation; they were also used as reference region for the Lee filter and SRAD.	31
Figure 17: Illustration (one of 80) of Filtered Images: a) Gold-Standard Image, b) Median Filter, c) Adaptive Weighted Median Filter, d) Anisotropic Diffusion, e) Median Guided Anisotropic Diffusion, f) Speckle Reducing Anisotropic Diffusion, g) Lee, h) Bilateral filter, i) Squeeze Box Function, j) Symmetrical Speckle-Reducing Filter, k) Interference based Speckle Filter and l) Interference based Speckle Filter followed by Anisotropic Diffusion.	33
Figure 18: The dashed curves illustrate: a) and b) initial homogeneous seed of Fuzzy Connectedness; c) and d) initial contours on Gradient Vector Flow; e) and f) initial contours on Level Set.	35
Figure 19: Top, speckled image and progressively filtered images. Bottom, gold-standard image and direct application of median filter. The dashed rectangle in I_GS depicts the homogeneous region used to calculate the mean and standard deviation. The dashed horizontal lines in IF3, F5, F9, F17, F32 and IF32 indicate the position of the profile shown in Figure 20.	38
Figure 20: Horizontal profile of level variation of IF3, F5, F9, F17, F32 (red) and IF32 (blue).	39
Figure 21: Block Diagram. Step 1 depicts the procedure to obtain the simulated ultrasound images sequence. Step 2 illustrates the sequence for the movement estimation	40
Figure 22: Planar representation of the phantom creation using a 2D matrix.	42
Figure 23: GUI for entering the tissue properties.	43
Figure 24: Informative table of common tissue properties.	44
Figure 25: GUI for entering probe position and applied forces.	44
Figure 26: GUI for entering the transducer properties. The signal ‘#’ represents the word number.	46
Figure 27: Transducer representation with 16 elements, element width = $110\mu m$, element height = $250\mu m$, element kerf = $22\mu m$. The blue level denotes the apodization intensity.	46
Figure 28: Left, linear imaging, the ultrasound beams are parallel. Right, angular imaging, the ultrasound beams are radial.	47
Figure 29: Examples of the synthetic aperture of US beams in four different positions. Elements in red represents the active elements.	48
Figure 30: If the algorithm will consume more memory than available, the user is asked if he wishes to remesh to a lower resolution.	49
Figure 31: In blue, the original mesh. In green, the mesh after remeshing to a lower number of nodes.	49
Figure 32: Phantom mesh with the probe surface represented in blue. In the zoomed detail, it is possible to observe the force vectors applied at the probe surface nodes. The scale is in meters.	51

Figure 33: Left, tetrahedron before deformation. Right, tetrahedron after deformation.....	53
Figure 34: a) Two-cycle sinusoidal wave. b) Hanning Function and c) two-cycle sinusoidal wave weighted with a Hanning function.....	54
Figure 35: Phantom representation where the red fraction is the region considered by the transducer	55
Figure 36: Illustration of the downsampling procedure.	56
Figure 37: Left, distribution of the RF data according to their position during acquisition. Right, resulting image after interpolation.....	57
Figure 38: Right, polar image, RF data put side by side. Right, rectangular image, result after bilinear interpolation.....	57
Figure 39: GUI for resulting images manipulation of a simulated nodule scan.....	59
Figure 40: Two-layered models used to validate FEM accuracy. The yellow structure represents the transducer surface, in other words, the surface through which the forces are applied. The blue arrows represent the applied forces.	61
Figure 41: Strain contrast vs. modulus contrast for the model A (right) and model B (left). The blue lines show the strain contrast as measured from 3-D FEM and the red line depicts the hypothetical ideal case when the strain contrast is equal to the modulus contrast.	62
Figure 42: Phantom with two regions, where each color represents one region. The yellow structures represent the transducer elements. Each region has length=50mm, width=50mm and depth=60mm.....	63
Figure 43: Statistics of the speckle noise in relation to the scatterers density. Left, mean ratio; right, speckle index	65
Figure 44: Statistics of the speckle noise in relation to the scattering amplitude. Left, mean ratio; right, speckle index.....	66
Figure 45: Histogram of the speckle noise distribution (blue) and Rayleigh Distribution (red) of one of the simulations.....	66
Figure 46: Simplified sequence of the construction of 2D intravascular phantom. a) planar model, b) triangular mesh, c) forces due to variation in internal pressure, d) scatterers and US beam radial ultrasound beam (yellow arrow), e) distribution of the data acquired from 128 equiangularly distributed beams, f) simulated B-mode IVUS image.....	68
Figure 47: Volumetric model of a coronary with lesion (in yellow) and catheter (blue line inside the lumen).....	69
Figure 48: Angle θ between the direction of the predecessor (blue vector) and the next neighbor (green vector).....	70
Figure 49: Plot of the function $\cos t(\theta)$	71
Figure 50: 3 different ways to distribute a curvature that sums 90°	71
Figure 51: Representation of the numeric guidewire model.	72

Figure 52: a) Illustration of si with its original length and its two consecutive vertices $Vi - 1$ and Vi , b) case in which the length of si is greater than the original length and fiL is applied in Vi towards $Vi - 1$ and vice versa c) case in which the length of si is smaller than the original length and fiL is applied in Vi away from $Vi - 1$ and vice versa. The vertical dashed lines indicate Vi and $Vi - 1$ original position.....	73
Figure 53: Illustration of fC , which forces the alignment of $Vi - 1$, Vi and $Vi + 1$	74
Figure 54: Sequence of frames illustrating the active contour deformation to equilibrium.....	78
Figure 55: Four physical phantoms utilized during the comparison of performance of the Dynamic Programming and Active contours path simulators. The red line and the green line depict the path estimated by the dynamic programming and active contours, respectively.	79
Figure 56: a-d) Pullback sequence of a free-tip catheter. e-f) Pullback sequence of a sheath-covered catheter.....	81
Figure 57: Typical longitudinal cross-section obtained during IVUS pullback. The saw-blade shape “artifacts” are caused by cardiac motion.	83
Figure 58: Combination of IVUS images from different cardiac cycles.	83
Figure 59: blood vessel alignment of different cardiac cycles.....	84
Figure 60: Numerical phantom with cylindrical lesion.....	85
Figure 61: First frame of one of the simulations of each model.	87
Figure 62: Border region, in yellow, representing the position of the neglected results to minimize the effect of the boundaries.	87
Figure 63: Three-step process used to build a heterogeneous phantom with cylindrical inclusion	90
Figure 64: Visual assessment of the elastography performed in the four physical phantoms. From left to right, B-mode image, strain elastography from the equipment, elastography computed using 2D Block Matching and Optical Flow, respectively.....	92
Figure 65: Model of coronary with lesion (in yellow).	93
Figure 66: Simulated IVUS image and the points that were tracked: p1 at the left border of the lesion, p2 at the inferior limit of the lumen and p3 at the center of the lesion. The yellow horizontal line at the upper left shows length of 10 pixels.....	95
Figure 67: Performance of movement estimation using 2D Block Matching. a, b and c show the movement estimation error, in pixels, of the 16 first IVUS images of the sequence (red) and the 16 images from phase 1 (blue) considering p1, p2 and p3, respectively. d, e and f illustrate the movement estimation error of the whole IVUS images of the sequence (red) and the images from phase 1 (blue) considering p1, p2 and p3, respectively.	95
Figure 68: Performance of movement estimation using Optical Flow. a, b and c shows the movement estimation error, in pixels, of the 16 first IVUS images of the sequence (red) and the 16 images from phase 1 (blue) considering p1, p2 and p3, respectively. d, e and f illustrate the	

movement estimation error of the whole IVUS images of the sequence (red) and the images from phase 1 (blue) considering p1, p2 and p3, respectively.	96
Figure 69: Tetrahedral mesh with two elements.	108
Figure 70: Illustration of the construction of the global stiffness matrix from two element stiffness matrix.	109
Figure 71: Illustration of the matrix reduction.	111
Figure 72: Block Diagram. Step 1 depicts the procedure to obtain the simulated IVUS images sequence. Step 2 illustrates the sequence for the elastography estimation.	112
Figure 73: Numeric phantom that simulates the vessel cross-section. A: adventitia, I/M: intima/media, Lu: lumen and Le: lesion.	113
Figure 74: a) gold-standard image, b) image borders (initial set of nodes), c) mesh creation, d) nodes and pressure forces.	115
Figure 75: Mesh (left) and gold standard images (right) with initial pressure (top), after incremental pressure of 5.33kPa (middle) and 100kPa (bottom).	116
Figure 76: Scatterers imaging. Initial position (left) and after rotation of 30° CCW (right). The yellow arrow represents the direction of the emitting beam.	117
Figure 77: Stack of the 256 B-mode vectors after downsampling forming the polar image.	119
Figure 78: Radial distribution of the vectors throughout the image. The yellow pixel p(x,y) represents a pixel that was not covered by the lines, R and α represents the distance from the center, and angulation, respectively.	119
Figure 79: Interpolated image.	120
Figure 80: Sequence of B-mode images. From right to left, initial position, after compression of 5.33Pa and after compression of 100kPa.	120
Figure 81: Block Diagram. Step 1 depicts the procedure to obtain the simulated IVUS images sequence. Step 2 illustrates the sequence for the movement estimation.	121
Figure 82: Simulated coronary, vessel wall in light red, lumen in red and two lesions in yellow.	122
Figure 83: Generated mesh dividing the simulated coronary in tetrahedrons.	123
Figure 84: Force (blue vectors) applied at the nodes located at the lumen border.	124
Figure 85: a) vessel mesh before deformation, b) vessel mesh after deformation, c) vessel cross-section before deformation, d) vessel cross-section after deformation.	125
Figure 86: Translation (left) and rotation (right) top put the reference at the catheter tip.	127

List of Tables

Table 1: Types of Interaction in Relation to the Scatterers Size	10
Table 2: Parameters used on Field II.....	28
Table 3: Filters compared in this study, abbreviations and parameters used	30
Table 4: Numeric Performances of the Filters Considering 80 Images	32
Table 5: Two sets of parameters used for each segmentation technique	34
Table 6: Functional performances for two different sets of parameters to each of the three segmentation techniques	35
Table 7: Filters numeric performances.....	38
Table 8: Transducer properties.....	63
Table 9: Results for different scatterers density.....	64
Table 10: Results for different scattering amplitude	65
Table 11: Goodness of fit between the histogram and Rayleigh distribution	67
Table 12: Accuracy results of the guidewire path simulators	80
Table 13: Differences between the four numeric models.....	85
Table 14: Parameters used in the numeric simulation.....	86
Table 15: Movement Estimation Errors of Optical Flow on models A and B.....	88
Table 16: Movement Estimation Errors of Optical Flow on models C and D.....	88
Table 17: Movement Estimation Errors of 2D Block Matching with different configurations	89
Table 18: Composition and properties of the physical phantom structures	90
Table 19: Configuration used in the B-mode imaging.....	91
Table 20: Parameter used in this simulation	93
Table 21: Catheter longitudinal displacement (LD) in mm and pressure increase (PI) in mmHg for each phase	94
Table 22: Parameters of finite elements method and brightness.....	113
Table 23: Parameters of ultrasound probe simulation.....	118
Table 24: Parameters of finite elements method	125

List of Abbreviations

1.000: one

1,000: one thousand

AC: active contours

AWMF: adaptive weighted median filter

Bi: bilateral filter

BM: 2D block matching

CVD: cardiovascular diseases

DP: dynamic programming

FEM: finite elements method

GUI: graphical user interface

ICOV: instantaneous coefficient of variation

ISF: interference based speckle filter

ISFAD: interference based speckle filter followed by anisotropic diffusion

IVUS: intravascular ultrasound

IVUSSim: intravascular ultrasound simulator

MGAD: median-guided anisotropic diffusion

MR: modulus ratio

OF: optical Flow

ROI: region of interest

SBF: squeeze box function

SI: speckle index

SR: strain ratio

SRAD: speckle reducing anisotropic diffusion

SSRF: symmetrical speckle-reducing filter

SSIM: structural similarity

SUS: sistema único de saúde (unified health system - Brazil)

UltraSSim: UltraSound Simulator

US: ultrasound (transcutaneous ultrasound)

Abstract

Ultrasonography (US) is used by physicians to help on diagnosis and interventions. It provides tomographic views of inner organs such as pancreas, aorta, inferior vena cava, liver, gall bladder, bile ducts, kidneys and spleen. The physician may utilize US to perform only a visual assessment or may also compress the tissue to analyze its dynamics, since lesion elasticity may be related to dangerousness. Consequently, several computational procedures have been developed in order to provide information about the elastic properties of the tissue. However, a thorough and objective evaluation of US computational procedures may be hindered by the difficult access to US images with the desired features and the lack of gold-standard parameters. Therefore, we developed a tool that is able to create numeric phantom that mimics the compression induced by physician with the transducer. The tissue deformation was based on finite elements method and the displacement of the scatterers were calculated using linear isomorphism. After the scatterer displacement, Field II was used to simulate the speckle noise. Thus, it is possible to create a sequence of US images with realistic deformation. This technique was implemented in Matlab and is available for free download. The phantom deformation was validated by measuring the strain contrast from double-layered phantoms.

Special attention was given to cardiovascular diseases due to their impact on Brazilian and world populations. During the last decades, the prevalence of cardiovascular pathologies has increased progressively, and has become a serious public health problem. They are among the major causes of death, hospitalizations and health expenses. In interventionist practice, intravascular ultrasound (IVUS) is used to obtain information about blood vessels and eventual pathologies. Therefore, we also created numeric IVUS phantoms. The simulation of the blood vessel was also based in finite elements method with linear isomorphism. However, a reliable IVUS simulation must consider the catheter path inside the blood vessel, because it determines the position of the transducer. Hence, we developed a new method, based on equilibrium of forces, to determine the minimum energy position of the catheter. The method was validated by comparing its position with the position of a real stainless steel IVUS guidewire and presented root mean squared error and Hausdorff mean smaller than 1 mm for both.

We used two different techniques to track and estimate deformation of different structures in the simulated US images, namely, Optical Flow and 2D Block Matching. We applied an innovative implementation of 2D block matching with sub-pixel linear interpolation and displacement propagation. Then, the estimated deformation from both methods were compared with the numeric gold-standard, and 2D block matching presented better results than optical flow.

After the work with numeric phantoms, real US equipment was utilized to acquire B-mode images from a physical phantom. Then, we performed the movement estimation of the imaged tissue to analyze its morphological and dynamic properties. The results were compared to the elastography images provided by the US equipment. In accordance with the results from the numeric simulation, 2D block matching presented better results than Optical flow. Finally, we performed the two speckle tracking on a set of numerically simulated IVUS images, where the images were divided into two sets of frames. The first set, S1, contained all the frames from the IVUS sequence and the second set, S2, contained only the frames corresponding to a specific phase of the cardiac cycle. Thus, we analyzed the trade-off between the impact of the cardiac motion and low frame rate. For the points located at the edges of the object, optical flow had a good performance for both S1 and S2. In homogenous regions, however, optical flow was able to track the points only in S2, suggesting that it is better to work with low frame rate and reduced cardiac motion, as in EKG-triggered IVUS acquisition. 2D block matching presented poor results in all points of both S1 and S2.

Besides the simulation of ultrasound acquisition with deformation and structure tracking, in this work, we also developed a new filtering technique that is able to remove the speckle texture without blurring the edges. The proposed filter presented the best results when compared to other nine filters from the literature. We also developed a metric that uses the speckle texture of the image to provide a parameter that may help the user decide the size of the window of the filter.

Resumo

Título: Estimativa do movimento de estruturas em imagens ecográficas

Ultrassonografia (US) é usada por médicos para ajudar em diagnósticos e intervenções. Ela fornece uma visada tomográfica de órgãos internos como, por exemplo, pâncreas aorta, fígado, bexiga, rins e baço. O médico pode utilizar a US para realizar apenas uma avaliação visual ou pode também comprimir o tecido para analisar a dinâmica, uma vez que a elasticidade da lesão pode estar relacionada à patologias. Conseqüentemente, diversos procedimentos computacionais vem sendo desenvolvidos com o intuito de fornecer ao médico informações acerca das propriedades elásticas do tecido. Entretanto, uma avaliação completa e objetiva dos procedimentos computacionais sobre US pode ser dificultada pelo difícil acesso a imagens com as propriedades desejadas ou pela falta de padrão-ouro para ser utilizado como referência. Portanto, nós desenvolvemos uma ferramenta capaz de criar *phantoms* numérico que imitam a compressão induzida por um médico através de um transdutor. A deformação do tecido é baseada em método de elementos finitos e o deslocamento dos espalhadores é calculado usando isomorfismo linear. Depois do deslocamento dos espalhadores, Field II foi utilizado para simular o ruído Speckle. Assim, foi possível a criação de uma sequência de imagens de US com deformação realística. Este método foi implementado em Matlab e está disponível para download sem custos. A deformação do *phantom* foi validada através da medição de contraste de compressão em *phantoms* de duas camadas.

Uma atenção especial foi dada a doenças cardiovasculares devido ao impacto que essas patologias causam no cenário médico do Brasil e do mundo. Nas últimas décadas, a prevalência de patologias cardiovasculares têm crescido progressivamente e se tornou uma séria questão de saúde pública. Elas estão entre as maiores causas de mortes, internações e gastos com saúde. Na prática intervencionista, o ultrassom intravascular (IVUS) é utilizado para obter informações do vaso sanguíneo e de eventuais patologias. Portanto, foi desenvolvida também uma simulação numérica de *phantoms* de IVUS. A simulação do vaso sanguíneo também se baseou em método dos elementos finitos e isomorfismo linear. Contudo, uma simulação confiável de IVUS deve considerar o caminho do cateter no interior do vaso sanguíneo, porque isto determina a posição do transdutor. Portanto, nós desenvolvemos um novo método, baseado em equilíbrio de forças, para determinar a posição de menor energia do cateter. O método foi validado através da comparação da posição estimada com a posição de um fio-guia de aço real e apresentou erro quadrático médio e média Hausdorff menor que 1 mm para ambos.

Foram utilizados dois métodos diferentes para rastrear e estimar a deformação de determinadas estruturas no tecido: *Optical Flow* e *2D Block Matching*. Foi aplicada uma implementação inovadora de *2D Block Matching* com interpolação sub-pixel linear e propagação de deslocamento. Posteriormente, a validação será feita comparando-se a o movimento estimado com o padrão-ouro numérico utilizado para construir a simulação. O *2D block matching* forneceu resultados melhores que o *optical flow*. Após a análise em *phantoms* numéricos, equipamento real de ultrassom foi utilizado para aquisição de imagens modo-B de *phantoms* físicos. Então, nós realizamos a estimativa do movimento das estruturas para analisar as propriedades morfológicas e dinâmicas do tecidos. Os resultados obtidos foram comparados com a elastografia fornecida pelo equipamento. Em acordo com os resultados obtidos na simulação numérica, o *2D block matching* novamente apresentou resultados melhores que o *optical flow*. Finalmente, os dois métodos de rastreamento foi aplicada em imagens simuladas de IVUS, que foram divididas em 2 conjuntos de quadros. O primeiro conjunto, S1, continha todos os quadros da sequência de IVUS e o segundo conjunto, S2, continha apenas os quadros correspondentes a uma fase específica do ciclo cardíaco. Sendo assim, foi analisado o balanço entre o impacto do movimento cardíaco e a taxa de quadros. Para os pontos localizados nas bordas dos objetos, *Optical flow* teve um bom desempenho para ambos S1 e S2. Em regiões homogêneas, entretanto, o *optical flow* foi capaz de rastrear os pontos em S2, sugerindo que é melhor o trabalho com movimento cardíaco reduzido em detrimento da taxa de quadros, tal qual a aquisição de IVUS engatilhado por ECG. Já o *2D block matching* apresentou um mau rastreamento para todos os pontos selecionados.

Além da simulação da aquisição de ultrassonografia com deformação e do rastreamento de estruturas, também desenvolvemos uma nova técnica de filtragem capaz de remover a textura speckle sem borrar as bordas. A técnica proposta apresentou os melhores resultados quando comparados com outros nove filtros da literatura. Foi desenvolvida também uma nova métrica que utiliza o ruído speckle da própria imagem para fornecer um parâmetro que pode ajudar o usuário a decidir o tamanho da janela de um filtro.

1. Introduction

1.1 Ultrasonography

Ultrasonography is an imaging technique based on ultrasound (US). In most medical applications, a transducer emits the US beam that is reflected or scattered by the tissue through which it travels and the same transducer receives the reflected/backscattered US wave. Then, the information is converted into images to serve as a diagnostic tool for physicians [1] (Figure 1). Therefore, ultrasonography provides a tomographic view of the insonified tissue.

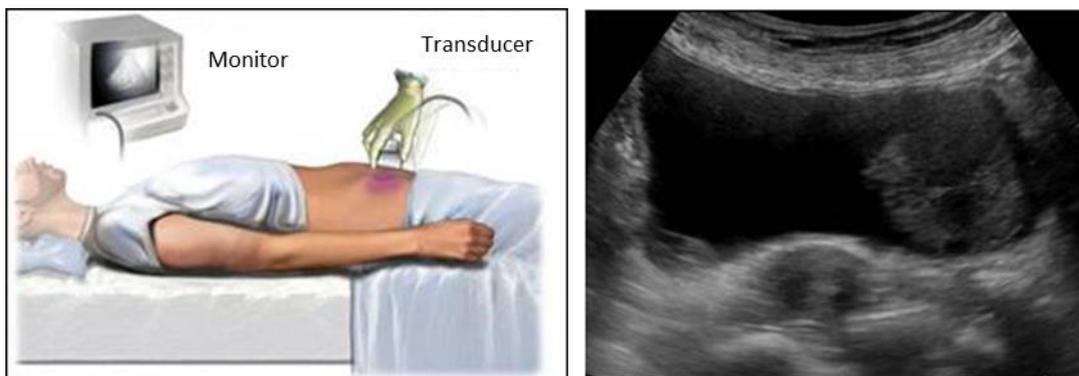


Figure 1: Left, illustration of a diagnostic ultrasound procedure. Right, B-mode ultrasound image of a lesioned bladder as visualized by the physician [2].

Because of its principle of formation, ultrasonography has an inherent image noise called speckle. Speckle noise is a granular texture produced by mutual interference of a set of scattered wavefronts. Depending on the phase of the wavefronts, the interference may be constructive or destructive, which results on brighter or darker pixels, respectively [3].

Among the numerous applications of US imaging, some common uses are imaging inner organs, such as pancreas, aorta, inferior vena cava, liver, gall bladder, bile ducts, kidneys, and spleen; guiding needles when injecting local anesthetic solutions near nerves or collecting tissue sample for biopsy; and checking on the development of the fetus during pregnancy.

Ultrasonography has been widely used for more than 50 years, however several researches are continuously being developed to improve its diagnostic capability or even to create new applications [4]. For instance, techniques related to segmentation and elastography are constantly being developed to work as an extra tool, because the US image alone may not allow the radiologist to have sufficient information to identify with precision the lesion composition and dangerousness [5].

Due to the importance of cardiovascular diseases, a special attention was dedicated to intravascular ultrasound (IVUS), which will be explained in Section 1.2.

1.2 Cardiovascular Diseases

Cardiovascular diseases (CVD) are the main cause of death in the planet – in 2008, 30% of all deaths were related to CVD [6]. The most common diseases are myocardial infarction, angina (chest pain), atherosclerosis and cerebral vascular accident (Figure 2).

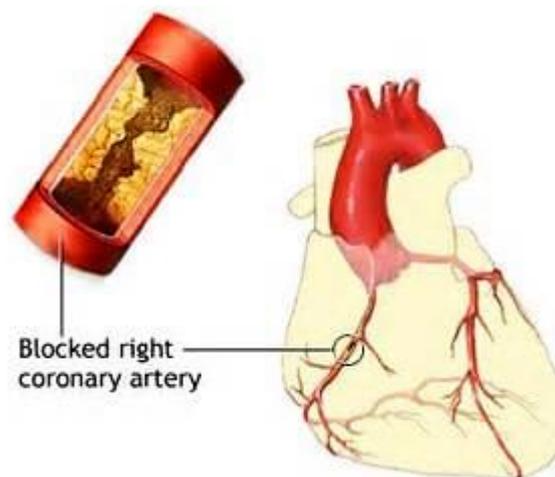


Figure 2: Illustration of a lesioned coronary (atherosclerosis) [7].

Cardiovascular diseases present an important impact on mortality, morbidity, ambulatory and hospitalizations expenses [8] in Brazil. The Brazilian Health Report (2011) indicates that CVD are responsible for around 30% of deaths. In 2003, 11% of inpatient hospitalizations on SUS were caused by problems in the cardiovascular system. In the same year, those diseases were responsible for 19.46% of hospitalizations expenses on SUS.

In 1930, they were responsible for 11.8% of deaths in the capitals of the country. In 1996, this percentage raised to 27.4% [9]. Such increase is related to longer life expectancy, which increases the old-aged population in the country. Other factors are also important, such as more sedentary lifestyle and fatty diet.

The situation is not different in developed countries. In the United States of America, the American Heart Association [10] estimated that around 64,400,000 people present at least one cardiovascular pathology and that 600,000 citizens die from CVD annually – corresponding to 25% of all causes of death. In Portugal, cardiovascular diseases are responsible for approximately 40% of death [11]. There have been significant developments in the global health and development agenda since the World Health Assembly in May 2012, where governments from 194 countries

agreed the first global mortality target on non-communicable diseases – including CVD – and made a commitment to reduce premature mortality by 25% by 2025 [12]. Considering those facts, several studies have been developed in order to enhance medical techniques, prevent cardiovascular diseases and improve outcomes in therapies and interventions.

1.2.1 Coronary Artery

Coronary arteries are the blood vessels that supply blood to the heart. Figure 3 illustrates the coronary anatomy and the coronary as seen by physicians during IVUS procedure, which will be explained in Section 1.2.2.

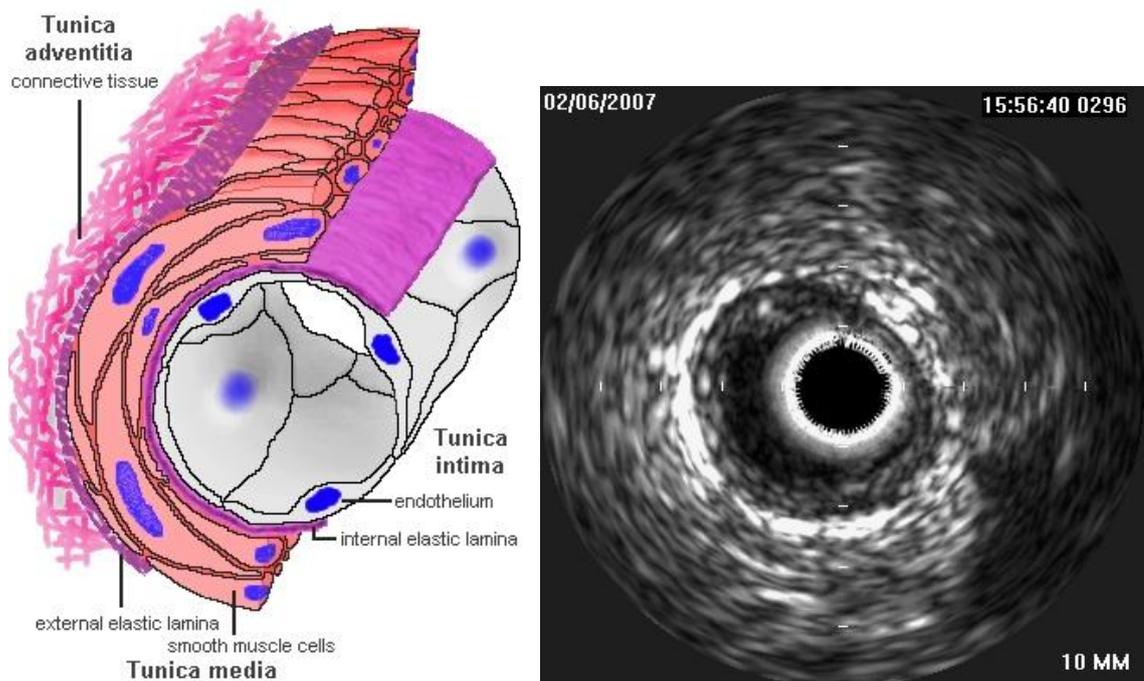


Figure 3: Artery anatomy [13] (left) and IVUS image (right).

Lumen is the channel through which the blood flows. It is the inner space delimited by the intima. Therefore, in pathologic cases, the lumen cross sectional area may decrease by the growth of the intima.

Tunica intima is an endothelial layer associated to connective tissue, which has direct contact with the blood in the vessel. In healthy arteries, this layer is thin. On the other hand, in pathologic cases, intima is thicker, asymmetric and eccentric, due to the presence of plaques and other tissues.

Tunica media is formed by a layer of circumferential smooth muscle and variable amounts of connective tissue. It is the actual vessel wall and can be identified in IVUS images as a black circle after the tunica intima.

Tunica adventitia, which is also known as tunica externa, consists mainly of connective tissue fibers. The tunica adventitia blends with the connective tissue surrounding the vessel. The definition of the outer limit of the tunica adventitia is therefore somewhat arbitrary.

The intima-media intersection is delimited by the internal elastic lamina. A second layer of elastic fibers, the external elastic lamina, delimits the tunica media from the adventitia.

1.2.2 IVUS

Intravascular ultrasound (IVUS) is an examination that provides real-time visualization of the interior of a blood vessel through catheterism. Its main target is the visualization of the coronaries, arteries that irrigate the heart.

1.2.2.1 Catheter

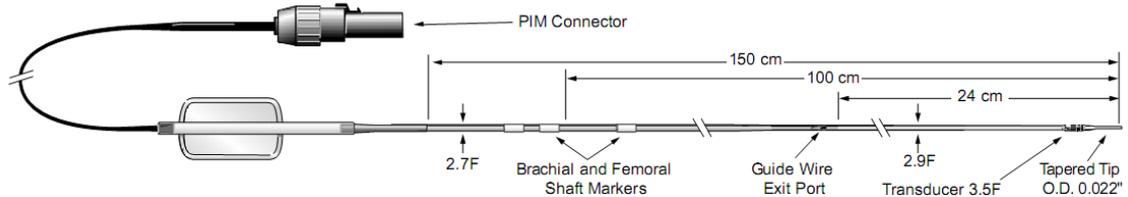


Figure 4: Catheter schematic representation [14]

IVUS catheter is a tube with transducers on its tip (Figure 4). Nowadays, there are two different types of IVUS catheter, both have approximately 1 mm diameter. One of them (Opticross™ Catheter, Boston Scientific) has only one transducer that spins mechanically around its longitudinal axis at 1800 rpm, emitting ultrasound pulses with frequency 20-40 MHz (Figure 5). Generally, the catheter is surrounded by a plastic sheath, which helps to guide the catheter and to keep a uniform rotational speed. The other one (EagleEye™ Catheter, Volcano) contains a series of fixed transducers (usually 64, at 20 MHz) equally spaced around the catheter longitudinal axis and they are activated sequentially. In both cases, the images are acquired at 10 to 30 frames per second. Those transducers emit ultrasound waves that are scattered/reflected when they reach the interface between two media with different acoustic impedance. The reflected wave is detected by the transducer and the signal is transformed in image.

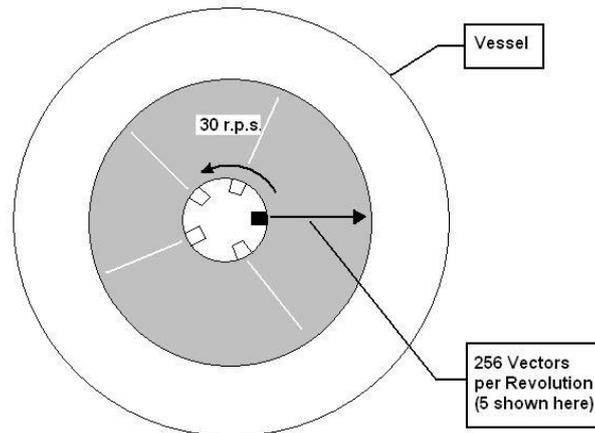


Figure 5: Cross sectional view of the catheter tip and its transducers. The transducers are sequentially activated in order to obtain information all around [15].

1.2.2.2 Data Acquisition

During IVUS procedure, the catheter, which was previously inserted into the patient, provides images of the blood vessel while it is pulled back. The pullback is performed by an external motor that retracts the catheter at a constant speed - typically 0.5 or 1.0 mm/s. So the images are acquired from the distal to the proximal part of the vessel in relation to the insertion position.

IVUS images are slices, cross-sectional to the catheter longitudinal axis, which provide information about the inner portion of the vessel, allowing the visualization and quantification not only of the vessel lumen, but also of the arterial wall and its structures, such as atherosclerotic plaque. Therefore, IVUS is an important instrument to assess the severity of a cardiovascular disease. In Figure 6, it is possible to observe segmented structures of an arterial wall [16].

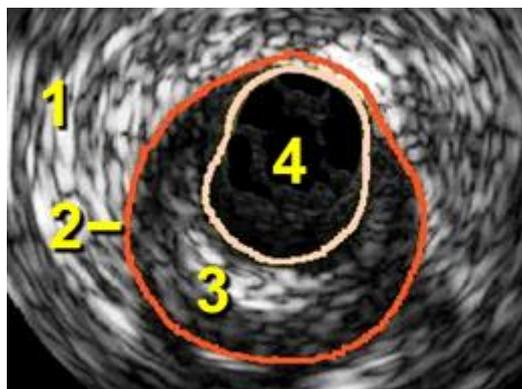


Figure 6: IVUS segmented image. 1- adventitia, 2- media, 3- intima and 4- lumen

Besides the measures of the dimensions of plaques, IVUS may also offer information about its composition by analyzing the echogenicity of the plaque (Figure 7). It is possible to distinguish

region with low echogenicity, high echogenicity or with extreme echogenicity and acoustic shadow, which are usually associated to lipid, fibrous and calcified lesions, respectively. The composition, morphology and mechanical properties of the vessel can be very important to characterize the risk of vascular diseases in patients.

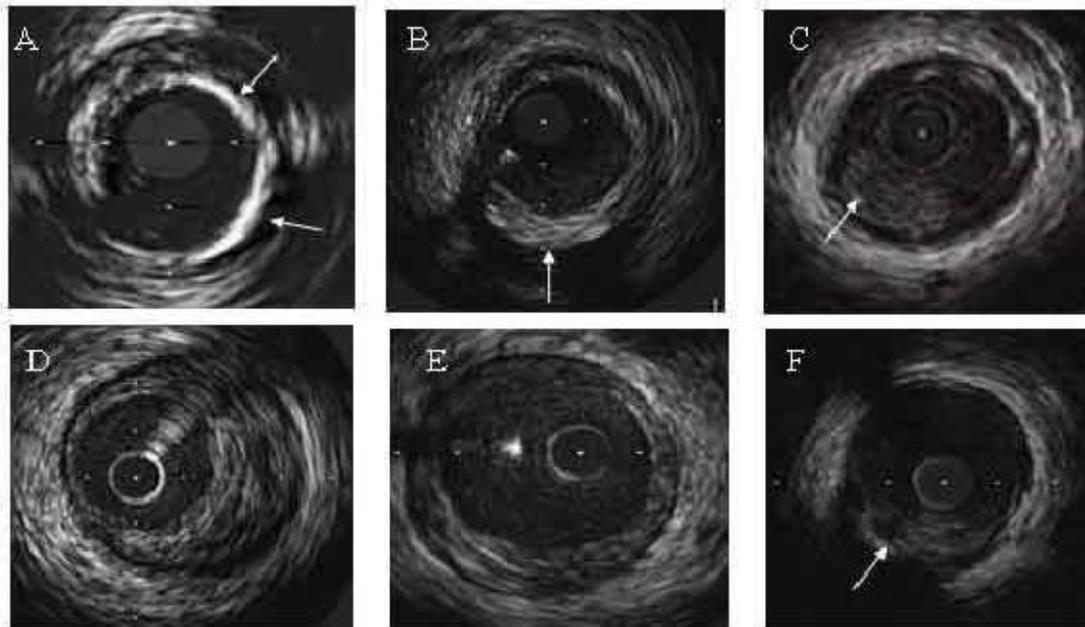


Figure 7: A is an example of calcified plaque; B, mostly fibrous plaque; C, thrombotic prominence closing the lumen; D, mostly fibrous plaque; E, mostly lipid plaque. F, lipid pool covered by a tissue with greater acoustic impedance [17].

Figure 8 illustrates how a lesion in the blood vessel may cause ischemia.

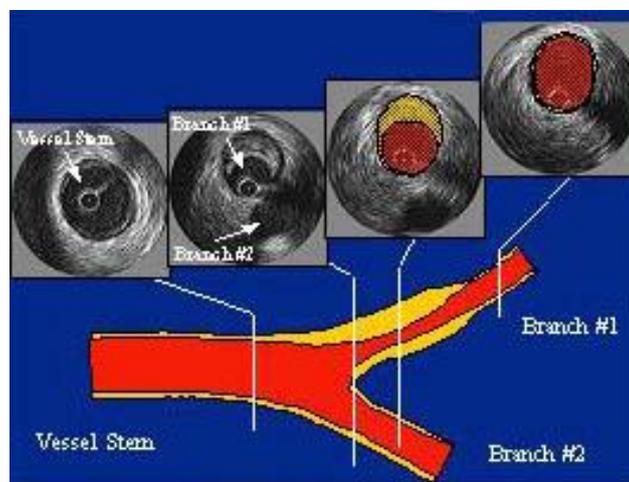


Figure 8: Blood vessel, whose lumen is reduced due to presence of atherosclerotic plaque (yellow) [18]

2. Objective

The main objective of this thesis is the estimation of mechanical properties of structures in tissues, in order to provide additional information for the characterization of lesions. It is based on ultrasound and involves calculation of movement and deformation of structures, tracking of points and elasticity estimation. Two scenarios were investigated: transcutaneous ultrasonography under different external pressure and intravascular ultrasound (IVUS) under natural cardiac pressure.

This study can be divided into five specific objectives:

- a) Numerous computational procedures involving ultrasound images are being developed. However, their evaluation may be problematic because it is difficult to access images with the desired features and usually the properties of the imaged tissue is not known with precision. An alternative solution is the numeric simulation of such images. Numerically simulated images give the user control over the properties of the tissue and have the advantage of providing a gold standard, which is important to enable a quantitative and precise evaluation of the computational procedures. Therefore, the first objective of this work is to include speckle noise on simulated ultrasound images as proposed by Bamber and Dickinson [19]-[21] and Jensen et al [22],[23], and compare their performances.
- b) Ultrasound images have an inherent noise called speckle, which makes the image more difficult to analyze. The second objective is to create, compare and evaluate a novel filtering technique, which is able to reduce speckle noise without blurring the edges of the imaged structures.
- c) Many computational procedures, e.g. speckle tracking, involve tissue deformation during ultrasound acquisition, such as the compressions induced with the transducer by the physician during US examination. In those cases, it is necessary to simulate a sequence of images corresponding to the deformation of the tissue. Hence, the third objective is to create, using finite elements method (FEM), deformable numeric phantoms that are able to provide information about the displacement of the US scatterers inside the tissue. Then, the information will be used to create a set of US images that simulates the image acquisition during deformation.

d) Due to the impact of cardiovascular diseases, this work also includes intravascular ultrasound. Although the method to simulate tissue deformation in IVUS may be similar to transcutaneous US, IVUS simulation has an important difference. The plane of the US imaging depends on the position of the catheter inside the blood vessel. Thus, the fourth objective of this work is to create a method to predict the position of the catheter inside the blood vessel.

e) The last objective is to locate the same structure in different compression stages (or different phases of the cardiac cycle), measure its movement and compare the results with the information from FEM. In addition, especially for IVUS, we will analyze the impact of the cardiac movement on the tracking performance.

3. Fundamentals and Bibliographic Review

In this section, we describe important concepts present in the literature that are related to this work.

First, the fundamentals of the speckle noise are explained, and then the concept of elastography is presented.

Speckle noise simulation allows us to include the characteristic noise of ultrasound images on gold-standard images. Two techniques to simulate speckle noise are presented. Next, two speckle filters that were developed in this work are presented along with the performance metrics that were used to compare the presented filters with other 9 filters. We also present a novel parameter that can be used to estimate automatically the window size of window-based speckle filters.

Finally, we present two methods - optical flow and 2D block matching - to track structures in a sequence of images. We can obtain information about the elasticity of those structures by analyzing their deformation.

3.1 Speckle Noise

Ultrasonography has an inherent image noise – speckle noise. It is a random granular texture produced by mutual interference of a set of scattered wavefronts.

Speckle noise is a multiplicative noise, i.e. it is related to the local grey level in any area and it is locally correlated. It depends on the ultrasound frequency and the acoustic impedance of the media through which it propagates [1].

The characteristics that govern the behavior of the ultrasound wave are ρ_0 , K and μ_a , mean density, adiabatic bulk elastic modulus and intensity absorption coefficient, respectively. Imaging systems aim to produce maps of the spatial distribution of these quantities, which is difficult to achieve. For perfect plane wave conditions, acoustic impedance is given by $Z = (\rho_0 \cdot K)^{1/2}$, which is a property of the medium only.

The structures that may scatter the ultrasound wave range in size from cell (10 μ m at 5MHz) to organ boundary (10cm at 5MHz). Depending on the imaged structure, different scattering phenomena may occur as shown in Table 1 [1].

Table 1: Types of Interaction in Relation to the Scatterers Size

Scale of interaction a: size of scatterer λ : wavelength	Frequency dependence	Scattering strength	Examples
$a \gg \lambda$ Geometrical region, ray theory for reflection and refraction	f^0	Strong	Diaphragm, large vessels, soft tissue, bone, cysts, eye orbit
$a \approx \lambda$ Stochastic region (diffractive)	variable	Moderate	Predominates for all structures (including in the other two categories)
$a \ll \lambda$ Rayleigh Region	f^4	Weak	Blood

3.2 Elastography

Elastography is an imaging technique that provides information about the elasticity of soft tissues. After applying compression on a tissue during imaging process, it is possible to measure the deformation level of different structures and build its elastograms, which may be very useful on detection of tumors and lesions (Figure 9).

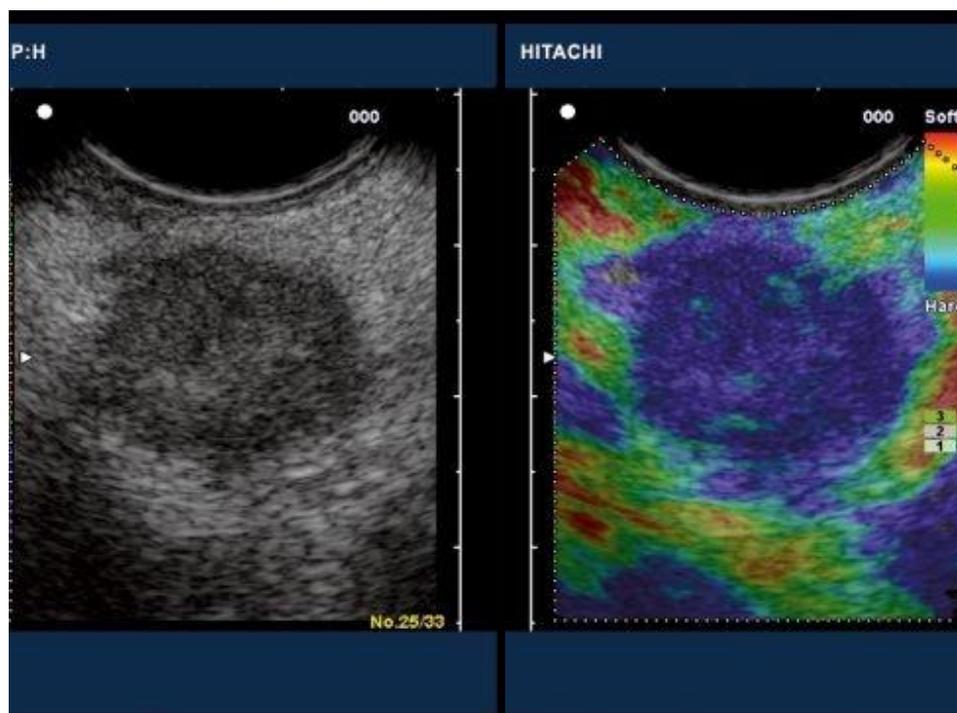


Figure 9: Left, fibroadenoma ultrasonography. Right, respective elastography, in which cool colors represent hard tissue; and warm colors, soft tissue. [24]

3.3 Speckle Noise Simulation

There are numerous computational procedures involving ultrasound images, such as filtering, segmentation and elastography. Unfortunately, the validation of those techniques is hindered by the lack of gold-standard parameters. An alternative solution is to simulate the ultrasound images. Thus, speckle noise simulation is a very common procedure in the literature [19]-[23], [25]-[27]. In this study, we included speckle noise using two different methods.

The first one was proposed by Bamber and Dickinson [19]-[20]. In the case of IVUS, an adaptation was performed due to the geometry of the transducers, which are not linearly organized, but circularly. The vessel cross-sectional slice was transformed from rectangular to polar coordinates, then speckle noise was included and the slice was transformed back to rectangular coordinates.

The second one is Field II, which was developed by Jensen *et al* [22]-[23], is a simulation that takes into account the transducer geometry. However, its complexity requires time to be processed and its numerous parameters make it difficult to work with.

Speckle noise simulation will be further explained on Section 4.1.

3.4 Speckle Filtering

Speckle noise makes recognition of the imaged structures more difficult for both humans and computers. Hence, we investigated several filtering techniques in order to attenuate its influence. We experimented 11 different filters including Interference based Speckle Filter (ISF) and Interference based Speckle Filter followed by Anisotropic Diffusion (ISFAD) [3], which are filters that were developed in this study and will be explained in the next sections. The comparison results are presented in Section 4.2. More details about the involved filters and methodology can be found in [3].

3.4.1 Interference based Speckle Filter (ISF)

Speckle noise is produced by mutual interference of a set of scattered wavefronts. Depending on the phase of the wavefronts, the interference may be constructive or destructive. ISF is a filter whose first step is to attenuate the destructive interference, because it carries little information about the imaged structures. For that purpose, for each pixel, the maximum between the median and the original value is selected. To eliminate the remaining bright speckles, we applied a median filter. The resulting image had minimized speckle effects.

ISF can be divided in three basic steps.

Step 1: Median Filter (Figure 10 B)

We applied a median filter to smooth the image. We have chosen a circular window instead of a square one to avoid “blocking” artifacts. Generally, the bigger the window radius the smoother gets the image texture. However, to avoid oversmoothing, the radius should not be bigger than image structures.

Step 2: Destructive Interference Suppression (Figure 10 C)

To smooth the image by eliminating only destructive interference pixels, for each position (x,y) , the maximum value between the median filtered image (I_{Median}) and the speckled image ($I_{Speckle}$) is selected.

$$\text{Eq. 1} \quad I_{Const.Int.}(x, y) = \max\{I_{Speckle}(x, y), I_{Median}(x, y)\}$$

where $I_{Speckle}$ is the original speckled image, I_{Median} is the median filtered image and $I_{Const.Int.}$ is the constructive interference only image.

Step 3: Constructive Interference Suppression (Figure 10 D)

The resulting image from step 2 had only bright speckles on a smooth background, so we applied another median filter with a small radius, e.g. 3.

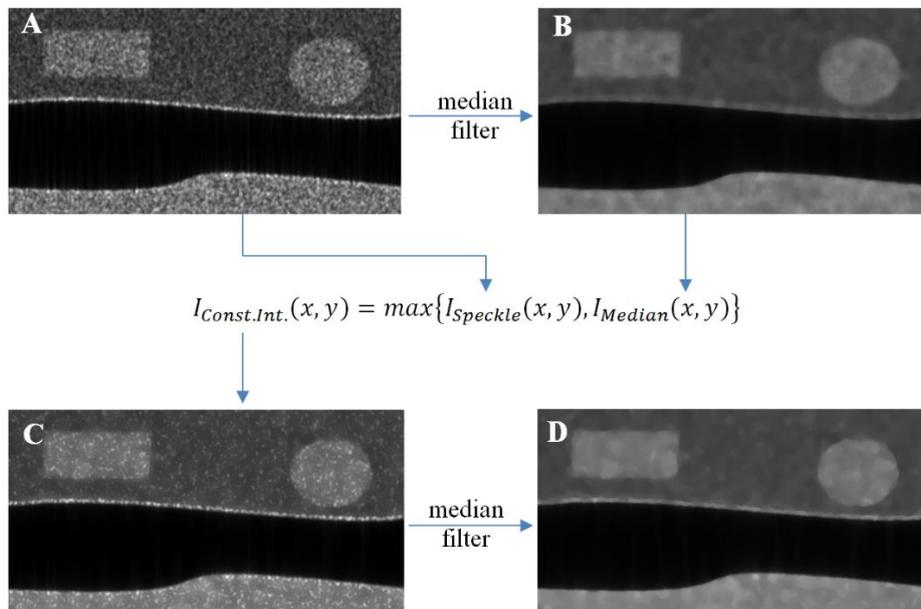


Figure 10: A) Numeric Speckled Image, which was simulated on Field II, B) Median Filtered Image, C) Constructive Interference only Image, D) Resulting Image

3.4.2 Interference based Speckle Filter followed by Anisotropic Diffusion (ISFAD)

ISF produces an image with well-defined edges and minimized speckle noise. Given that such images are good candidates for Anisotropic Diffusion, the performance of ISF followed by Anisotropic Diffusion (ISFAD) was also tested, by applying anisotropic diffusion to the output image of ISF.

3.4.3 Filters performance metrics

In this work, we utilized three different metrics to evaluate the numeric performance of the filters: Root Mean Square Error (RMSE), Structural Similarity Index (SSIM) and Ultrasound Despeckling Assessment Index (USDSAI). We also used different functional parameters: True positive Rate, False Positive Rate, Accuracy and Hausdorff Distance. The functional parameters evaluates the combined performance of the filters followed by computational procedures, which will be explained in Section 3.4.4.

3.4.3.1 Root Mean Squared Error (RMSE)

The RMSE measures, in average, how different each pixel of the filtered image is from the gold-standard image. In this study, RMSE was calculated as follows

$$\text{Eq. 2} \quad \text{RMSE}(I_{GS}, I_F) = \sqrt{\frac{\sum \sum (I_F(x,y) - I_{GS}(x,y))^2}{\sum \sum I_{GS}(x,y)^2}}$$

where I_F and I_{GS} represent the filtered and gold-standard images, respectively, and (x,y) represents the coordinates of the pixel.

3.4.3.2 Structural Similarity Index (SSIM)

SSIM measures the resemblance between two images and it is derived from the mathematically defined universal quality index. In this work, this index was used to compare the gold-standard images with the speckle-filtered images. SSIM takes into consideration three different comparisons: luminance (l), contrast (c) and correlation (s) [28].

$$\text{Eq. 3} \quad l(I_{GS}, I_F) = \frac{2 \cdot \mu_{GS} \cdot \mu_F + C_1}{\mu_{GS}^2 + \mu_F^2 + C_1}$$

$$\text{Eq. 4} \quad c(I_{GS}, I_F) = \frac{2 \cdot \sigma_{GS} \cdot \sigma_F + C_2}{\sigma_{GS}^2 + \sigma_F^2 + C_2}$$

$$\text{Eq. 5} \quad s(I_{GS}, I_F) = \frac{\sigma_{GS,F} + C_3}{\sigma_{GS} \cdot \sigma_F + C_3}$$

$$\text{Eq. 6} \quad \text{SSIM}(I_{GS}, I_F) = s(I_{GS}, I_F) \cdot l(I_{GS}, I_F) \cdot c(I_{GS}, I_F)$$

where I_{GS} , I_F , μ , σ^2 and $\sigma_{GS,F}$ are gold-standard image, filtered image, image mean, image variance and images covariance, respectively. The three constants are $C_1 = (K_1 \cdot d_r)^2$, $C_2 = (K_2 \cdot d_r)^2$ and $C_3 = C_2/2$, with $d_r = 255$ (8-bit image) representing the dynamic range of the ultrasound images. As suggested in [28], we used $K_1 = 0.01$ and $K_2 = 0.03$. After simplifications, SSIM is given by:

$$\text{Eq. 7} \quad \text{SSIM}(I_{GS}, I_F) = \frac{(2 \cdot \mu_{GS} \cdot \mu_F + C_1)(2 \cdot \sigma_{GS,F} + C_2)}{(\mu_{GS}^2 + \mu_F^2 + C_1)(\sigma_{GS}^2 + \sigma_F^2 + C_2)}$$

SSIM is calculated using an 8x8 sized sliding window in the whole window and the mean value is obtained. The mean value ranges from -1 to 1 ($-1 < Q < 1$), the highest value is 1 and occurs if $I_F = I_{GS}$, whereas the lowest value occurs if $I_F = 2 \cdot \mu_{GS} - I_{GS}$. Thus, -1 stands for a bad similarity between the original and resulting images and 1 stands for a good similarity between them.

3.4.3.3 Ultrasound Despeckling Assessment Index (USDSAI)

USDSAI is a metric that quantifies the improvement made by a filter to a speckled image [27]. This index is defined as

$$\text{Eq. 8} \quad \tilde{Q}(I_F) = \frac{Q(I_F)}{Q(I_{Sp})}$$

where

$$\text{Eq. 9} \quad Q(I) = \frac{\sum_{k \neq l} (\mu_k - \mu_l)^2}{(N-1) \sum_{k=1}^N \sigma_k^2}$$

I_{Sp} and I_F are the speckled and filtered images, respectively, N is the number of different structures in the image, and μ and σ^2 are the mean and variance of each structure, respectively. Effectively, USDSAI measures to what extent a filtered image had its speckle noise reduced. If a

filter smooths an image and reduces speckle, the denominator of Eq. 9 is decreased, and $Q(I_F)$ and $\tilde{Q}(I_F)$ are increased. On the other hand, if the speckle noise is left almost untouched, $Q(I_F)$ tends to be equivalent to $Q(I_{Sp})$ and thus $\tilde{Q}(I_F)$ is close to one

3.4.3.4 True Positive Rate, False Positive Rate and Accuracy

True positive rate (TP%) and false positive rate (FP%) were calculated by

$$\text{Eq. 10} \quad TP\% = \frac{|SE \cap GS|}{|GS|} \cdot 100\%$$

$$\text{Eq. 11} \quad FP\% = \frac{|SE - GS|}{|GS|} \cdot 100\%$$

where GS (gold-standard) is the set of pixels that belong to the structure in the original simulated image and SE (segmented) is the set of pixels inside the segmented region. The operator $| \cdot |$ represents cardinality of the set.

However, true and false positive rates should not be interpreted separately as they may lead to incorrect assessment of segmentation performance when considered individually. For instance, a segmentation can provide a 100% true positive rate if the segmented area includes the entire image, as that guarantees the object of interest is contained in the segmentation. In order to account for the interaction between true and false positive rates, in this work, a combined accuracy metric calculated was computed as

$$\text{Eq. 12} \quad AC\% = \frac{TP\% + (100 - FP\%)}{2}$$

3.4.3.5 Hausdorff Distance

The Hausdorff distance is the maximal distance of a set to the nearest point in another set [29]. In this work, the Hausdorff distance was used to obtain the maximal distance of the segmentation contour to the gold-standard reference set.

$$\text{Eq. 13} \quad HD(SE, GS) = \max_{s \in SE} \min_{g \in GS} \|s - g\|$$

Hausdorff mean, which is the mean value of the distances between the pixels in SE and the nearest pixels in GS, was calculated as follows:

$$\text{Eq. 14} \quad HM(SE, GS) = \frac{1}{n(SE)} \sum_{s \in SE} \min_{g \in GS} \|s - g\|$$

Given the synthetic nature of the datasets used in this study, the results for both the Hausdorff distance and mean are presented in pixels.

3.4.4 Computational procedures used to evaluate the functional filter performance

In this work, we measured the impact of the filters when applied prior to segmentation techniques: Fuzzy Connectedness, Gradient Vector Flow (GVF) and Level Set.

3.4.4.1 Fuzzy Connectedness

Fuzzy connectedness is a segmentation technique that uses fuzzy logic on its mathematical operations and takes into consideration two main parameters: Adjacency and similarity [30]. So, the closer and the more similar the region is to the seed, the stronger is the connectivity between them.

Adjacency relation assigns to every pair (c, d) of pixels a value between zero and one. The closer c and d are spatially to each other, the greater is this number. This is intended to be a ‘‘local’’ phenomenon. The fuzzy adjacency relation is denoted by α and the degree of adjacency assigned to any pixels (c, d) by $\mu_\alpha(c, d)$. Although $\mu_\alpha(c, d)$ may be a fuzzy relation, in this work we utilized a binary one, as proposed by [21].

$$\text{Eq. 15} \quad \mu_\alpha(c, d) = \begin{cases} 1, & \text{if } |c - d| \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Fuzzy affinity relation takes into consideration adjacency and similarity on its calculation. It determines the link strength between c and d .

$$\text{Eq. 16} \quad \mu_K(c, d) = \mu_\alpha(c, d) \cdot \exp - \left(\frac{f(c) + f(d) - 2m}{s} \right)^2$$

where m and s are the mean value and the standard deviation of the pixels intensity in the region of interest, respectively.

A path is a concatenation of a series of links. The path strength is given by the weakest link of the path.

$$\text{Eq. 17} \quad \mu_\rho(c, d) = \min[\mu_K(c^{(1)}, c^{(2)}), \mu_K(c^{(2)}, c^{(3)}), \dots, \mu_K(c^{(m-1)}, c^{(m)}), \mu_K(c^{(m)}, d)]$$

The connectivity between two pixels (c, d) is given by the strongest path of all possible paths between them:

$$\text{Eq. 18} \quad \mu_K(c, d) = \max(\mu_\rho(c, d))$$

In other words, to spread the connectivity calculation, *min-max* operation is computed.

The result is a fuzzy image whose pixels values represent the connectivity to the seed. A binary result may be obtained by thresholding the image.

3.4.4.2 Gradient Vector Flow (GVF)

Gradient Vector Flow (GVF) [31] is a variation of snakes, which is an active contour $\Gamma(s) = [x(s), y(s)]$, $s \in [0, 1]$, that moves through the spatial domain of an image to minimize the following parametric functional

$$\text{Eq. 19} \quad E = \int_0^1 \frac{1}{2} [\alpha |\Gamma'(s)|^2 + \beta |\Gamma''(s)|^2] + E_{ext}(\Gamma(s)) ds$$

where α and β are weighting parameters that control the snake's tension and rigidity, respectively, $\Gamma'(s)$ and $\Gamma''(s)$ denote the first and second derivatives of $\Gamma(s)$ with respect to s , and the external energy function E_{ext} is derived from the image. Let $I(x, y)$ be a gray-level image, a possible traditional external energy would be

$$\text{Eq. 20} \quad E_{ext}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2$$

where G_σ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator.

In order to minimize E , a snake must satisfy the following Euler-Lagrange equation

$$\text{Eq. 21} \quad \alpha \Gamma'''(s) - \beta \Gamma''''(s) - \nabla E_{ext} = 0$$

The solution is found by discretizing the equation and solving it iteratively with the time variable t .

$$\text{Eq. 22} \quad \alpha \Gamma'''(s, t) - \beta \Gamma''''(s, t) - \nabla E_{ext} = \Gamma_t(s, t)$$

when the solution stabilizes, the term $x_t(s, t)$ vanishes and we achieve a solution of Eq. 21.

Specially in GVF, $-\nabla E_{ext}$ is substituted by $\vartheta = [u(x, y), v(x, y)]$, which is the field that minimizes the following functional [31]

$$\text{Eq. 23} \quad \varepsilon = \int \int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 \cdot |\vartheta - \nabla f|^2 dx dy$$

where $f(x, y) = -E_{ext}(x, y)$ is the edge map.

3.4.4.3 Level Set

Level Set [32] is a variational formulation of active contours that consists of an internal energy term that penalizes the deviation of the level set function from a signed distance function, and an external energy term that drives the motion of the zero level set toward the desired image features, such as object boundaries. The resulting evolution of the level set function is the gradient flow that minimizes the overall energy functional. In this work, we implemented Level Set energy function as proposed by [32].

$$\text{Eq. 24} \quad \varepsilon(\phi) = \mu P(\phi) + \lambda \mathcal{L}_g(\phi) + \nu A_g(\phi)$$

where $\mu, \lambda > 0$ and ν are weighting constants and ϕ is curve function. The internal energy, length and area terms are given by

$$\text{Eq. 25} \quad P(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dx dy$$

$$\text{Eq. 26} \quad \mathcal{L}_g(\phi) = \int_{\Omega} g \times \delta(\phi) \times |\nabla \phi| dx dy$$

$$\text{Eq. 27} \quad A_g(\phi) = \int_{\Omega} g \times H(-\phi) dx dy$$

respectively, where δ is the univariate Dirac function, H is the Heaviside function, Ω is the signed distance function in \mathfrak{R}^2 dependent of the image and g is the edge indicator function

$$\text{Eq. 28} \quad g = \frac{1}{1 + |\nabla G_{\sigma} * I|^2}$$

where G_{σ} is the Gaussian kernel with standard deviation σ and I is the gray level image.

3.4.5 Automatic Window Size Estimation for Speckle Noise Filters

Besides the creation of two new filtering techniques – ISF and ISFAD – we also developed a method that provides information about the speckle noise present in B-mode images.

This method analyses the image that will be filtered in order to automatically compute the window size of the filter.

Let I_{Sp} be the original speckled image (Figure 11). Firstly, the set of local maxima, $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$, and minima, $\{\beta_1, \beta_2, \dots, \beta_n\}$, of I_{Sp} are located.

$$\text{Eq. 29} \quad \text{Maxima}(I_{Sp}) = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$$

$$\text{Eq. 30} \quad \text{Minima}(I_{Sp}) = \{\beta_1, \beta_2, \dots, \beta_n\}$$

Next, the map M_α with the same size as I_{Sp} is built. In this map, the pixels located in the local maxima position, $\alpha_1, \alpha_2, \dots, \alpha_m$, are set to 1 and 0 elsewhere (Figure 11). Analogously, the map M_β is built using the local minima, $\beta_1, \beta_2, \dots, \beta_n$.

$$\text{Eq. 31} \quad M_\alpha(p) = \begin{cases} 1, & \text{if } p = \alpha_1, \alpha_2, \dots, \alpha_m \\ 0, & \text{elsewhere} \end{cases}$$

$$\text{Eq. 32} \quad M_\beta(p) = \begin{cases} 1, & \text{if } p = \beta_1, \beta_2, \dots, \beta_n \\ 0, & \text{elsewhere} \end{cases}$$

Then, distance transform is applied to both M_α and M_β , creating D_α and D_β (Figure 11).

$$\text{Eq. 33} \quad D_\alpha = \text{distanceTransform}(M_\alpha)$$

$$\text{Eq. 34} \quad D_\beta = \text{distanceTransform}(M_\beta)$$

Next, the distance of each minima to the nearest maximum, $D_{min2max}$, is obtained by retrieving the values of D_α in the positions given by the local minima, $\{\beta_1, \beta_2, \dots, \beta_n\}$ (Figure 11). Analogously, the distance of each maxima to the nearest minimum, $D_{max2min}$, is obtained.

$$\text{Eq. 35} \quad D_{min2max} = \{D_\alpha(\beta_1), D_\alpha(\beta_2), \dots, D_\alpha(\beta_n)\}$$

$$\text{Eq. 36} \quad D_{max2min} = \{D_\beta(\alpha_1), D_\beta(\alpha_2), \dots, D_\beta(\alpha_m)\}$$

Finally, the mean value of the distances – considering both $D_{min2max}$ and $D_{max2min}$ – is used as a parameter to define the window size.

$$\text{Eq. 37} \quad R = \frac{D_\alpha(\beta_1)+D_\alpha(\beta_2)+\dots+D_\alpha(\beta_m)+D_\beta(\alpha_1)+D_\beta(\alpha_2)+\dots+D_\beta(\alpha_m)}{m+n}$$

This technique provides a parameter that measures the “oscillation length” of the noise. Therefore, we will call the proposed method speckle noise oscillation length, SNOL. Figure 11 depicts an explanatory diagram of the proposed method.

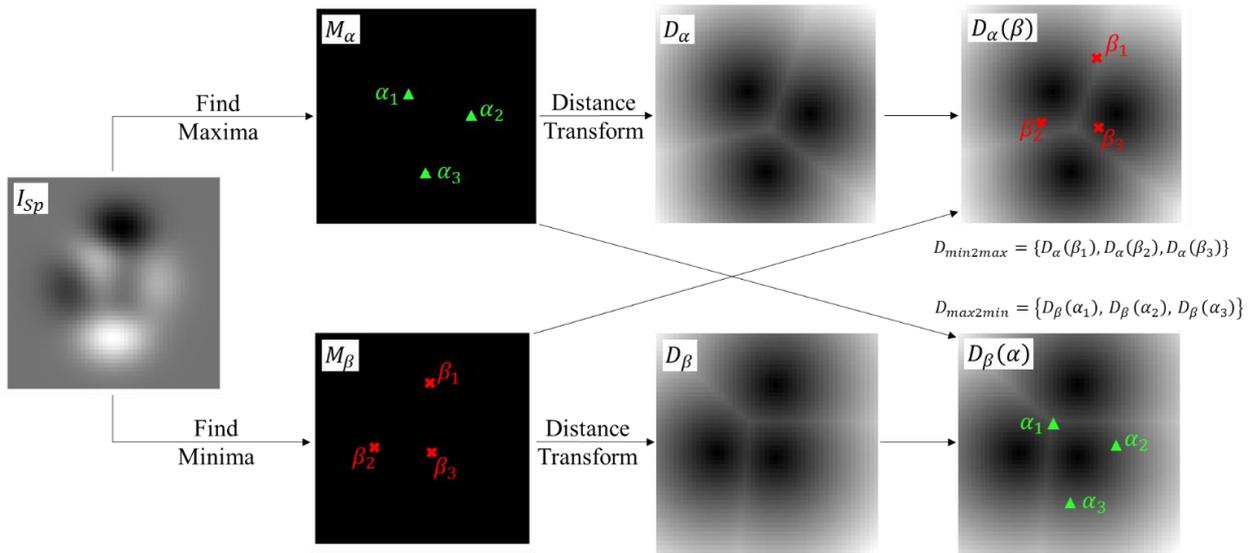


Figure 11: Explanatory diagram of SNOL.

If, after filtering with this parameter, the resulting image does not yet meet the user needs, he may reapply this method to the last filtered image. In this recursive way, the window size gets larger with the progressive filtering. As can be seen in the Section 4.3, the recursive method provides better results than a filter with large window size applied at once.

3.5 Movement Estimation

The movement estimation of the imaged tissue allows us to study its mechanical properties. As explained in Section 3.2, the mechanical properties of a tissue may be related to its health.

The main tissue mechanical property is the stiffness, which may be estimated through different techniques: RF signal processing, B-mode image processing, shear wave and acoustic radiation force impulse (ARFI).

The movement estimation in RF signal processing and B-mode image processing are very similar and will be explained together. The main difference between them is that the former analyses the signal received by the transducer and the latter utilizes the images formed with the information from the RF signal. In both cases, during the clinical ultrasound examination, a compression is applied to the tissue in order to deform it. The most common way to deform the tissue is to apply a light manual longitudinal compression/decompression using a conventional transducer, alternatively deformation can be produced by respiratory movements and in the case of IVUS, the compression is caused by the variation of the blood pressure variation. Then, deformation is calculated by measuring variations in RF signals or B-mode images, in the case of RF processing and B-mode image processing, respectively. Next, the profile of tissue deformation is converted to an elastic modulus from which an image called elastogram is derived.

The absolute value of the deformation along the longitudinal axis is proportional to the intensity of the compression force exerted. However, the force exerted by the compression is unknown to the equipment. The impossibility of defining the intensity of the force exerted allows calculation only of the deformability ratio of the various tissues and not the absolute elasticity. For this reason, elastography by compression provides only qualitative and not quantitative information.

Shear wave elastography is a transient elastography technique. An acoustic radiation force is induced by the ultrasound beam, which perturbs the insonified tissue. In turn, this vibratory acoustic force causes the tissues to 'react' with a restoring force, generating mechanical shear waves which propagate in a sliding fashion between tissue planes. The shear waves are then captured by a receptor in the ultrasound probe in 10-20 millisecond intervals (ultrafast imaging), approximately 100 times faster frame rate than conventional B-mode imaging. Shear waves are much slower compression waves (1—10 m/s) compared to the waves from conventional ultrasound B-mode imaging (1500 m/s).

ARFI can be used in two different ways. One is qualitative, as used in strain imaging which employs a short acoustic impulse of high intensity to deform the tissue elements and create a static map (elastogram) of the relative stiffness of the tissues. Another is quantitative, as used in shear wave elastography, which employs a primary acoustic impulse focused on a region of interest where it generates pressure waves in transverse propagation able to deform the tissues. The primary impulse is followed by a few interrogating impulses distributed in the surrounding tissues and designed to calculate the propagation velocity of pressure waves. Propagation velocity and attenuation of the waves are related to the stiffness and viscoelasticity of the tissue. The waves travel faster in stiff tissues than in non-stiff tissues. ARFI quantification provides pressure wave

velocity but not spatial distribution. Both the qualitative and quantitative variant of the ARFI method reduce interobserver variability but provide only static information and not dynamic information like elastography by compression.

In this work, we will focus on B-mode image processing. This can be performed using any conventional US equipment, which makes it inexpensive. In the next section, we will explain the two different methods that will be used to estimate the tissue movement: Optical Flow and 2D Block Matching.

3.5.1 Optical Flow

Optical Flow consists on tracking pixels with similar intensity in a sequence of image frames [33]. Consider a patch of the brightness pattern $E(x, y, t)$ that is displaced a distance δx in the x-direction and δy in the y-direction in time δt . If we assume the brightness of the patch is constant, then

$$\text{Eq. 38} \quad E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) = E(x, y, t) + E_x \delta x + E_y \delta y + E_t \delta t + \epsilon$$

where E_x , E_y and E_t are partial derivatives in x , y and t , respectively, and ϵ contains the higher orders of the derivatives, which will be neglected.

Therefore, after subtracting $E(x, y, t)$ from both sides and dividing by δt we have the following function, which must be minimized.

$$\text{Eq. 39} \quad E_b = E_x u + E_y v + E_t \rightarrow 0$$

where u and v are the speed components and E_x , E_y and E_t are partial derivatives in x , y and t , respectively. However, it is not possible to determine a solution yet. So, another equation is considered.

$$\text{Eq. 40} \quad E_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$$

The equation above is related to smoothness of movement through the images. Therefore, the solution will be the one with E_b closest to zero and the smoothest E_c , in other words, we need to minimize the following equation.

$$\text{Eq. 41} \quad E^2 = \iint (\alpha^2 E_c^2 + E_b^2) dx dy$$

where α is a weighting parameter.

3.5.2 2D Block Matching

2D Block Matching is a method used to estimate the tissue motion by comparing the similarity of the intensities from the ROI of an image to next image of the sequence. Usually, it is applied locally to find the displacement of a block of pixels between two successive frames, as illustrated on Figure 12 [34].

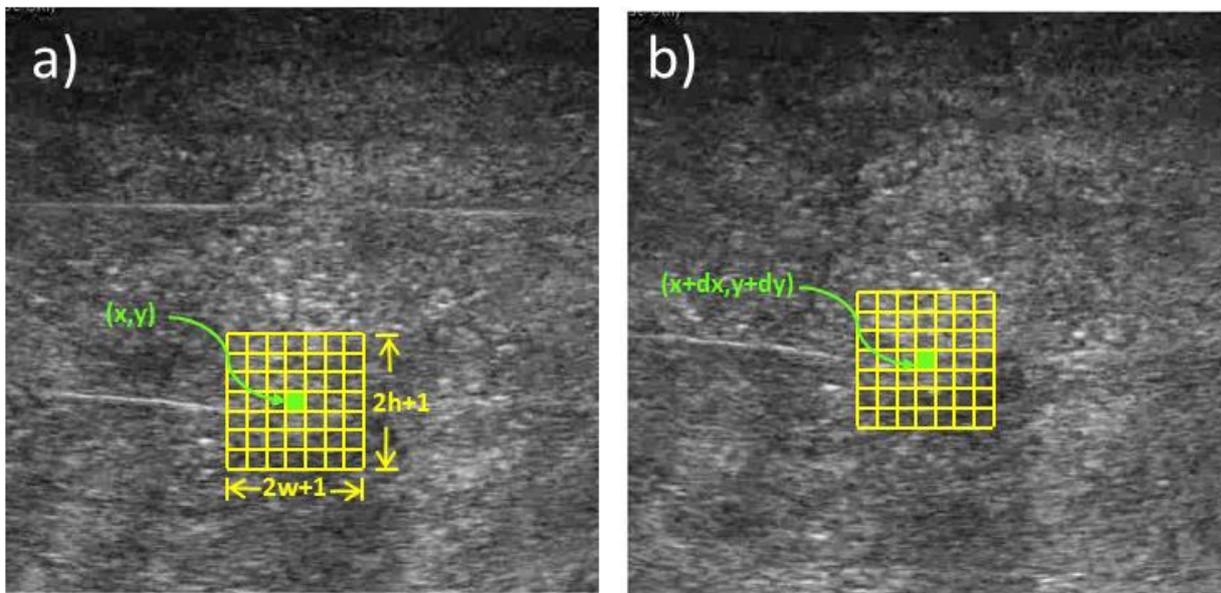


Figure 12: Two-dimensional similarity search algorithm geometry. The kernel, with center (x, y) and size $[2w + 1] \times [2h + 1]$, from one image (a) is swept across the search region in the successive image (b) and the block matching algorithm calculates the position $(x + d_x, y + d_y)$ with the best similarity inside the search region.

There are several criteria to estimate the pixels similarity. In this work, we compared the performance of three: Normalized Correlation Coefficient (NCC), sum of squared differences (SSD) and sum of absolute differences (SAD).

$$\text{Eq. 42} \quad SAD(x, y, d_x, d_y) = \sum_{k=-w}^w \sum_{l=-h}^h |I_b(x + k, y + l) - I_a(x + d_x + k, y + d_y + l)|$$

$$\text{Eq. 43} \quad SSD(x, y, d_x, d_y) = \sum_{k=-w}^w \sum_{l=-h}^h (I_b(x + k, y + l) - I_a(x + d_x + k, y + d_y + l))^2$$

$$\text{Eq. 44} \quad NCC(x, y, dx, dy) =$$

$$\frac{\sum_{k=-w}^w \sum_{l=-h}^h (I_b(x+k, y+l) - \bar{I}_b(x, y)) (I_a(x+d_x+k, y+d_y+l) - \bar{I}_a(x+d_x, y+d_y))}{\sqrt{\sum_{k=-w}^w \sum_{l=-h}^h (I_b(x+k, y+l) - \bar{I}_b(x, y))^2 \sum_{i=1}^l \sum_{j=1}^k (I_a(x+d_x+k, y+d_y+l) - \bar{I}_a(x+d_x, y+d_y))^2}}$$

where I_b and I_a are the image frames before and after compression, respectively. (x, y) are the coordinates of the center of the matching block, whose size is $[2w + 1] \times [2h + 1]$. (d_x, d_y) is the displacement of the block, and \bar{I}_b and \bar{I}_a are the mean pixel values of the corresponding blocks.

In the case of NCC, the displacement (d_x, d_y) that produces the maximum result is considered the new location of the block. On the other hand, for SSD and SAD, the minimum result will provide the optimal displacement.

In this work, we included two additional procedures in the Block Matching, namely sub-pixel linear interpolation and displacement propagation, which are able to increase resolution, reduce computation time and prevent kernel mismatching errors.

Sub-pixel linear interpolation

Usually, the displacement map provided by Block Matching consists of a series of steps (Figure 13c), because it is necessary a displacement with the size of a pixel to ‘trigger’ the displacement detection with Block Matching. In order to consider the sub-pixel displacement, the steps are converted to ramps (Figure 13d).

Let x_s and x_{s+k} , be the coordinates of the pixels of two consecutive steps. Then, the pixels between the steps $\{x_{s+1}, x_{s+2}, \dots, x_{s+k-1}\}$ are assigned a linear interpolated displacement value from the displacement of x_m and x_n .

$$\text{Eq. 45} \quad d(x_i) = d(x_s) \cdot \left(\frac{k-i}{k}\right) + \left(\frac{i}{k}\right) \cdot d(x_{s+k}), \quad i = 1, 2, \dots, k-1$$

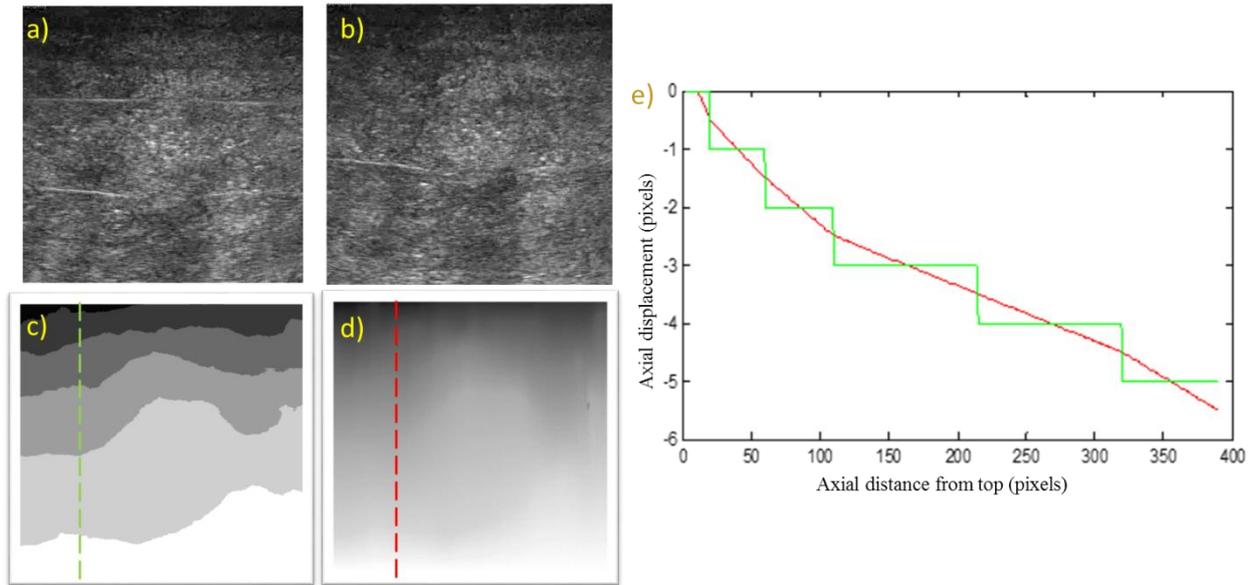


Figure 13: a) and b) US image before and after compression, respectively. c) and d) displacement map in the vertical direction before and after sub-pixel linear interpolation, respectively. e) Plot of the 'step' displacements along the vertical in c) (green) and the 'ramp' displacements along the vertical line in d) (red).

As seen in Figure 13, this technique takes into consideration sub-pixel displacement and increases displacement resolution.

Displacement Propagation

Let (d_x, d_y) be the displacement of the pixel (x, y) . In the case of transcutaneous US imaging, the frame rate is high in relation to the movement caused by the compression. Thus, the neighboring pixels should have a displacement between, $d_x - 1 < d'_x < d_x + 1$ and $d_y - 1 < d'_y < d_y + 1$, where (d'_x, d'_y) is the displacement of the neighboring pixel. Therefore, the search is restricted to a $[3 \times 3]$ region with center (d_x, d_y) . This procedure reduces computation time and prevents block misplacement, since speckle is a texture that may have similar patterns in different parts of the image.

In this work, for the transcutaneous US, the starting pixel was the pixel at the center of the top line, because that is the closest pixel to the transducer, which works as a static reference. Hence this position is likely to have the smallest relative displacement.

4. Speckle Noise Simulation and Filtering

There are numerous computational procedures involving ultrasound images, such as filtering, speckle tracking, segmentation and elastography. Thorough tests of those procedures is not always possible due to difficult access to clinical images with the desired parameters or due to the presence of artifacts. Besides, the validation of those techniques is usually hindered by the lack of gold-standard parameters. An alternative solution is to simulate the ultrasound images. Because of that, speckle noise simulation is a very common procedure in the literature [19]-[23], [25]-[27]. In Section 4.1, we tested the inclusion of speckle noise using two different techniques and performed a comparison between them.

Since speckle noise makes object recognition more difficult for both humans and computers, it is common to perform speckle noise filtering in order to improve the outcomes of the following procedures. In Section 4.2, we present a comparison of the performance of 11 different speckle noise filters – two of them were developed during this work.

4.1 Speckle Noise Simulation

In this study, we simulated speckle noise using two techniques: one proposed by Bamber and Dickinson [19]-[20] and the other by Jensen et al [22]-[23]. Both procedures are presented below as well as the comparison of their performances.

In order to obtain different simulations, we created 20 different scatterers random distribution for each method. The resulting images, therefore, differed from one another in speckle distribution.

4.1.1 Using the Convolutional Approach

Let the image sequence of cross sectional slices of a blood vessel as acquired on IVUS procedures be represented by $R_n(x, y)$. The coordinates x and y refer to horizontal and vertical axes, respectively; and the index n refers to the position of the slice in the image sequence.

The cross sectional slices are converted from Cartesian (rectangular) to polar coordinates, $P_n(r, \theta)$, where r and θ refer to radial and lateral axes, respectively; and n refers to the position of the slice in the image sequence. After the transformation, the transducers, which were circularly distributed and uniformly spaced, are organized on a linear array with equal intervals throughout θ axis.

To simulate the scatterers, Gaussian noise $G_n(r, \theta)$ with zero mean and small variance (e.g. $\sigma_G=0.01$) was included to the cross sectional slices [19].

$$\text{Eq. 46} \quad I_n(r, \theta) = P_n(r, \theta) \cdot G_n(r, \theta)$$

where $P_n(r, \theta)$ is a noiseless representation of contrast as in Figure 14.

The scattering-reflection is ruled by the following point spread function (PSF):

$$\text{Eq. 47} \quad h(r, \theta) = \sin(k_0 \cdot r) \cdot \exp\left(\frac{-r^2}{2\sigma_r^2}\right) \cdot \exp\left(\frac{-\theta^2}{2\sigma_\theta^2}\right)$$

where $k_0 = 2\pi f_0/c$, c is the ultrasound speed ($\cong 1540\text{m/s}$), f_0 is the center frequency, σ_r^2 is the signal pulse width in the axial direction and σ_θ^2 is the signal pulse width in the lateral direction. In this simulation, we used $\sigma_r = 1 \text{ mm}$ and $\sigma_\theta = 2.1^\circ$.

The function $h(r, \theta)$ is a separable function and may be rewritten as $h(r, \theta) = h_r(r) \cdot h_\theta(\theta)$, where $h_r(r) = \sin(k_0 \cdot r) \cdot \exp\left(\frac{-r^2}{2\sigma_r^2}\right)$ has an ondulatory characteristic (sinusoidal) with Gaussian attenuation in the radial direction, while in the lateral direction $h_\theta(\theta) = \exp\left(\frac{-\theta^2}{2\sigma_\theta^2}\right)$ there is only Gaussian attenuation.

Assuming a linear and space invariant imaging system, the detected signal is the convolution of the cross sectional slice and the PSF.

$$\text{Eq. 48} \quad V_n(r, \theta) = h(r, \theta) * I_n(r, \theta)$$

where $V_n(r, \theta)$ is the n -th cross sectional slice of the vessel.

Let $V_{Hn}(r, \theta)$ represent the Hilbert transform of $V_n(r, \theta)$, then the detected envelope amplitude is given by:

$$\text{Eq. 49} \quad A_n(r, \theta) = |V_n(r, \theta) + j \cdot V_{Hn}(r, \theta)|$$

Since the RF signals are equally spaced throughout θ axis, we considered only the following signals $A_n(r, \theta_m)$, where $\theta_m = \{m \cdot 2\pi/N_B, m \in \mathbb{Z} | 1 \leq m \leq N_B\}$, N_B is the number of emitted and received ultrasound beams.

In order to convert the image back to rectangular coordinates, we distributed A_n radially from the center of the rectangular image with angle (θ_m) . That way we simulated the data acquisition made by the transducers. We applied bilinear interpolation to fill the blank spaces that were not covered by the transducers.

Finally, we normalized the acquired value in relation to the noiseless image using the mean value of two homogeneous regions. Then, we rounded the value so there were only integer numbers and the images were stored as 8-bit.

That way, we obtained cross sectional slices with IVUS characteristic noise. Therefore, from this moment on we will refer to them just as images.

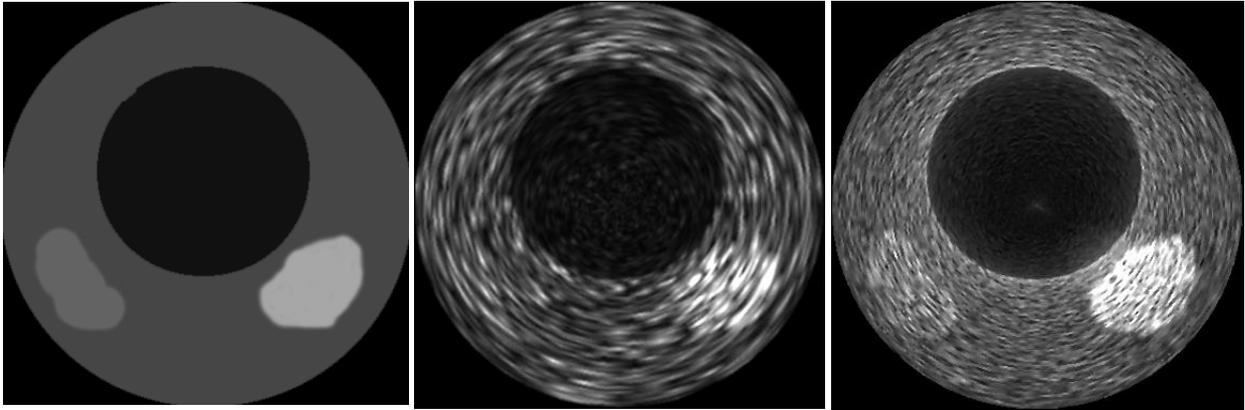


Figure 14: Speckle noise simulation using convolutional approach. Left – cross sectional slice of the simulated vessel. Center – Speckle noise inclusion using 64 emitted beams and 40 MHz. Right - Speckle noise inclusion using 128 emitted beams and 50 MHz.

4.1.2 Using Field II

Field II is a speckle noise simulator that takes into consideration the transducer geometry and properties, such as multiple foci zones.

We downloaded the Field II package through the website [23] and configured two different types of transducers: solid-state and mechanical (Table 2).

Table 2: Parameters used on Field II

Parameter	Symbol	Mechanical	Solid State
Number of elements	-	1	64
Catheter diameter	-	1mm	1mm
Transducer size	-	750x2250 μ m	46x700 μ m
Distance between transducers	-	-	3 μ m
Focal Point	-	8 mm	8 mm
IVUS resolution	Res_{IVUS}	80 μ m/pixel	40 μ m/pixel
Ultrasound center frequency	f_0	40 MHz	20 MHz
Sampling frequency	f_s	400 MHz	200MHz
Ultrasound speed	C	1540 m/s	1540 m/s

The excitation of the transducer consisted on a two-cycle sinusoidal wave and its impulse response is given by a two-cycle sinusoidal wave weighted with a Hanning window as suggested by [22]. The receiving transducer is the same as the emitter and the impulse response for the reception aperture is equal to the impulse response for the transmission aperture.

The simulated images are illustrated on Figure 15.

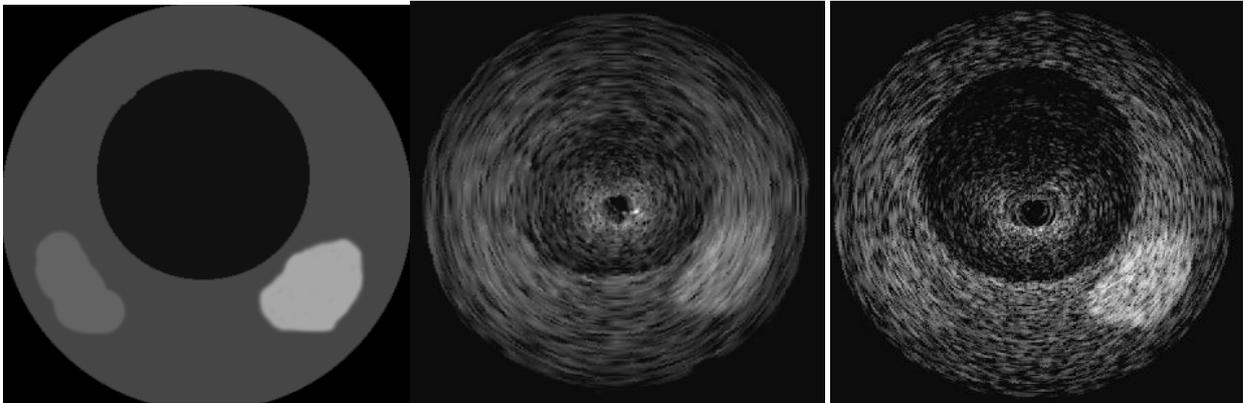


Figure 15: Speckle noise simulation using Field II. Left - cross sectional slice of the simulated vessel. Center - Speckle noise inclusion using the solid-state transducer at 20 MHz. Right - speckle noise inclusion using the mechanical transducer at 40 MHz.

4.1.3 Convolutional Approach and Field II comparison

The convolutional approach ignores the transducer geometry and assumes the following simplifications [1]:

- constant ultrasound speed
- no multiple scattering
- PSF spatially invariant
- no attenuation
- no system noise
- PSF separable into axial and lateral directions

Thus, it takes into consideration just a few parameters: signal pulse width in the axial and lateral directions, ultrasound speed and center frequency, number of transducers and Gaussian noise variance.

On the other hand, Field II is much more complex. It considers the transducer geometry: number of transducers, width, height, distance between them and focal point. It also requires numerous information about the ultrasound wave, impulse response, emitter/receiver aperture, etc. Many of the required parameters are not widely published by the hardware manufacturers (e.g. transducer focal point), which makes Field II hard to work with.

Since the convolutional approach is an open algorithm, it is easy to modify and improve the code. It is also possible to include new information in order to minimize the effects of the assumed simplifications.

The convolutional approach is up to 2000 times faster than Field II. Hence, there is a trade-off between realism and speed. The convolutional approach takes about 0.7s to simulate a speckled image, whereas Field II takes about 1500s.

Although Field II package is available on the internet - ready to use with examples, a lot of time is required to learn about the program and start using it properly.

Since Field II is more realistic and due to its acceptance in the scientific community [26], [27], in this work we will utilize only Field II to simulate the US system.

4.2 Speckle Noise Filters Performances

During this work, we also performed an extensive study dedicated to measure the performance of filters on removing speckle texture without blurring edges to improve object recognition, especially by computers. The study involved 11 different filters, including the filters developed during this work, ISF and ISFAD (Sections 3.4.1 and 3.4.2, respectively). Table 3 summarizes the set of speckle filters, including their abbreviation and the parameters used in this study. Those parameters were empirically selected to maximize Structural Similarity, which was defined in Section 3.4.3.2 of this work.

Table 3: Filters compared in this study, abbreviations and parameters used

Filter	Abbreviation	Filter Parameters
Median	Med	Radius = 15 pixels
Adaptive Weighted Median Filter	AWMF	Radius = 7 pixels , $c = 0.3 \text{ pixel}^{-1}$
Anisotropic Diffusion	AnisDiff	Num.Iter. = 600, $k = 1$, $\lambda \cdot \Delta t = 0.5$,
Median Guided Anisotropic Diffusion	MGAD	Num.Iter.= 100, $k = 1$, $\lambda \cdot \Delta t = 1$, $\beta=0.2$, Median Radius = 2pixel
Speckle Reducing Anisotropic Diffusion	SRAD	Num.Iter. = 400, $k = 1$, $\lambda \cdot \Delta t = 0.5$, reference region*
Lee	Lee	Radius = 10 pixels, reference region*
Bilateral Filtering	Bi	$\sigma_d = 5 \text{ pixels}$, $\sigma_r = 50$ (8-bit image)
Squeeze Box Function	SBF	Num.Iter.=1000, Gaussian noise added each 50 iter. with $\sigma = 0.1$
Symmetrical Speckle-Reducing Filter	SSRF	Window Size = 11 x 11 pixels, $T_g = 3$, $T_{VM} = 4$, $T_g' = 2$
Interference based Speckle Filter	ISF	Radii = 15 pixels and 3 pixels
ISF followed by AnisDiff	ISFAD	Radii = 10 pixels and 3 pixels, numIter = 500, $\lambda \cdot \Delta t = 0.5$, $k = 1$

*The reference region is the dashed rectangle in Figure 16a and Figure 16c

Num.Iter= number of iterations

We created a synthetic dataset of simulated ultrasound images for which ground truth is known and predefined, which facilitates objective evaluation. Therefore, speckle noise was

simulated using Field II (Section 4.1.2). In order to facilitate reproduction of this study, the ultrasound images of a cyst were simulated as proposed by [23], and will be referred to as Sim1. This simulation has also been used previously by [26], [27]. A second simulation, Sim2, which was generated with different characteristics, was also performed. The gold-standard synthetic and speckle-simulated images for both simulations are given in Figure 16.

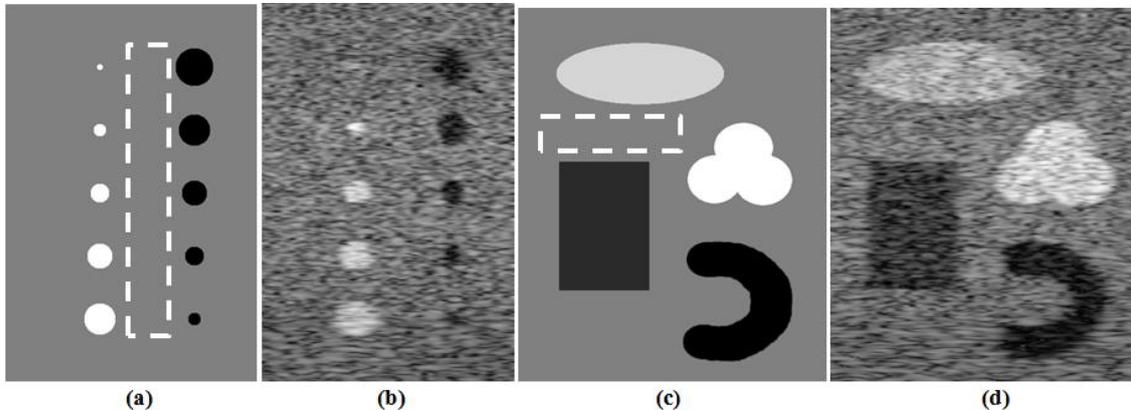


Figure 16: a) Sim1 gold-standard image, b) Sim1 speckled image, c) Sim2 gold-standard image, and d) Sim2 speckled image. The dashed rectangles depict the homogeneous regions that were used to calculate mean and standard deviation; they were also used as reference region for the Lee filter and SRAD.

Both simulations used the same parameters as [23] and generated 8-bit images with 600x400 pixels in Sim1 and 800x600 pixels in Sim2. Finally, for each simulation, two series of 20 test images with differing speckle distributions were generated. In one series the transducer center frequency was 3.5 MHz (as in [23]) and in the other 7.0 MHz. Hence, the two series have distinct speckle pattern. In other words, 40 different and uniformly distributed scatter-profiles were created to each simulation, gold-standard image. Next, the speckled images were filtered using the speckle filters listed in Table 3 and, then, evaluated.

Quantitative evaluation metrics were applied in this work to allow for a more accurate assessment of relative performance between the various filters investigated. Both numeric and functional evaluation criteria were considered and applied to 80 cases (2 models x 2 frequencies x 20 scatter distributions).

Numeric evaluation of the filters involved calculating the mean and standard deviation on a homogeneous region (Figure 16), and the RMSE, SSIM and USDSAI on the whole image.

The functional improvement offered by the filters in allowing for more accurate segmentations by various established algorithms on the post-filtered images was also evaluated. Functional evaluation involved first applying three different segmentation techniques, namely: Fuzzy Connectedness, Gradient Vector Flow and Level Set to segment the simulated cysts in speckle-noise images. Then, the true positive rate, false positive rate, accuracy, Hausdorff distance and Hausdorff mean between the algorithms segmentations and ground-truth were measured.

All the filters were applied to the simulated speckled images of Sim1 (Figure 16b) and the numerical metrics were calculated using the gold-standard image of Sim1 as reference (Figure 16a).

Table 4 provides the filters performance and Figure 17 illustrates the filtered images.

Table 4: Numeric Performances of the Filters Considering 80 Images

Filter	μ	Σ	RMSE $\times 100$	SSIM $\times 100$	USDSAI $\times 100$
Gold Std	59.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	100.00 ± 0.00	∞
Speckled	58.68 ± 2.22	10.85 ± 0.55	21.35 ± 1.04	47.92 ± 0.35	100.00 ± 0.00
Med	59.34 ± 2.25	2.68 ± 0.67	11.62 ± 1.33	93.60 ± 0.32	287.25 ± 6.72
AWMF	59.31 ± 2.26	2.84 ± 0.66	11.71 ± 1.33	93.34 ± 0.39	286.36 ± 5.73
AnisDiff	57.90 ± 2.19	4.40 ± 0.62	13.93 ± 1.45	87.49 ± 0.41	250.29 ± 4.26
MGAD	58.80 ± 2.27	2.17 ± 0.68	11.65 ± 1.52	90.16 ± 0.48	287.70 ± 4.51
SRAD	61.03 ± 2.01	1.93 ± 0.60	11.90 ± 2.58	93.64 ± 1.65	259.96 ± 7.07
Lee	57.89 ± 2.16	2.79 ± 0.91	13.46 ± 1.55	89.00 ± 2.28	202.06 ± 6.10
Bi	57.95 ± 2.22	4.10 ± 0.63	13.35 ± 1.47	88.88 ± 0.41	264.71 ± 4.20
SBF	59.20 ± 2.20	4.39 ± 0.60	13.07 ± 1.15	85.28 ± 0.59	232.88 ± 3.16
SSRF	58.92 ± 2.25	4.51 ± 0.66	13.21 ± 1.29	84.48 ± 0.59	284.79 ± 2.40
ISF	61.36 ± 2.22	2.45 ± 0.61	12.29 ± 0.73	93.67 ± 0.31	296.39 ± 5.26
ISFAD	61.63 ± 2.16	0.75 ± 0.61	10.28 ± 0.84	95.78 ± 0.19	350.46 ± 9.11

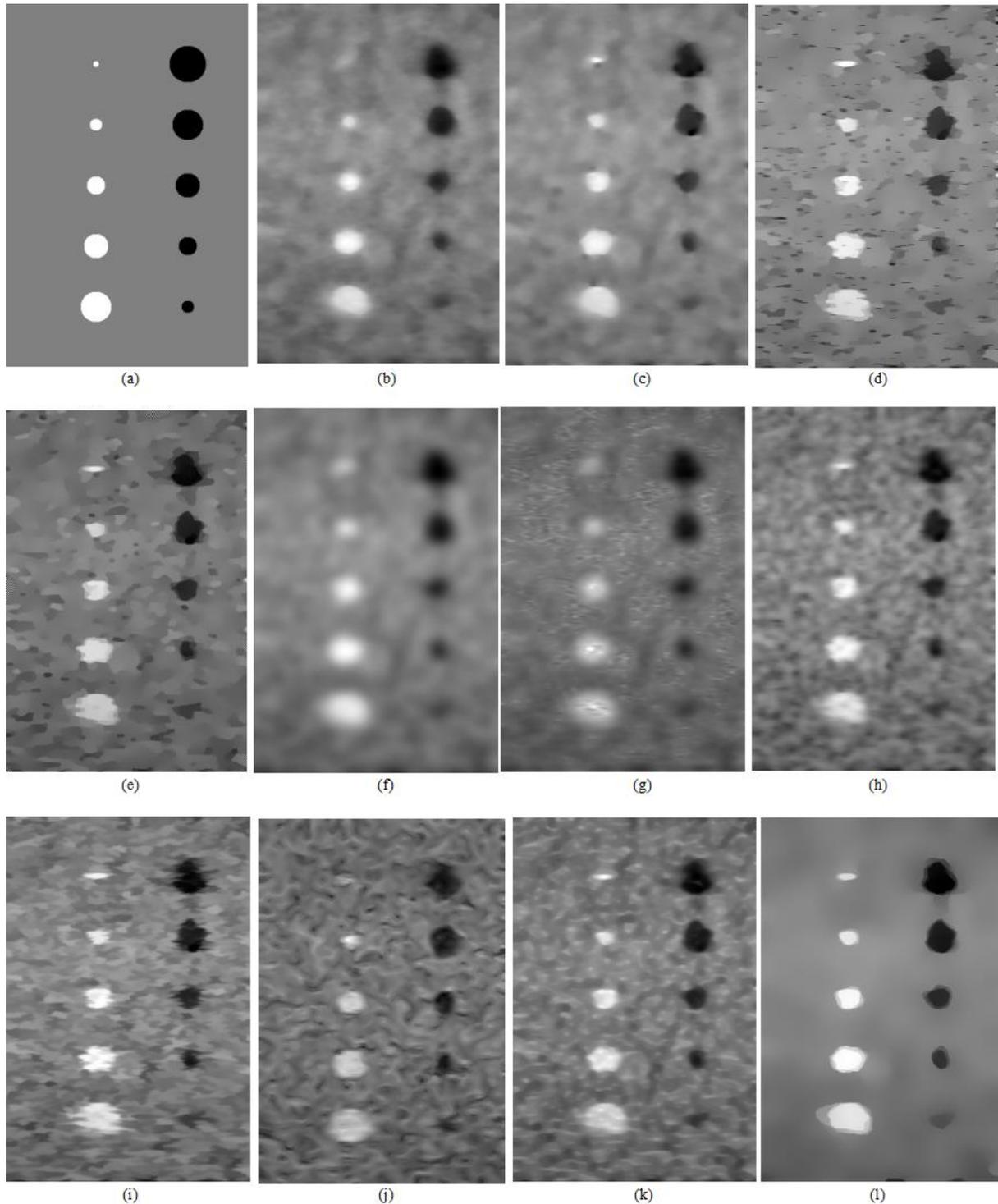


Figure 17: Illustration (one of 80) of Filtered Images: a) Gold-Standard Image, b) Median Filter, c) Adaptive Weighted Median Filter, d) Anisotropic Diffusion, e) Median Guided Anisotropic Diffusion, f) Speckle Reducing Anisotropic Diffusion, g) Lee, h) Bilateral filter, i) Squeeze Box Function, j) Symmetrical Speckle-Reducing Filter, k) Interference based Speckle Filter and l) Interference based Speckle Filter followed by Anisotropic Diffusion.

In order to obtain a functional evaluation, each of the three segmentation techniques were applied on the filtered images of Sim2 using two different sets of parameters for each technique, which were chosen empirically and are given in Table 5. Two different initialization curves were used for each set of parameters. In Figure 18, the dashed contours represent the initial contours for

GVF and Level Set, and the initial homogeneous seed of Fuzzy Connectedness. Difficult initializations, to be different from the real delineation, were used in order to assess more accurately the segmentation of the four types of structures in the filtered images. The results of the functional evaluation are summarized in Table 6. The standard deviations of the results were low, so they were omitted in Table 6 for clarity.

Table 5: Two sets of parameters used for each segmentation technique

Fuzzy Connectedness	Parameters1	Parameters2
Adjacency	4-neighbor binary relation	
Affinity Function	$\frac{(I(c)+I(d)-2.\mu_H)^2}{2\sigma_H}$	
Strength of Connectivity Threshold	0.8	0.9
Gradient Vector Flow	Parameters1	Parameters2
Regularization Parameter	0.2	0.4
Number of Iterations (GVF)	600	400
Elasticity	0.01	0.1
Rigidity	500	1
Viscosity	1	10
External Force Weight	0.1	0.8
Number of Iterations (Snakes)	100	100
Level Set	Parameters1	Parameters2
Dirac Function Parameter	1.5	1
Time Step	2	5
Internal Energy (Penalty) Coefficient	0.1	0.04
Length Coefficient	3	3
Area Coefficient	0.25	-0.8
Number of Iterations	6000	3000

*I(c) and I(d) are pixels intensities, μ_H and σ_H represents the homogeneous seed mean and standard deviation.
The parameters were chosen to maximize the SSIM

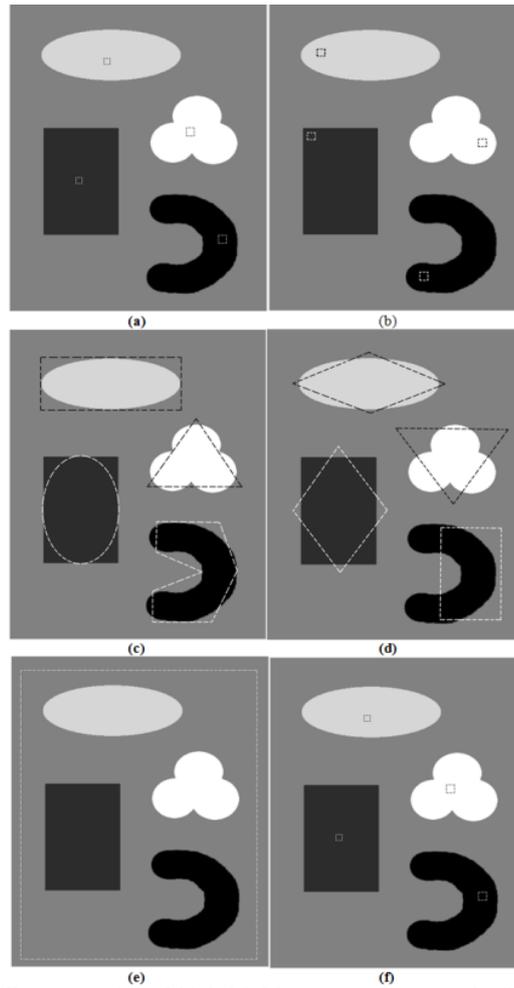


Figure 18: The dashed curves illustrate: a) and b) initial homogeneous seed of Fuzzy Connectedness; c) and d) initial contours on Gradient Vector Flow; e) and f) initial contours on Level Set.

Table 6: Functional performances for two different sets of parameters to each of the three segmentation techniques

Filter	Fuzzy Connectedness					Gradient Vector Flow					Level Set				
	TP %	FP %	AC %	HD	HM	TP %	FP %	AC %	HD	HM	TP %	FP %	AC %	HD	HM
Gold Std	100	0.0	100	0.0	0.0	100	0.0	100	0.0	0.0	100	0.0	100	0.0	0.0
Speckled	92.8	89.2	51.8	154.4	50.5	74.7	21.7	76.5	48.8	17.6	50.6	71.6	39.5	94.3	46.6
Med	95.4	6.2	94.6	20.3	5.4	82.0	22.5	79.7	46.9	15.1	74.2	12.0	81.1	63.4	15.3
AWMF	96.2	7.0	94.6	28.9	5.9	83.0	23.4	79.8	48.9	15.1	75.8	6.8	84.5	58.1	11.0
AnisDiff	93.8	6.0	93.9	60.0	15.6	81.5	30.4	75.6	56.8	17.8	81.6	20.2	80.7	109.2	18.1
MGAD	94.6	5.1	94.8	44.4	13.0	68.1	32.1	68.0	82.6	39.9	62.9	17.4	72.7	96.2	21.7
SRAD	95.9	5.9	95.0	11.6	4.5	79.5	21.8	78.9	51.0	17.0	50.0	97.1	26.5	133.8	65.5
Lee	97.9	152.4	22.7	165.5	63.1	72.6	31.6	70.5	64.3	22.7	45.0	32.0	56.5	86.3	27.6
Bi	97.1	8.4	94.4	31.4	6.9	80.0	25.2	77.4	62.5	16.4	57.6	17.5	70.0	65.5	19.9
SBF	96.6	10.2	93.2	32.9	7.5	82.6	17.9	82.3	48.0	14.5	51.9	38.0	57.0	88.8	32.5
SSRF	96.1	8.5	93.8	50.0	11.4	77.3	27.6	74.9	52.0	19.3	55.8	48.4	53.7	93.6	35.1
ISF	97.5	8.5	94.5	23.0	5.2	80.8	25.1	77.9	50.2	17.0	72.2	9.6	81.3	62.8	14.6
ISFAD	95.5	4.2	95.7	13.8	4.1	90.8	9.7	90.5	41.5	10.7	93.7	5.7	94.0	45.9	4.6

As shown in Table 4, all filters were able to decrease the standard deviation and increase the SSIM and USDSAI. All median-based filters presented SSIMs greater than 0.9 and USDSAI greater than 2.8, SRAD and MGAD obtained good results in terms of numeric evaluation, but they notably blurred edges, thus leading to poor segmentation.

AnisDiff, MGAD, SBF and SSRF were also able to remove the speckle noise, but introduced a new texture pattern with well-defined borders, and were associated with poor segmentation performance.

As can be seen in Table 6, many filters allowed the segmentation algorithms to attain a TP% close to 100%. However, many of these segmentations are also associated with a high FP%, indicating over-segmentation. It is also possible to observe cases in which objects were under-segmented, where a low FP% is followed by a low TP%. AC% takes into consideration both TP% and FP%, thus over-segmentation and under-segmentation both lead to a low AC%. ISFAD resulted in good segmentation with AC% greater than 90% in all cases.

ISF is a simple but effective filter. This filter is easy to implement and apply, because its only parameters are the radii of the two medians filters, which may be chosen according to the size of the imaged structures and the speckle texture. In addition, because it is able to reduce speckle without blurring object edges, the resulting filtered-image is a good candidate for anisotropic diffusion.

ISF is a speckle filter that takes into consideration the physics related to the interference of ultrasound waves, which is the origin of speckle noise. Bright and dark pixels are related to constructive and destructive interference, respectively. ISF eliminates the dark pixels by comparing and choosing the brighter pixel between its original value and the local median. Because the filter suppresses destructive interference and preserves the constructive one, the local region mean of an image tends to increase which could be a disadvantage when looking at purely quantitative evaluation metrics, such as RMSE and SSIM. However, in this study, ISFAD presented the best results for both metrics. ISFAD also resulted in the best standard deviation and USDSAI. The increase in the mean intensity value by itself is not important, because the contrast of the image can simply be readjusted.

Usually, the performance of a segmentation technique is highly dependent on its numerous parameters. The choice of such parameters is not always intuitive, and in some cases, they are selected empirically, which may be a time consuming process. In this study, two different sets of parameters were utilized for each of three segmentation techniques, and ISFAD resulted in accuracy greater than 90% for structures with well-defined borders, which suggests that ISFAD

improves the ability to segment some structures, in turn reducing the dependence of the segmentation algorithms on their parameter sets.

More results can be found in [3].

Speckle noise filtering will be used to post-process US images in order to improve computational procedures such as optical flow, 2D Block Matching and elastography estimation.

4.3 SNOL performance

As explained in Section 3.4.5, the Speckle Noise Oscillation Length analyses the image that will be filtered in order to compute automatically the window size of the filter.

In order to perform a quantitative validation, we created computationally simulated ultrasound images using Field II. Using the same model, 100 different scatterers distributions were generated, obtaining, hence, a stack with 100 different speckled images.

In order to obtain the gold-standard image I_{GS} , for each pixel position (x, y) , the median value considering the 100 values in the same (x, y) position of the stack was obtained.

$$I_{GS}(x, y) = \text{median}\{I_1(x, y), I_2(x, y), \dots, I_{100}(x, y)\}$$

As evaluation metrics, we used mean and standard deviation in a homogeneous region (Figure 19), and RMSE and SSIM in the whole filtered image.

To validate the proposed method, median filter was chosen because it has good performance on speckle noise [3] and has only one variable, the window size. We utilized circular window instead of square ones to avoid ‘blocking’ artifacts. Therefore, SNOL provides the radius of the window of median filter.

Then, the proposed method and the median filter were applied recursively to the 100 images. In the example, SNOL was applied on the speckle image I_{Sp} and it estimated a 3-pixel radius. Then, median filter was applied, generating I_{F3} . Next, SNOL was applied on I_{F3} , suggesting a 5-pixel radius. Median filter was applied on I_{F3} , generating $I_{F3,F5}$. Analogously, the median filter with radius 9, 17 and 32 were progressively applied, generating $I_{F3,F5,F9}$, $I_{F3,F5,F9,F17}$ and $I_{F3,F5,F9,F17,F32}$, respectively.

Next, the mean, standard deviation, RMSE and SSIM were calculated and the mean values of the results are shown in Table 7. Since the standard deviations for RMSE and SSIM were small, they were omitted for clarity. The results of the direct application of median filter with radius 3, 5, 9, 17 and 32 pixels - I_{F3} , I_{F5} , I_{F9} , I_{F17} and I_{F32} , respectively - were also included for comparison.

Figure 19 illustrates the filtered images for visual assessment.

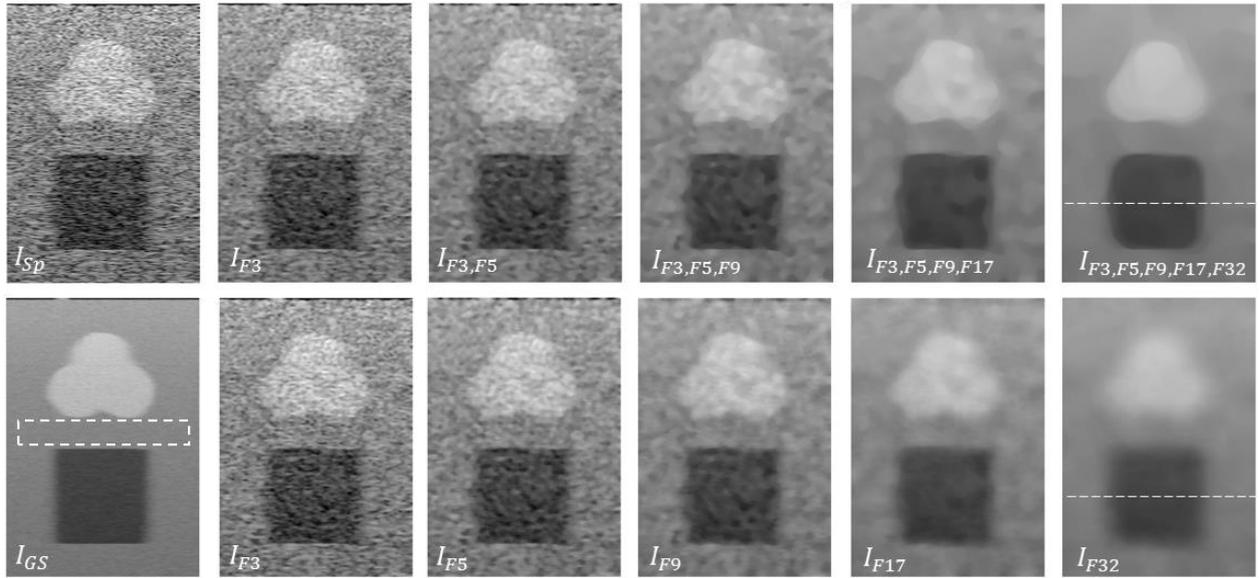


Figure 19: Top, speckled image and progressively filtered images. Bottom, gold-standard image and direct application of median filter. The dashed rectangle in I_{GS} depicts the homogeneous region used to calculate the mean and standard deviation. The dashed horizontal lines in $I_{F3,F5,F9,F17,F32}$ and I_{F32} indicate the position of the profile shown in Figure 20.

Table 7: Filters numeric performances

Image	Mean	Standard Deviation	RMSE	SSIM
I_{GS}	132.7	3.92	0.00	1.00
I_{Sp}	127.6	32.10	0.61	0.13
I_{F3}	130.8	18.01	0.54	0.43
$I_{F3,F5}$	131.5	12.3	0.47	0.66
$I_{F3,F5,F9}$	131.8	8.00	0.39	0.79
$I_{F3,F5,F9,F17}$	131.9	5.43	0.31	0.83
$I_{F3,F5,F9,F17,F32}$	131.4	3.93	0.28	0.83
I_{F3}	130.8	18.02	0.54	0.43
I_{F5}	131.3	12.98	0.47	0.62
I_{F9}	131.6	8.43	0.39	0.76
I_{F17}	131.8	5.60	0.31	0.82
I_{F32}	131.4	6.66	0.28	0.82

As can be seen in Table 7, the progressive filtering has a slightly better performance than applying the filter with large window size at once. However, in Figure 19, it is possible to notice the benefits of progressive filtering using SNOL, where the speckle texture was removed and the edges were preserved. In Figure 19, I_{F17} and I_{F32} presented blurred object borders, while in $I_{F3,F5,F9,F17}$ and $I_{F3,F5,F9,F17,F32}$ the edges are still sharp. Figure 20 illustrates the variation of the gray level in a horizontal line (shown in Figure 19) of $I_{F3,F5,F9,F17,F32}$ and I_{F32} . It is possible to see that the level variation in $I_{F3,F5,F9,F17,F32}$ (red) is more abrupt, so the border detection is easier. On the other hand, I_{F32} (blue) provides a smooth change of level, making the border position more difficult to recognize.

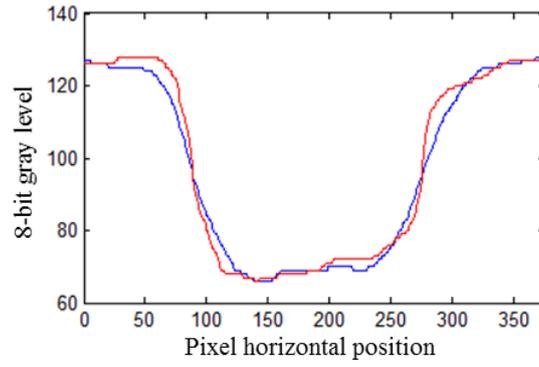


Figure 20: Horizontal profile of level variation of $I_{F3,F5,F9,F17,F32}$ (red) and I_{F32} (blue).

5. Creation of US Images Using 3D phantoms

In this section, we present the procedure to obtain a sequence of transcutaneous US images using 3D phantoms (Step1 of Figure 21). Step 2 of Figure 21 is explained in Section 6.

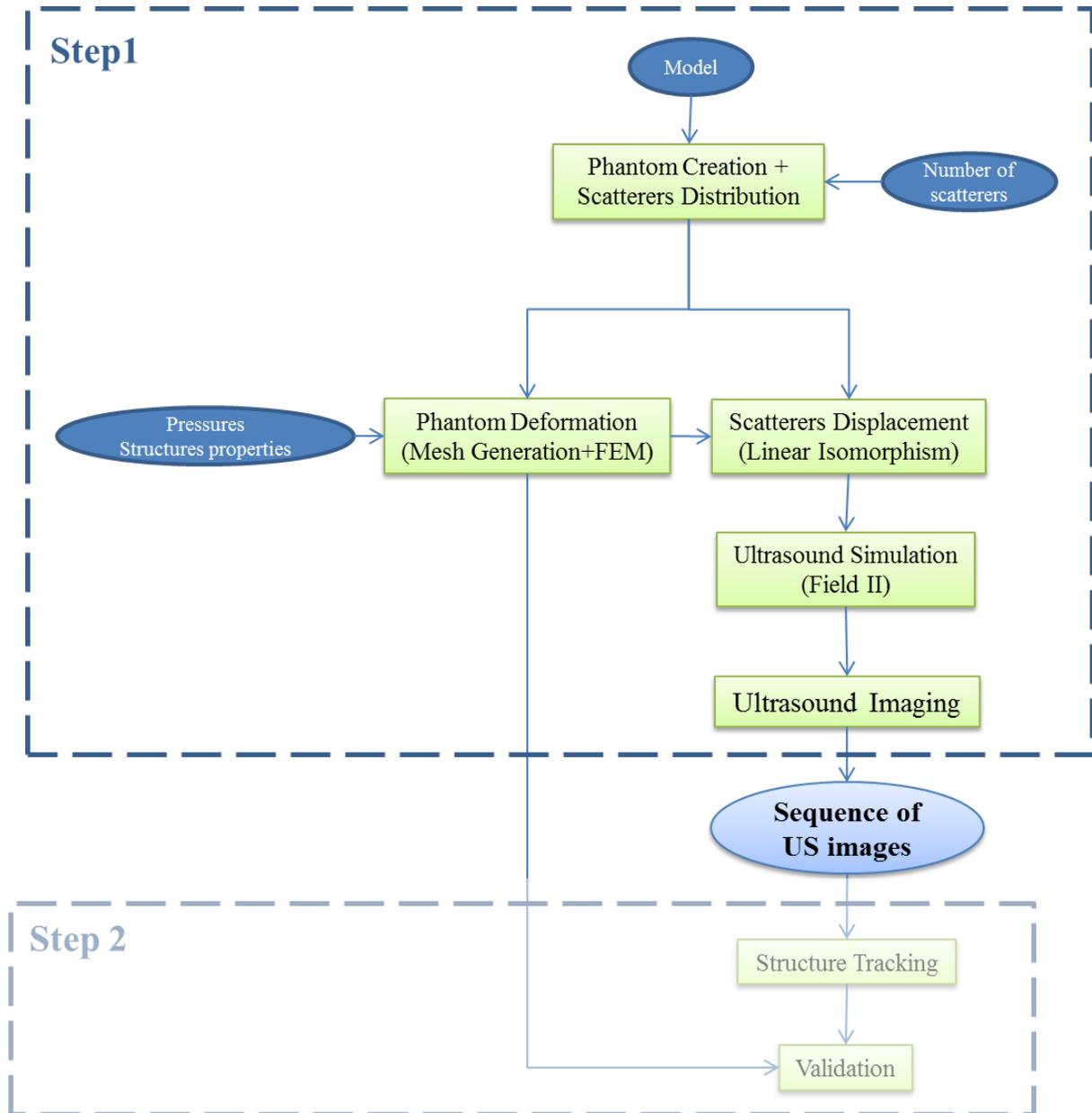


Figure 21: Block Diagram. Step 1 depicts the procedure to obtain the simulated ultrasound images sequence. Step 2 illustrates the sequence for the movement estimation

In this work, we have created a software based on Matlab® that creates ultrasound numeric phantoms generating simulated ultrasound images (Step 1 of Figure 21). The software is called UltraSSim (UltraSound Simulator) and is available online [35] free from charges for purposes of education and research.

The software framework involves models based on related studies; realistic structures deformation using FEM [36]-[39], and speckle modeling based on Field II [22], [23]. The deformation is carried out according to the pressure applied by the probe and the corresponding Young Modulus of each structure. The speckle modeling is performed taking into account the US transducer, scatterers density, acoustic impedance and attenuation of each structure.

UltraSSim framework methodology is divided into five major blocks, **Phantom Creation**, **Phantom Deformation**, **Scatterers Movement**, **US Simulation** and **US Imaging**. The first and last blocks require user interaction; all other blocks are computed automatically. First, in **Phantom Creation** step, a volumetric phantom is created by the user; the model would correspond to the tissue that will be imaged, containing any type or combination of structures or lesions and with no pressure on it. The user must also inform the phantom structures acoustic and elastic properties, the applied pressures and the transducer properties. Second, during the **Phantom Deformation** stage, the existing borders and regions in the model are automatically identified. Then, a finite element mesh is generated using iso2mesh and deformed through FEM according to the elastic properties and pressures inserted by the user in the first step. Third, in the **Scatterers Movement** block, point scatterers are distributed throughout the phantom and then displaced according to the mesh deformation through linear isomorphism. Fourth, in the **US Simulation** stage, Field II is applied to simulate the interaction between the ultrasound beam and the point scatterers; as output a set of vectors are computed, which correspond to the data acquired during the US examination. Finally, during the **US Imaging** block, the vectors are used to generate US B-mode images, which the user is able to manipulate.

The Equipment and Software used for the simulated representations was a personal computer with an Intel® Core™ i7, 2.67 GHz, 8 GB of RAM, Windows 7 and MATLAB® (2011a – 64bits) (MathWorks, Inc., Natick, MA, USA) with iso2mesh [40],[41], and Field II.

The evaluation was performed by analyzing two different aspects, the accuracy of the lesion deformation and the final image texture. In order to do that, we created US phantoms models in different conditions and characteristics, and computing and comparing our outcomes with published studies.

5.1 User Inputs

5.1.1 Phantom Creation

The inserted phantom may be any three-dimensional matrix where each value of the voxels represents one structure.

Although the algorithm works with volumetric phantom (therefore 3-dimensional matrix), Figure 22 illustrates a planar phantom with two-dimensional matrix for clarity.

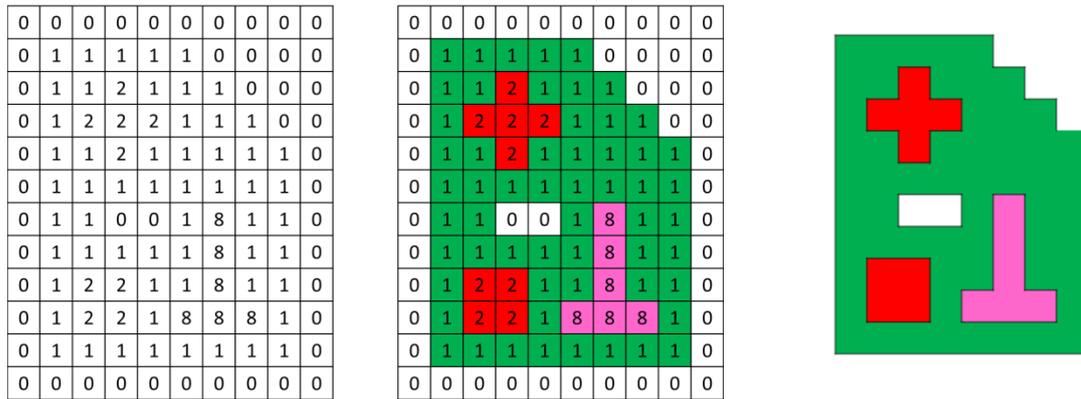


Figure 22: Planar representation of the phantom creation using a 2D matrix.

All voxels with value 0 are considered background. Any other value stands for a tissue. If there are zero-valued voxels in the middle of the phantom, a hole is represented. If there are two disjoint groups of voxels with the same value, the algorithm recognizes them as being made of the same material.

5.1.2 Defining Simulation Parameters

In this section, we present all the parameters the user needs to introduce in order to begin the simulation. The user enters the values through three GUIs (Graphical User Interface): phantom properties, applied forces and transducer properties.

5.1.2.1 Phantom Properties

For each structure, the user is requested to enter four properties: Elasticity (Young Modulus), Scattering Amplitude, Density of Scatterers and Attenuation (Figure 23).

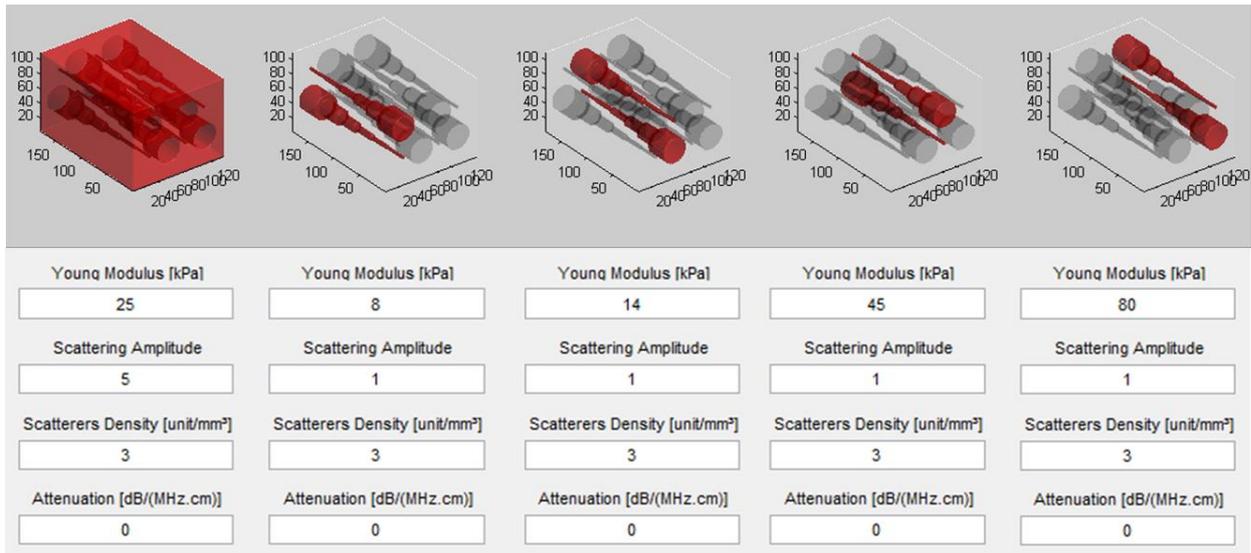


Figure 23: GUI for entering the tissue properties

Elasticity, or Young Modulus, is a measure of stiffness of the elastic structure. It is defined as the ratio of stress over strain. In this work, its unit is kPa and it is utilized to compute the stiffness matrix used in Finite Elements Method (FEM), a procedure used to calculate deformations that will be explained on Section 5.2.2.

Scattering Amplitude defines the intensity with which the particles scatter the US wave. In other word, it determines the influence of the scatterers over the point spread function during the US simulation.

The Scatterers Density, scattering particles per volume unit, depends on the tissue material and on the ultrasound beam central frequency. Unfortunately, we could not find a set of coherent and reliable publications related to the density of scattering particles within a tissue. Jensen [23] published three examples distributing the particles uniformly throughout the whole phantom: 3.33, 2.22 and 6.67 particles/mm³ on the cyst, fetus and kidney examples, respectively, with US central frequency 3.5, 5 and 7 MHz, respectively. On the other hand, [42] simulated intravascular ultrasound utilizing different density of the scattering particles in the same phantom according to the region: 8000, 800 and 100 particles/mm³ for the fibrous cap, vessel wall and lipid, respectively, with US central frequency 20 MHz.

Attenuation is the reduction in amplitude of the ultrasound signal as it travels through the medium. The attenuation coefficient, α , unit is dB/(MHz.cm). In order to mitigate the effects of attenuation over the image, the signal amplitude is compensated according to depth. In clinical cases it is known as TCG (time-gain control), which will be explained on Section 5.6.3.

If Help button is pressed, a window (Figure 24) informs the properties of some common tissues.

Tissue	Young Modulus [kPa]	Attenuation [dB/(cm.MHz)]
Blood	2,200,000	0.2
Bone Cortical	18,600,000	6.9
Bone Trabecular	10,400,000	9.94
Breast	2.2	0.75
Cardiac	100	0.52
Fat	.017	0.48
Liver	.64	0.5
Muscle	24-400	1.09
Tendon	1,800,000	4.7

Figure 24: Informative table of common tissue properties.

5.1.2.2 Applied Forces

In clinical examination, the physician, besides analysing the image, may produce some pressure with the probe during the ultrasound imaging in order to observe the tissues deformation. This algorithm was created so as to simulate such pressure. The user enters the size of the probe surface, where pressure is applied and the intensity of pressure (Figure 25).

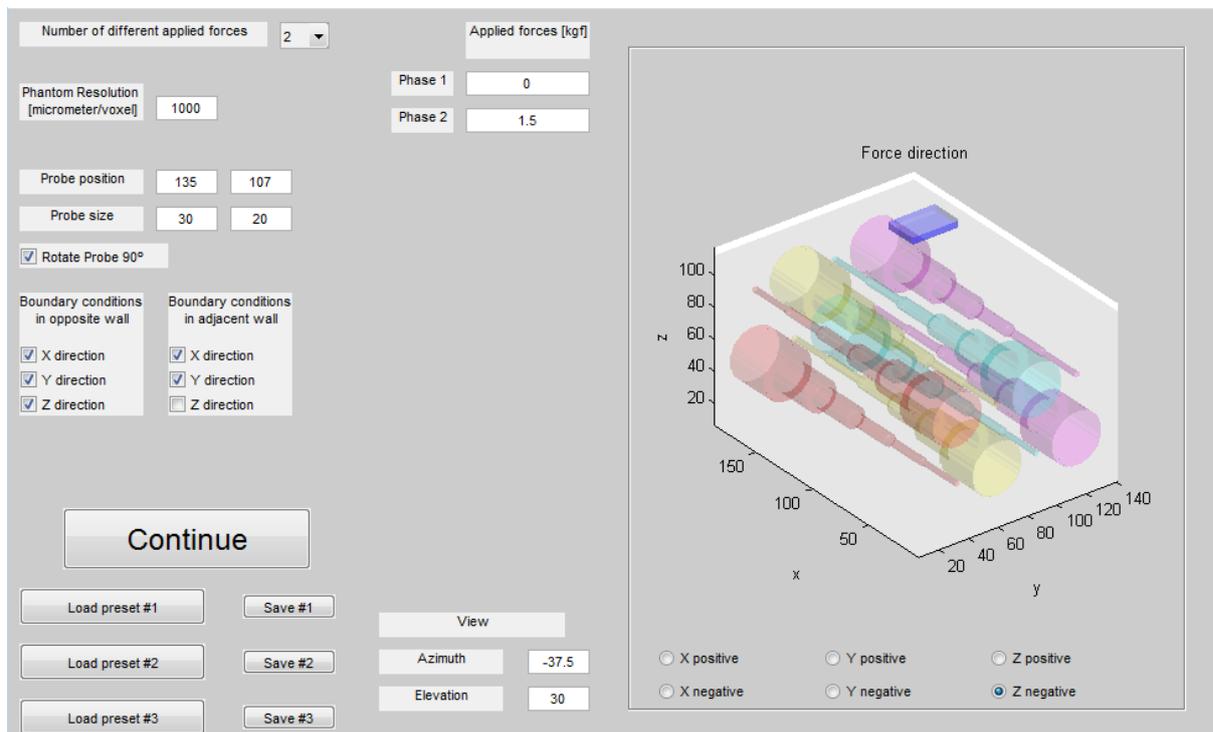


Figure 25: GUI for entering probe position and applied forces.

Force Direction defines in which of the six faces of the phantom the pressure is applied: X, Y or Z in *positive or negative* direction.

Number of Different Applied Forces defines how many images will be generated, each one with a different deformation due to the Applied Forces.

Applied Forces are used to compute the simulated pressure as explained above. The input unit kgf was chosen, rather than N or kPa , in order to be more intuitive to the user. The pressure is internally calculated by the algorithm as follows.

$$\text{Eq. 50} \quad P = \frac{f * 9.81}{A_{probe}}$$

where P is the pressure in Pa , f is the applied force in kgf , which is converted to N using the acceleration of gravity 9.81 m/s^2 , A_{probe} is the probe surface area in m^2 .

Probe Position is the position of the center of the probe surface and its unit is pixel. Probe Size defines the width and height of the probe surface in pixels. The user has also the option of inverting width and height by checking the box Rotate Probe 90° , this is important once the transducer elements array will be distributed along the width direction.

The blue square in Figure 25 illustrates the probe surface.

Finally, the user must enter the Phantom Resolution that is used to convert all the measures of distance and size from voxel to meters.

5.1.2.3 Ultrasound Transducer Properties

The user must define the ultrasound transducer parameters in order to simulate the hardware involved in ultrasound imaging (Figure 26)

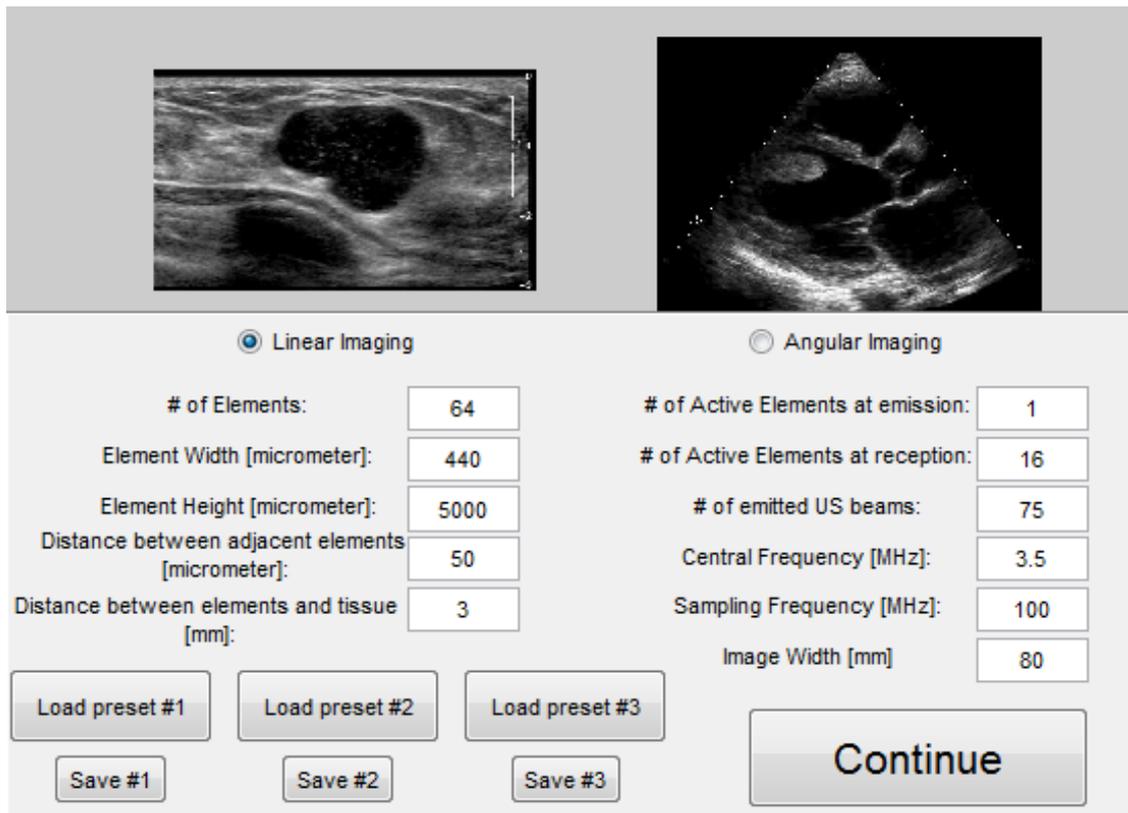


Figure 26: GUI for entering the transducer properties. The signal ‘#’ represents the word number.

Number of elements denotes the number of physical elements that compound the transducer linear array. Element Width and Element Height represents the width and height of each rectangular element, respectively. Distance Between two elements (kerf) is the gap size between two adjacent elements (Figure 27).

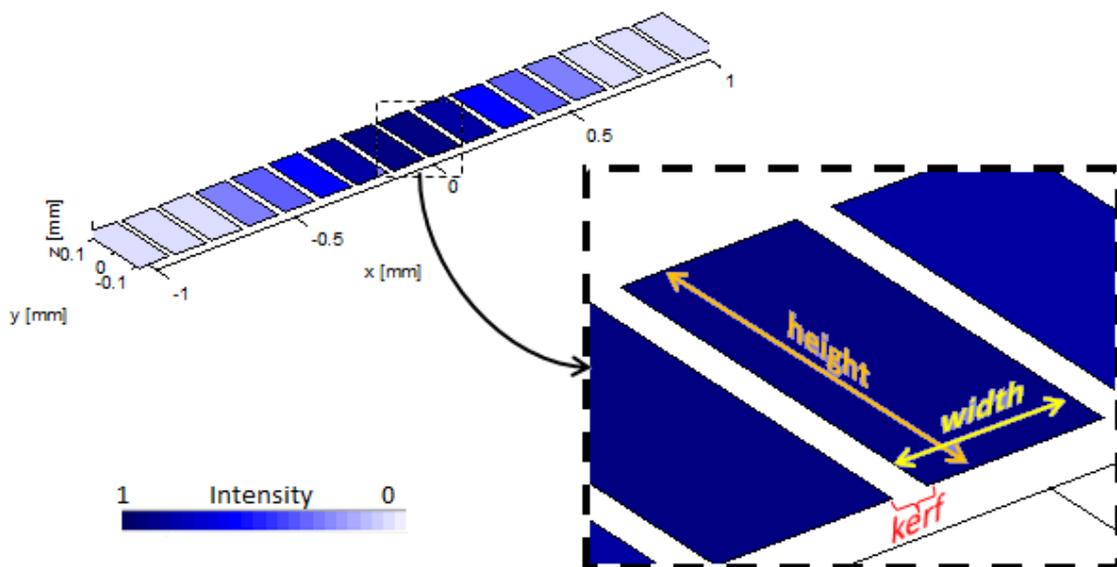


Figure 27: Transducer representation with 16 elements, element width = $110\mu\text{m}$, element height = $250\mu\text{m}$, element kerf = $22\mu\text{m}$. The blue level denotes the apodization intensity.

Number of Active Elements at Emission (N_e) and Number of Active Elements at Reception (N_r) define the apodization during emission and reception, respectively. The apodization consists on activating the entered number of transducers and weighting their power through Hanning Function (Figure 27).

Number of Emitted Beams defines the number of ultrasound waves that will be emitted to form an image. If Linear Imaging is selected the beams will be perpendicular and uniformly distributed along the array axial direction (Figure 28, left). On the other hand, if Angular Imaging is selected, the beams will propagate from the array center equiangularly distributed from $-\theta/2$ to $\theta/2$ (Figure 28, right), where θ is the angle entered by the user in Sector Angle (Figure 26). In both cases, the beam trajectory is computed using phased array.

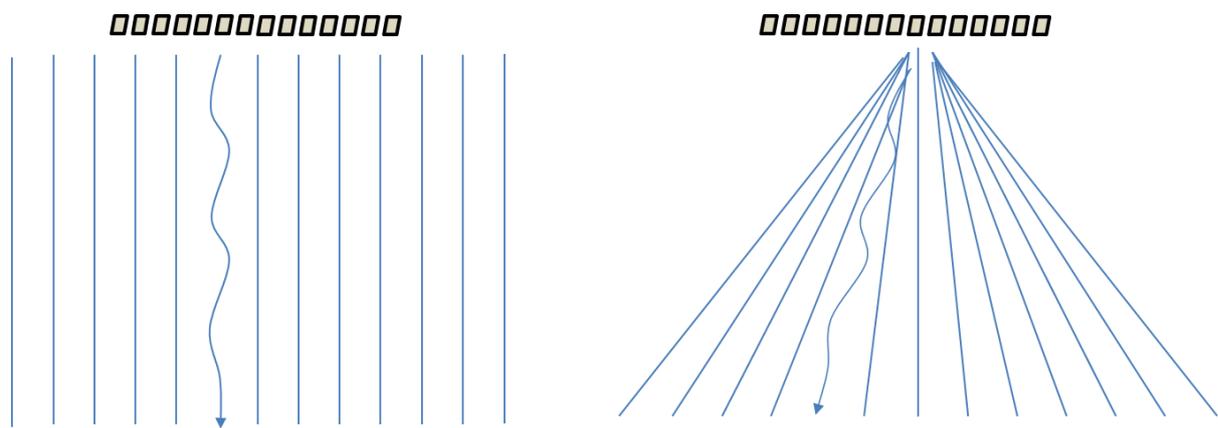


Figure 28: Left, linear imaging, the ultrasound beams are parallel. Right, angular imaging, the ultrasound beams are radial.

Considering the example on Figure 26, the transducer array width is 31 mm and the image width is 80 mm. The linear imaging is performed using 75 US beams that are uniformly distributed along the 80 mm. In UltraSSim, the transducer array is fixed and centered in relation to the image. Therefore, in this case, since the image width is larger than the transducer array width, there is emission and reception of US beams out of the array surface. Such beams are obtained through synthetic aperture using the nearest N_e elements for emission and the nearest N_r elements for reception. For the beams located within the array width, as the position of the beams sequentially moves from one end to the other, the apodization moves accordingly in order to recruit the set of elements that are closest to the beam. In other words, the 75 US beams ‘sweep’ the 80 mm from one end to the other and each US beam is built through synthetic aperture using the nearest N_e and N_r elements for emission and reception, respectively. Figure 29 depicts a simplified illustration of four examples of US beams in different positions with the corresponding recruited elements.

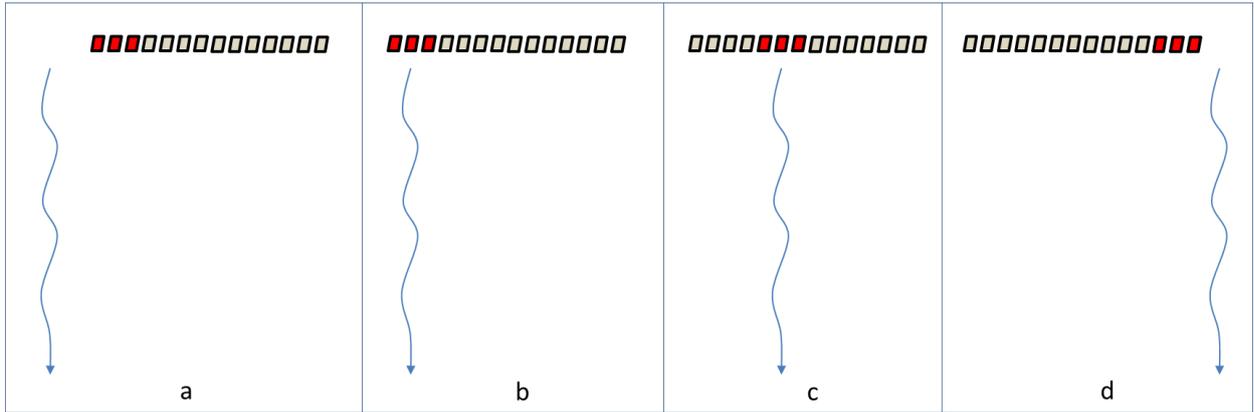


Figure 29: Examples of the synthetic aperture of US beams in four different positions. Elements in red represents the active elements.

Central Frequency defines the frequency of the emitted ultrasound waves and Distance between Elements and Tissue denotes the gap between the transducer elements and the phantom.

5.2 Phantom Deformation

In order to perform the phantom deformation, a mesh is generated and Finite Elements Method is applied, as explained in the next sections.

5.2.1 Mesh Generation

The phantom volume is divided into a set of tetrahedrons utilizing iso2mesh, developed by [40] and available at [41]. Iso2mesh is a free Matlab/octave-based mesh generation and processing toolbox. It can create 3D tetrahedral finite element (FE) mesh from volumes through Delaunay tetrahelization. Each tetrahedron is assigned a structure number, which is the same number as the structure in the volumetric phantom.

Before starting the iso2mesh, UltraSSim calculates the maximum number of nodes. The number of nodes is limited by the usage of RAM memory during the construction of the stiffness matrix in the Finite Elements Method, which will be explained on Section 5.2.2. The maximum number of nodes is calculated as follows

$$\text{Eq. 51} \quad (3 * N_{nodes})^2 * 8 \leq RAM_{avail} * 0.8$$

where N_{nodes} is the number of nodes and it is multiplied by 3 because each node has 3

coordinates $\{x, y, z\}$. The product is squared as the stiffness matrix is square and the result is

multiplied by eight because each matrix element has *double* precision. RAM_{avail} is the available RAM memory and it is multiplied by 0.8 in order to utilize only 80% of it.

Usually, the fewer nodes, the tetrahelization gets rougher (Figure 31). On the other hand, if the memory usage exceeds the available memory, FEM calculation may be very time-consuming. Hence, there is a trade-off and the user is asked for a decision (Figure 30).

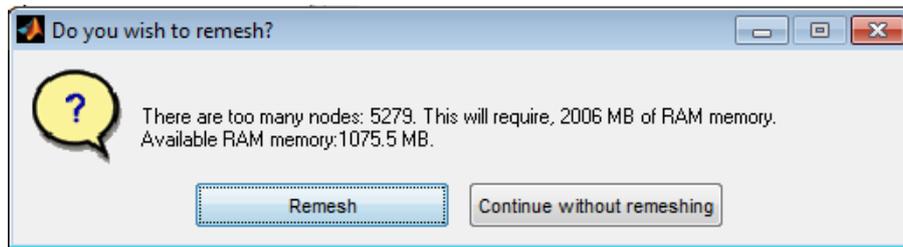


Figure 30: If the algorithm will consume more memory than available, the user is asked if he/she wishes to remesh to a lower resolution.

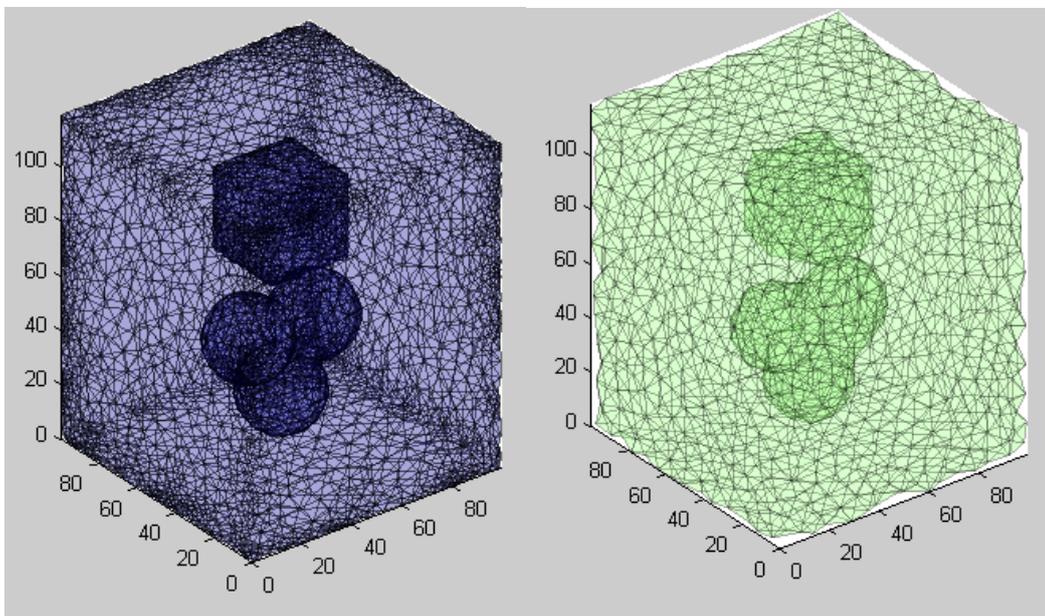


Figure 31: In blue, the original mesh. In green, the mesh after remeshing to a lower number of nodes.

The only *iso2mesh* parameter that UltraSSim changes is *distbound*, which defines the maximum distance between the center of the element surface bounding circle and center of the element bounding sphere. Its initial value is 1. If it exceeds condition Eq. 51 and the user decides to remesh according to the available memory, *distbound* is progressively incremented by 0.05 until Eq. 51 is satisfied.

All other parameters are used as default by *iso2mesh*.

5.2.2 Deformation through finite elements

In the first GUI, the user enters the Young Modulus of each structure. In the second GUI, the user inserts the applied pressures, transducer width and height. With that information, the algorithm is able to calculate the phantom static deformation, through Finite Elements Method (FEM) applied to the generated mesh. The application of FEM is explained in details in Section 11.1.

The software recognizes the mesh triangular faces located at the interface between the transducer elements and the phantom elements. For each face, the area is calculated as follows.

$$\text{Eq. 52} \quad A_{face} = \frac{1}{2} \cdot \text{abs} \left(\begin{vmatrix} x_A - x_C & x_B - x_C & 1 \\ y_A - y_C & y_B - y_C & 1 \\ z_A - z_C & z_B - z_C & 1 \end{vmatrix} \right)$$

where A,B,C and D are the tetrahedral nodes (vertices), $\{x_i, y_i, z_i\}$ are the nodes coordinates.

Then, the intensity of the force applied perpendicularly on the given face is

$$\text{Eq. 53} \quad F_{face} = P \cdot A_{face}$$

Since the face is compounded by three nodes, each node has its force incremented.

$$\text{Eq. 54} \quad F_{node} = F_{node} + F_{face} / 3$$

The incrementing is divided by 3 because it is applied three times, one for each node of the triangle.

In the beginning, each node has null force and the nodes that belong to the faces located at the interface transducer-tissue have their value incremented (Figure 32).

The boundary conditions consist of the set of nodes located at the phantom distal wall in relation to the transducer face. Those nodes have no degree of freedom so they cannot move in any direction.

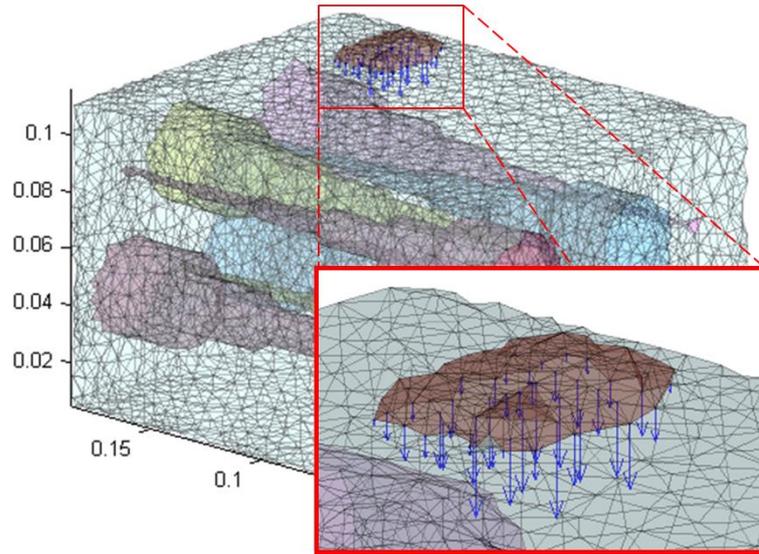


Figure 32: Phantom mesh with the probe surface represented in blue. In the zoomed detail, it is possible to observe the force vectors applied at the probe surface nodes. The scale is in meters.

5.3 Scatterers Movement

5.3.1 Scatterers Distribution

As explained above, a tetrahedral mesh is generated. Each tetrahedral element has four vertices, thus it is determined by four nodes $\{A, B, C, D\}$.

$$\text{Eq. 55} \quad \begin{cases} P_A = \{x_A, y_A, z_A\} \\ P_B = \{x_B, y_B, z_B\} \\ P_C = \{x_C, y_C, z_C\} \\ P_D = \{x_D, y_D, z_D\} \end{cases}$$

where P_i is a node position and $\{x_i, y_i, z_i\}$ are the nodes coordinates.

Then, the element volume is computed.

$$\text{Eq. 56} \quad V_{elem} = \frac{1}{6} \cdot \text{abs} \left(\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_A & x_B & x_C & x_D \\ y_A & y_B & y_C & y_D \\ z_A & z_B & z_C & z_D \end{vmatrix} \right)$$

where A,B,C and D are the tetrahedral nodes (vertices), $\{x_i, y_i, z_i\}$ are the nodes coordinates, $| \quad |$ is the matrix determinant operator and $\text{abs}(\quad)$ is the absolute value.

After computing the volume, the algorithm retrieves the element scatterer density, which was introduced by the user through GUIs (Figure 23). Then, the number of scatterers inside the element is calculated as follows

$$\text{Eq. 57} \quad N_{elem} = V_{elem} \cdot d_i$$

where V_{elem} is the element volume and d_i is the scatterers density of the tissue I , to which the element belong.

Next, N_{elem} scatterers are uniformly distributed inside the element volume. Utilizing the nodes position $\{P_A, P_B, P_C, P_D\}$, each scatterer has its position $P = \{x, y, z\}$ determined by four constants $\{c_A, c_B, c_C, c_D\}$.

$$\text{Eq. 58} \quad P = c_A \cdot P_A + c_B \cdot P_B + c_C \cdot P_C + c_D \cdot P_D \rightarrow \begin{cases} x = c_A \cdot x_A + c_B \cdot x_B + c_C \cdot x_C + c_D \cdot x_D \\ y = c_A \cdot y_A + c_B \cdot y_B + c_C \cdot y_C + c_D \cdot y_D \\ z = c_A \cdot z_A + c_B \cdot z_B + c_C \cdot z_C + c_D \cdot z_D \end{cases}$$

To guarantee that the scatterer is inside the tetrahedron, the following rules must be satisfied.

$$\text{Eq. 59} \quad \begin{cases} 0 \leq c_A, c_B, c_C, c_D \leq 1 \\ c_A + c_B + c_C + c_D = 1 \end{cases}$$

Each scatterer is assigned a scattering amplitude A based on the region to which it belongs.

$$\text{Eq. 60} \quad A = g(1) * 10^{\mathcal{S}}$$

where $g(1)$ is a number from a Gaussian distribution with zero mean and small standard deviation, e.g. 1. \mathcal{S} is the tissue global scattering amplitude entered by the user (Figure 23).

The choice of \mathcal{S} and scatterers density by the user according to the desired RF_{signal} mean amplitude (or image mean brightness) will be explained in Section 5.8.2.

5.3.2 Scatterers Displacement

As seen on Section 5.2.2, the tetrahedral mesh is subjected to compression, and then deformed. In order to calculate the new position of each scatterer after deformation, we applied linear isomorphism [43].

Let $ABCD$ be a tetrahedron, P_A, P_B, P_C, P_D be its respective vertices positions and P be a point inside of $ABCD$ (Figure 33 Left). If the tetrahedron is deformed in such a way that the vertices are

moved to the positions $P'_A P'_B P'_C P'_D$, then the inside point is moved to the position P' , assuming that the transformation is linear.

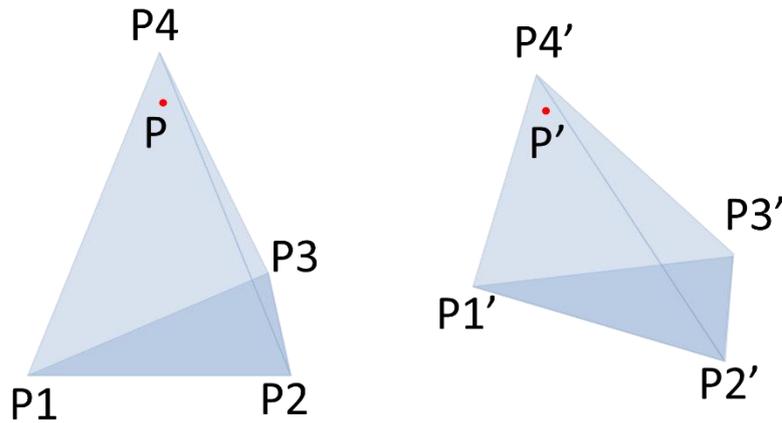


Figure 33: Left, tetrahedron before deformation. Right, tetrahedron after deformation.

In order, to find the new position of the scatterer after deformation, we repeated equation

$$\text{Eq. 61} \quad P = c_A \cdot P_A + c_B \cdot P_B + c_C \cdot P_C + c_D \cdot P_D \rightarrow \begin{cases} x = c_A \cdot x_A + c_B \cdot x_B + c_C \cdot x_C + c_D \cdot x_D \\ y = c_A \cdot y_A + c_B \cdot y_B + c_C \cdot y_C + c_D \cdot y_D \\ z = c_A \cdot z_A + c_B \cdot z_B + c_C \cdot z_C + c_D \cdot z_D \end{cases}$$

Since the nodes position after deformation $P'_A P'_B P'_C P'_D$ are known through FEM (Section 5.2.2). The new position of the each scatterer may be determined using its constants c_A, c_B, c_C, c_D as follows

$$\text{Eq. 62} \quad \begin{cases} P'_A = \{x'_A, y'_A, z'_A\} \\ P'_B = \{x'_B, y'_B, z'_B\} \\ P'_C = \{x'_C, y'_C, z'_C\} \\ P'_D = \{x'_D, y'_D, z'_D\} \\ P' = \{x', y', z'\} \end{cases}$$

$$\text{Eq. 63} \quad P' = c_A \cdot P'_A + c_B \cdot P'_B + c_C \cdot P'_C + c_D \cdot P'_D \rightarrow \begin{cases} x' = c_A \cdot x'_A + c_B \cdot x'_B + c_C \cdot x'_C + c_D \cdot x'_D \\ y' = c_A \cdot y'_A + c_B \cdot y'_B + c_C \cdot y'_C + c_D \cdot y'_D \\ z' = c_A \cdot z'_A + c_B \cdot z'_B + c_C \cdot z'_C + c_D \cdot z'_D \end{cases}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x'_A & x'_B & x'_C & x'_D \\ y'_A & y'_B & y'_C & y'_D \\ z'_A & z'_B & z'_C & z'_D \end{bmatrix} \cdot \begin{bmatrix} c_A \\ c_B \\ c_C \\ c_D \end{bmatrix}$$

5.4 Ultrasound Simulation - RF data acquisition

The ultrasound simulation is performed using Field II, which is able to simulate the interaction of the ultrasound pressure field with the scatterers.

The excitation of the transducer consists on a two-cycle sinusoidal wave (Figure 34a) and its impulse response is given by a two-cycle sinusoidal wave weighted with a Hanning window (Figure 34c). The transducer central frequency is set by the transducer through the input GUIs (Section 5.1.2.3) and the sampling frequency is preset to 100MHz, as suggested by [22]-[23].

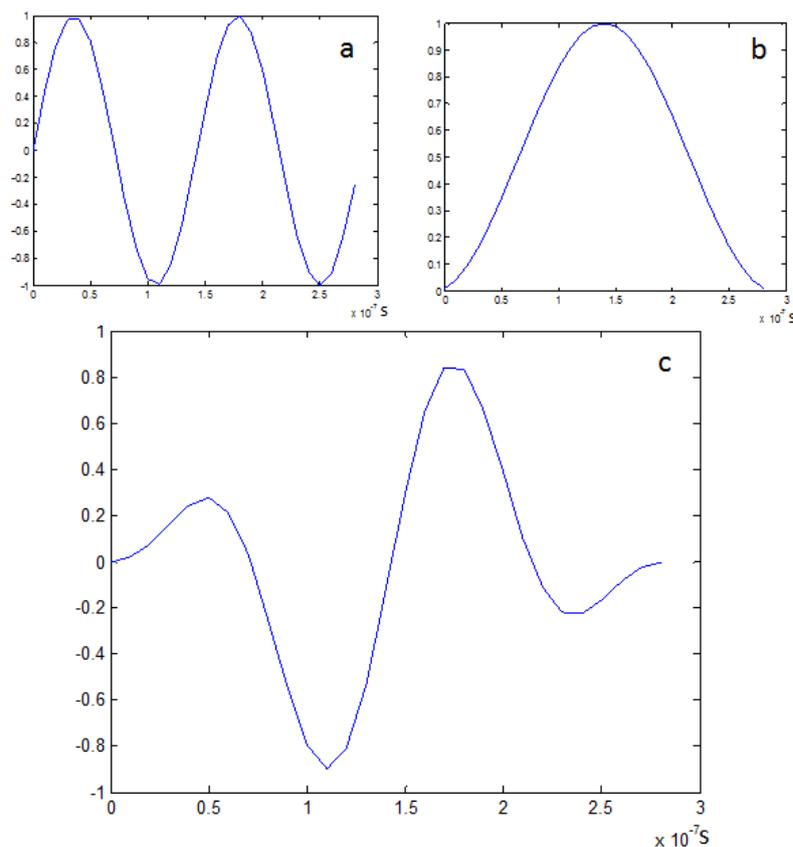


Figure 34: a) Two-cycle sinusoidal wave. b) Hanning Function and c) two-cycle sinusoidal wave weighted with a Hanning function

The receiving transducer is the same as the emitter and the impulse response for the reception aperture is equal to the impulse response for the transmission aperture (Figure 34c).

The emitted beams are focused at a distance corresponding to 85% of the phantom axial limits. The dynamic reception focus is set from the phantom proximal wall to 1.5 times the distal wall divided by 7 equal distances.

In order to optimize timing performance, the algorithm takes into consideration only the scatterers that are located within a slice of the phantom with 15 mm-thickness (Figure 35).

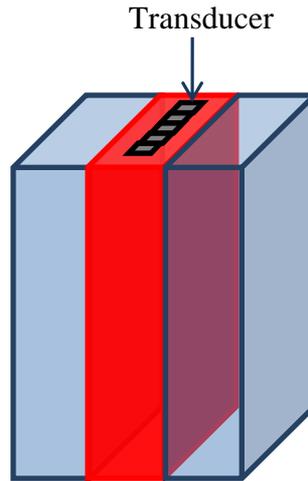


Figure 35: Phantom representation where the red fraction is the region considered by the transducer

Field II outputs RF_{line} data, which is a vector that corresponds to the sampled wave amplitude received by the transducer. In total, Field II generates one folder for each deformation (Number of different applied forces in Figure 25) and each folder contains a number of RF_{line} defined in Number of emitted US beams in Figure 26.

5.5 Ultrasound Imaging – Attenuation

The attenuation is calculated considering each *step distance* of the US beam during its travel. *Step distance* is the distance from one acquired sample to the next and is calculated as follows.

$$\text{Eq. 64} \quad d_s = \frac{c}{f_s}$$

where c is US speed in the medium and f_s is the sampling frequency.

Therefore, each RF_{line} sample will have a corresponding attenuation factor, which is calculated as follows.

$$\text{Eq. 65} \quad att_i = att_{i-1} * 10^{-\alpha \cdot d_s \cdot f_0 / 20}$$

where i is the sample index, f_0 is the US central frequency and α is the attenuation coefficient inserted by the user in Figure 23.

Finally, the attenuated signal is obtained by multiplying each sample of the RF_{line} to its corresponding attenuation factor.

5.6 Ultrasound Imaging – B-mode image formation

In order to obtain the B-mode image, firstly, the signal envelope ($RF_{envelope}$) is calculated as follows

$$\text{Eq. 66} \quad RF_{envelope} = |RF_{line} + jH(RF_{line})|$$

where $H(\)$ is the Hilbert Transform, $| \ |$ is the absolute value and RF_{line} is the RF signal provided by Field II.

Next log-compression was performed

$$\text{Eq. 67} \quad RF_{logComp} = \log(RF_{envelope})$$

Finally, the resulting signal is downsampled with the following sampling factor.

$$\text{Eq. 68} \quad D = \frac{Res_{IVUS} * f_s}{c}$$

where Res_{IVUS} is the image resolution in $m/pixel$.

The downsampling was performed selecting only the first element of a group of D consecutive samples and disregarding the others (Figure 36).

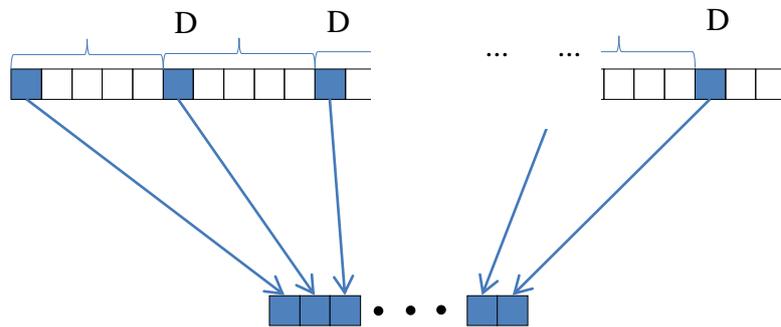


Figure 36: Illustration of the downsampling procedure.

5.6.1 Linear Imaging

In the linear imaging, the resulting lines are vertically distributed according to their position during acquisition (Figure 37 left). Then, linear (horizontal) interpolation is performed to fill the empty pixels (Figure 37 right).

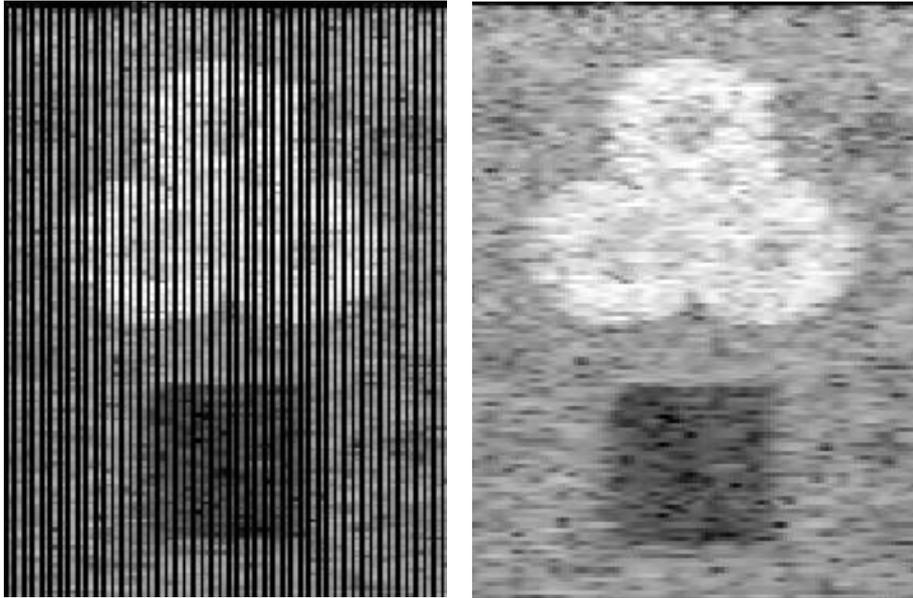


Figure 37: Left, distribution of the RF data according to their position during acquisition. Right, resulting image after interpolation.

5.6.2 Angular Imaging

The polar image is formed by the stack of the RF lines (Figure 38, left). The RF lines are radially and equiangularly distributed from $-\theta/2$ to $\theta/2$, where θ is the Sector Angle entered by the user (Figure 26).

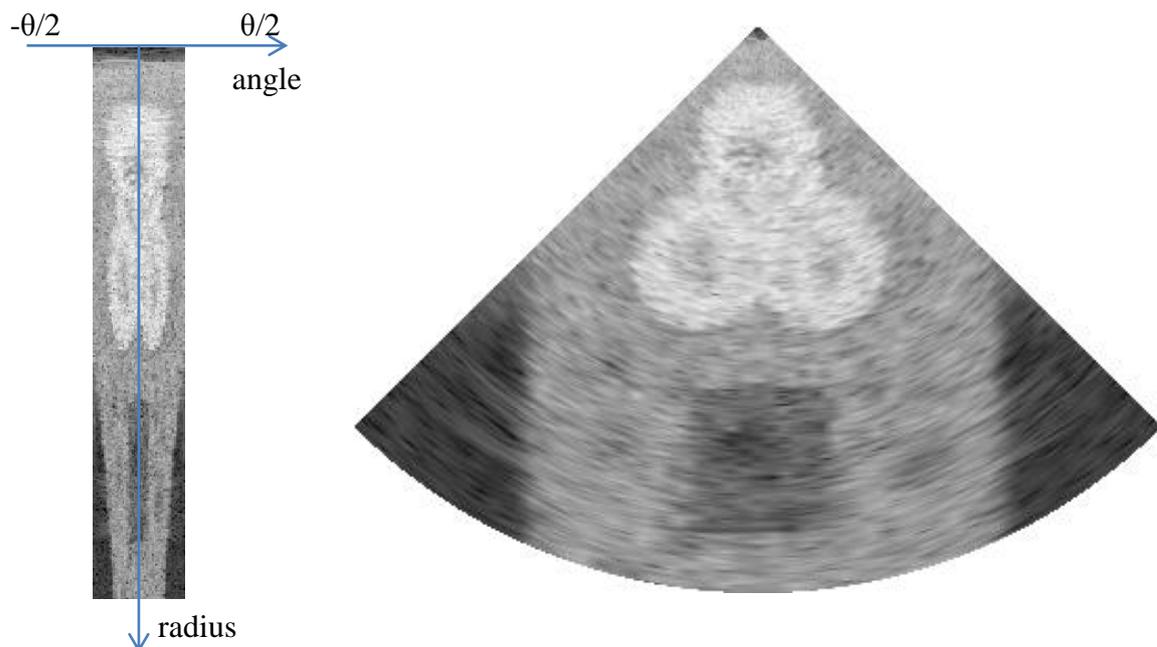


Figure 38: Right, polar image, RF data put side by side. Right, rectangular image, result after bilinear interpolation.

In order to fill the empty spaces in the image that were not covered by the radial signals, bilinear interpolation was performed (Figure 38 right). Let $p(x, y)$ be a pixel that was not covered

by the radially distributed RF lines. First we calculate its distance from the image center, R (radius), and its angulation, θ .

$$\text{Eq. 69} \quad R = \|(x, y) - (x_c, y_c)\|$$

$$\text{Eq. 70} \quad \alpha_{RF} = \theta \cdot \frac{n_{RF\text{lines}}}{\Theta}$$

where (x, y) are the coordinates of the pixel that was not covered by the radial signals, (x_c, y_c) are the coordinates of the source of the signal, $n_{RF\text{lines}}$ is the number of lines that compound the image and Θ is the image sector.

Then, the four pixels that are used to calculate $p(x, y)$ were retrieved from the polar image and bilinear interpolation was performed as follows.

$$\text{Eq. 71} \quad \begin{cases} p_1 = p_{Polar}(\lceil R \rceil, \lceil \theta_{RF} \rceil) \\ p_2 = p_{Polar}(\lceil R \rceil, \lfloor \theta_{RF} \rfloor) \\ p_3 = p_{Polar}(\lfloor R \rfloor, \lceil \theta_{RF} \rceil) \\ p_4 = p_{Polar}(\lfloor R \rfloor, \lfloor \theta_{RF} \rfloor) \end{cases}$$

$$\text{Eq. 72} \quad p(x, y) = (\lceil R \rceil - R) \cdot (\lceil \theta_{RF} \rceil - \theta_{RF}) \cdot p_4 + (\lceil R \rceil - R) \cdot (\theta_{RF} - \lfloor \theta_{RF} \rfloor) \cdot p_3 + \\ (R - \lfloor R \rfloor) \cdot (\lceil \theta_{RF} \rceil - \theta_{RF}) \cdot p_2 + (R - \lfloor R \rfloor) \cdot (\theta_{RF} - \lfloor \theta_{RF} \rfloor) \cdot p_1$$

where $\lceil x \rceil$ and $\lfloor x \rfloor$ represent the ceil and floor of x .

The resulting image simulates the B-mode ultrasonography as in clinical examination.

5.6.3 Visualizing the B-mode images

In order to facilitate the user interaction with the results, a GUI provides the images and controls so the user can change settings related to brightness and contrast (Figure 39). This way, he is able to emphasize the gray level of the image regions within his interests. He is also able to change the time-gain-control (TCG), image size and resolution.

The TCG is a tool used to mitigate the effects of attenuation during imaging. It gradually increases the signal intensity (pixel brightness) from the proximal part to the distal part of the image in relation to the transducer. In this work, the increment is linearly proportional to the distance from the transducer and its intensity is dynamically set by the user (Figure 39).

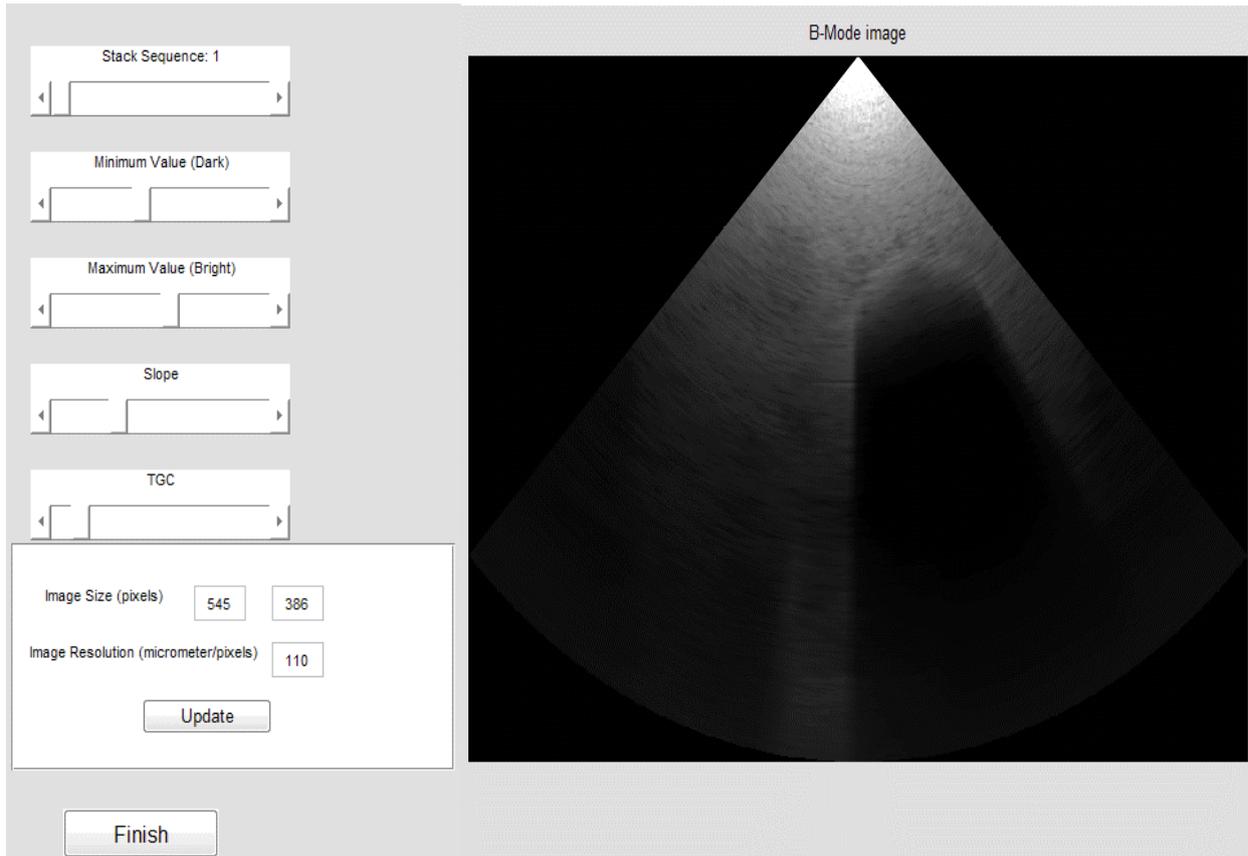


Figure 39: GUI for resulting images manipulation of a simulated nodule scan.

After the user decides the imaging parameters, he may export the stack of images to the Matlab Workspace. The output is a tridimensional matrix $\{x,y,N\}$, where x and y correspond to the orthogonal coordinates and N represent the number of images.

5.7 Deformation Maps

Utilizing the same image parameters (image size and resolution), the algorithm creates displacement maps and strain maps, which can be used as gold-standard in deformation related studies, such as speckle tracking, optical flow or elastography.

The displacement map is a 4D matrix. Three dimensions represent the phantom volume that was imaged, where each voxel holds the value of its displacement after deformation (Eq. 73). The fourth dimension represents the number of different deformations (due to different applied pressures). The outputs are u_x , u_y and u_z , which are the displacements in x , y and z direction respectively.

$$\begin{aligned}
 u_x(V_i) &= x_{post}(V_i) - x_{pre}(V_i) \\
 u_y(V_i) &= y_{post}(V_i) - y_{pre}(V_i) \\
 u_z(V_i) &= z_{post}(V_i) - z_{pre}(V_i)
 \end{aligned}$$

Eq. 73

where $x_{pre}(V_i)$ and $x_{post}(V_i)$ are the x-coordinate of the voxel i before and after deformation, respectively. $u_x(V_i)$ is the displacement in the x direction of the voxel i before deformation. The other variables are analogous.

The computation of *post* positions are performed in a similar way to the scatterers displacement, Section 5.3.2.

The algorithm also computes the strain map as follows.

$$\begin{aligned}
 \frac{\partial u_x(x,y,z)}{\partial x} &= \frac{u_x(x+1,y,z) - u_x(x-1,y,z)}{(x+1) - (x-1)} \\
 \frac{\partial u_x(x,y,z)}{\partial y} &= \frac{u_x(x,y+1,z) - u_x(x,y-1,z)}{2} \\
 \frac{\partial u_x(x,y,z)}{\partial z} &= \frac{u_x(x,y,z+1) - u_x(x,y,z-1)}{2} \\
 \frac{\partial u_y(x,y,z)}{\partial x} &= \frac{u_y(x+1,y,z) - u_y(x-1,y,z)}{2} \\
 \frac{\partial u_y(x,y,z)}{\partial y} &= \frac{u_y(x,y+1,z) - u_y(x,y-1,z)}{2} \\
 \frac{\partial u_y(x,y,z)}{\partial z} &= \frac{u_y(x,y,z+1) - u_y(x,y,z-1)}{2} \\
 \frac{\partial u_z(x,y,z)}{\partial x} &= \frac{u_z(x+1,y,z) - u_z(x-1,y,z)}{2} \\
 \frac{\partial u_z(x,y,z)}{\partial y} &= \frac{u_z(x,y+1,z) - u_z(x,y-1,z)}{2} \\
 \frac{\partial u_z(x,y,z)}{\partial z} &= \frac{u_z(x,y,z+1) - u_z(x,y,z-1)}{2}
 \end{aligned}$$

Eq. 74

5.8 Phantom Validation

5.8.1 FEM Accuracy

In order to validate the proposed algorithm, two different models were created (Figure 40). Let the model shown in (Figure 40, left) be the model A and the model in (Figure 40, right) be the model B. Model A and model B consisted of $12 \times 25 \times 32 \text{ mm}^3$ and $25 \times 25 \times 32 \text{ mm}^3$ phantoms, respectively. Each model was compounded by two layers with the same dimensions. In both models, the pressure was applied at the top (Figure 40) and the boundary conditions consisted of the nodes at the bottom of the phantom, which were unable to move in any direction.

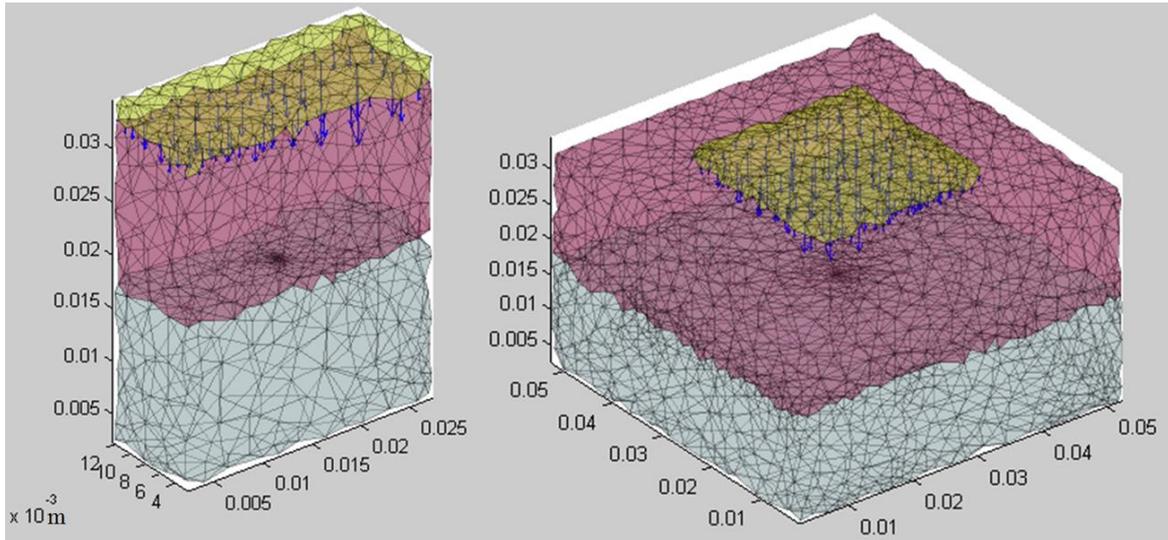


Figure 40: Two-layered models used to validate FEM accuracy. The yellow structure represents the transducer surface, in other words, the surface through which the forces are applied. The blue arrows represent the applied forces.

Several combinations of different stiffness were tested, so the results are presented in function of the modulus contrast (MC) between the two materials that compound the phantom.

$$\text{Eq. 75} \quad MC = \frac{E_1}{E_2}$$

where E_1 and E_2 are the Young Moduli of the top and bottom layers, respectively.

A wide range of MC values was tested: from 10^{-3} to 10^3 . In other words, the top layer stiffness ranged from 1000 times stiffer to 1000 softer in relation to the bottom layer.

The results are presented in strains contrast, SC.

$$\text{Eq. 76} \quad SC = \frac{\text{strain}_1}{\text{strain}_2}$$

where strain_1 and strain_2 are the mean strain of the top and bottom layers, respectively. The strains measures were taken from the axial line that contains the center of the transducer.

The results of both models are shown in Figure 41.

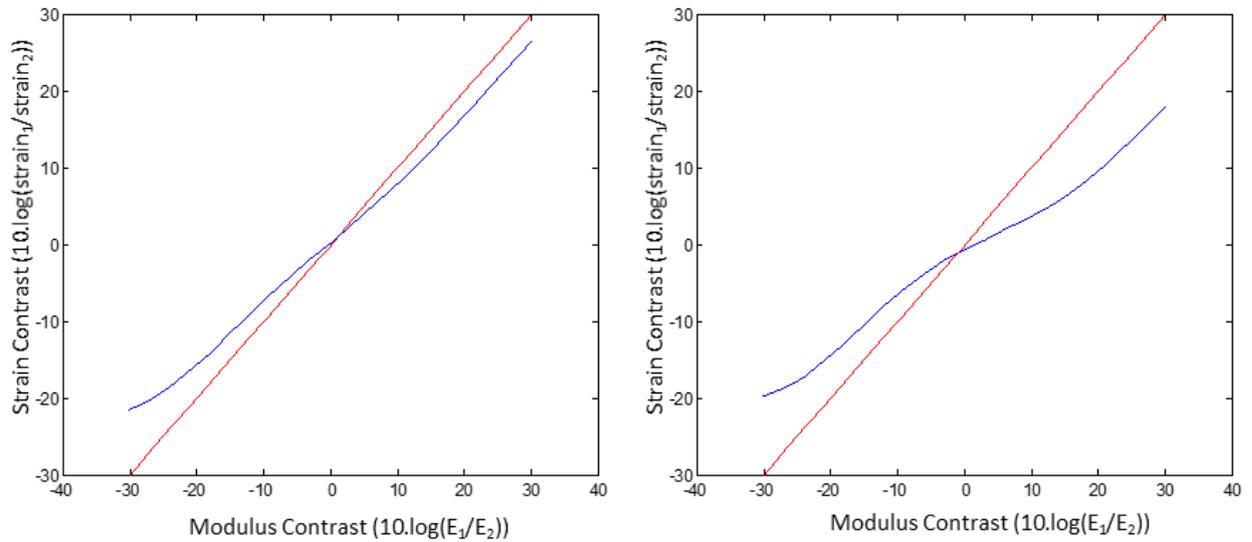


Figure 41: Strain contrast vs. modulus contrast for the model A (right) and model B (left). The blue lines show the strain contrast as estimated by 3-D FEM and the red line depicts the hypothetical ideal case when the strain contrast is equal to the modulus contrast.

The results obtained in this work are similar to those published in [44]. As expected, model A presented results closer to the ideal case, because in model B the deformation at the probe axis suffered a restoration influence from the tissue in the neighborhood that was not under direct pressure.

5.8.2 Tissue Contrast, Speckle Index and Histogram

The speckle noise was analyzed statistically, in order to verify its validity. Three parameters were taken into consideration: tissue contrast, speckle index and histogram.

To analyze the incorporated speckle noise, two different sets of simulation were performed. First, only the scatterers density was varied; second, the scattering amplitude was varied. All the simulations in this section were carried out using the same transducer properties (Table 8). The phantom was compounded by two regions, a reference region (region1) and a target region (region 2) (Figure 42).

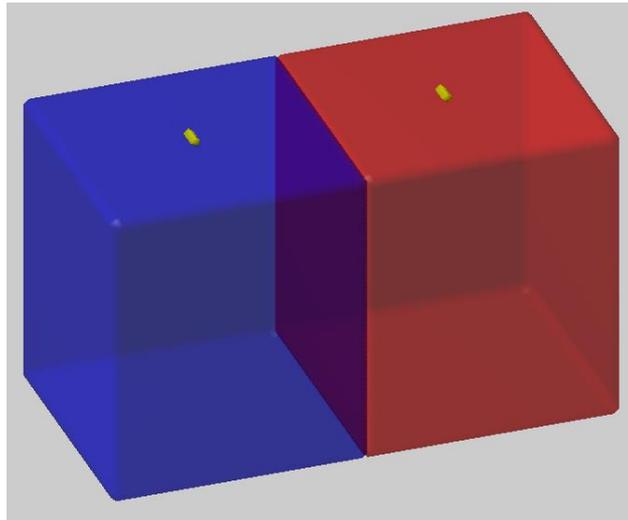


Figure 42: Phantom with two regions, where each color represents one region. The yellow structures represent the transducer elements. Each region has length=50mm, width=50mm and depth=60mm.

Table 8: Transducer properties

US central frequency	3.5 MHz
Number of elements	2
Number of emitted US beams	2 (one for each region)
Active elements during emission	1
Active elements during emission	1
Element width	440 μ m
Element Height	5000 μ m
Distance between two adjacent elements	22.5 mm

The tissue contrast was calculated using the mean ratio, MR, which refers to the ratio of the mean value of the RF signal envelope without log-compression (Eq. 66) from target region over the mean value of the RF signal envelope without log-compression from the reference region. Each RF signal contained 2500 samples. Finally, the speckle index, SI, of the RF signal envelope without log-compression of the target region was computed. The parameters were computed as follows.

$$\text{Eq. 77} \quad \left| \begin{array}{l} MR = \frac{\mu_{target}}{\mu_{reference}} \\ SI = \frac{\sigma_{target}}{\mu_{target}} \end{array} \right.$$

where μ and σ represents the mean and standard deviation, respectively; and the subscripts *reference* and *target* denotes the reference region and target regions, respectively.

Finally, the histogram was compared to the theoretical Rayleigh distribution utilizing Chi-Square Test. The mean value of the samples was taken into account to calculate the σ in the Rayleigh PDF.

$$\text{Eq. 78} \quad f(x, \sigma) = \frac{x}{\sigma^2} \cdot \exp\left(-x^2/2\sigma^2\right)$$

$$\text{Eq. 79} \quad \sigma = \mu \cdot \sqrt{\frac{2}{\pi}}$$

5.8.2.1 Influence of scatterers density

In order to verify the influence of the scatterers density over the speckle noise, 9 different values of scatterers density were tested in the target region, while the scatterers density of the reference region and the scattering amplitude of both regions were maintained constant (Table 9). Each one of the nine simulations was performed 20 times.

The results are shown in Table 9 and Figure 43.

Table 9: Results for different scatterers density

Sim.	Density of scatterers (scat/mm ³)		Scattering Amplitude		MR	SI
	Region 1	Region 2	Region 1	Region 2		
1	N1 = 5	N2 = 0.625	Z1 = 10	Z2 = 10	0.3533	0.8460
2	N1 = 5	N2 = 1.25	Z1 = 10	Z2 = 10	0.4973	0.8338
3	N1 = 5	N2 = 2.5	Z1 = 10	Z2 = 10	0.6982	0.8262
4	N1 = 5	N2 = 5	Z1 = 10	Z2 = 10	0.9801	0.8206
5	N1 = 5	N2 = 10	Z1 = 10	Z2 = 10	1.4506	0.8130
6	N1 = 5	N2 = 20	Z1 = 10	Z2 = 10	1.9299	0.8194
7	N1 = 5	N2 = 40	Z1 = 10	Z2 = 10	2.8275	0.8114
8	N1 = 5	N2 = 80	Z1 = 10	Z2 = 10	3.8555	0.8214
9	N1 = 5	N2 = 160	Z1 = 10	Z2 = 10	5.7907	0.8457

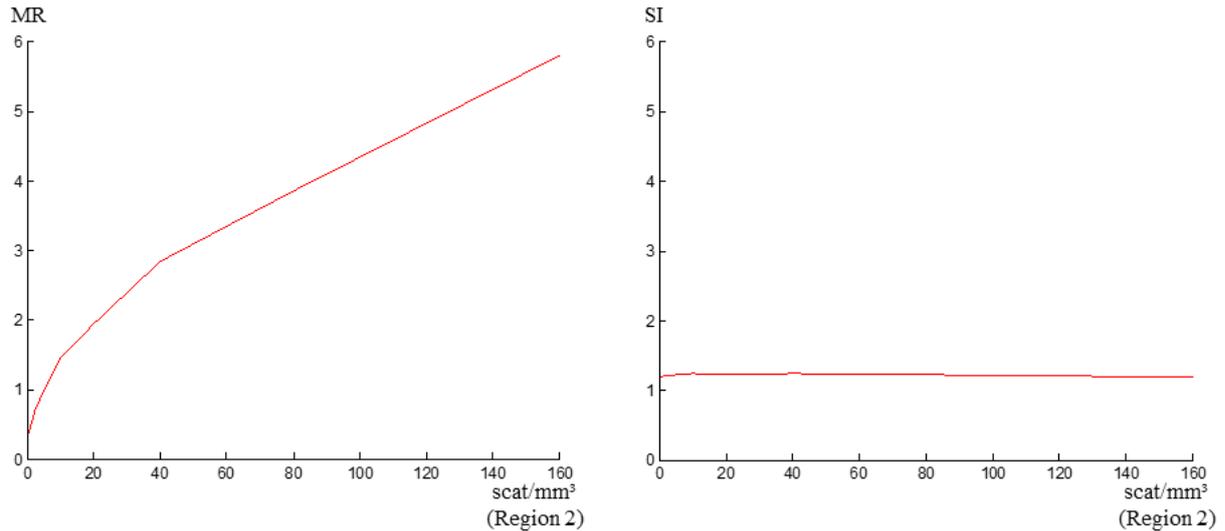


Figure 43: Statistics of the speckle noise in relation to the scatterers density of Region 2. Left, mean ratio; right, speckle index.

As can be seen in in Table 9 and Figure 43, if the scatterers density is doubled, the MR is multiplied by $\sqrt{2}$ while the SI remains constant and close to its theoretical value, $\sqrt{2/\pi}$.

5.8.2.2 Influence of scatterers amplitude

Since we wanted to analyze only the influence of the scattering amplitude without the interference of eventual differences in spatial distributions, a set of scatterers was uniformly distributed inside the volume in region 1 and then another set of scatterers was distributed in region 2 mirrored to the region 1 relative to interface between the two regions. Then, the scattering amplitudes were assigned to each scatterer according to Table 10, where nine different values of scatterers amplitude were tested in the target region, while the scatterers density of the reference region and the scattering amplitude of both regions were maintained constant.

The above-mentioned procedure was performed 20 times for each one of the nine simulations. The results are shown in Table 10 and Figure 44.

Table 10: Results for different scattering amplitude

Sim.	Density of scatterers (scat/mm ³)		Scattering Amplitude		MR	SI
	Region 1	Region 2	Region 1	Region 2		
1	N1 = 5	N2 = 5	Z1 = 10	Z2 = 0.625	0.0625	0.8187
2	N1 = 5	N2 = 5	Z1 = 10	Z2 = 1.25	0.1250	0.8187
3	N1 = 5	N2 = 5	Z1 = 10	Z2 = 2.5	0.2500	0.8187
4	N1 = 5	N2 = 5	Z1 = 10	Z2 = 5	0.5000	0.8187
5	N1 = 5	N2 = 5	Z1 = 10	Z2 = 10	1.0000	0.8187
6	N1 = 5	N2 = 5	Z1 = 10	Z2 = 20	2.0000	0.8187
7	N1 = 5	N2 = 5	Z1 = 10	Z2 = 40	4.0000	0.8187
8	N1 = 5	N2 = 5	Z1 = 10	Z2 = 80	8.0000	0.8187
9	N1 = 5	N2 = 5	Z1 = 10	Z2 = 160	16.0000	0.8187

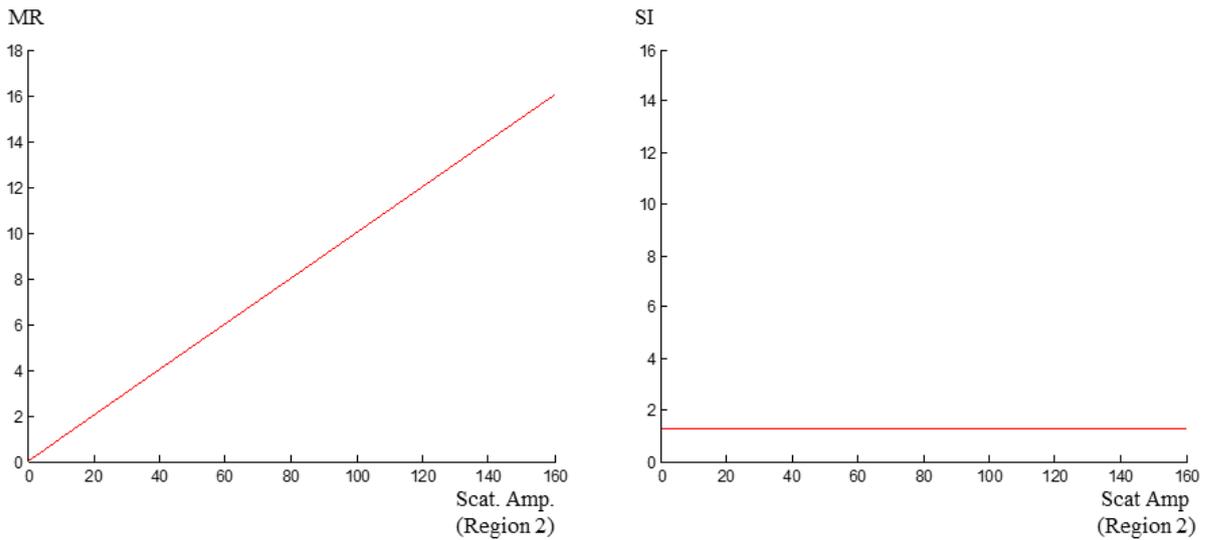


Figure 44: Statistics of the speckle noise in relation to the scattering amplitude of Region 2. Left, mean ratio; right, speckle index.

As can be seen in Table 10 and Figure 44, if the scattering amplitude is doubled, the MR also doubles while the SI remains constant and close to its theoretical value, $\sqrt{2/\pi}$.

5.8.2.3 Speckle Noise Histogram

Figure 45 shows the speckle noise histogram of one of the 360 (9x20x2) simulations and the probability density function of the Rayleigh distribution. The histogram consists of 20 bins equally divided. All the values of the histogram were divided by the integral of the graph in order to have integral equals to one and match the Rayleigh graph.

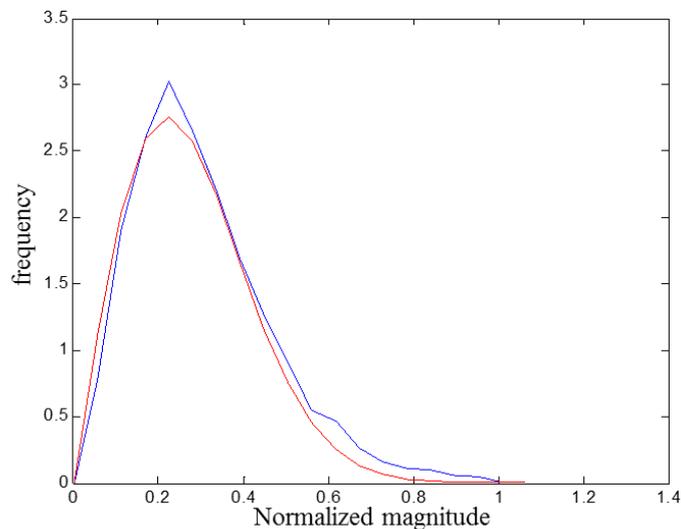


Figure 45: Histogram of the speckle noise distribution (blue) and Rayleigh Distribution (red) of one of the simulations.

Chi-square (χ^2) test was performed to evaluate the goodness of fit between the obtained histogram and the Rayleigh Distribution. Table 11 shows the mean χ^2 value for each simulation. The significance level was set to $\alpha = 0.05$ and 19 degrees of freedom was used.

Table 11: Goodness of fit between the histogram and Rayleigh distribution

Varying scatterers density		Varying scatterers amplitude	
Sim.	χ^2	Sim.	χ^2
1	0.0355	1	0.1305
2	0.0355	2	0.0239
3	0.0355	3	0.0067
4	0.0355	4	0.0304
5	0.0355	5	0.0062
6	0.0355	6	0.0075
7	0.0355	7	0.1062
8	0.0355	8	0.0090
9	0.0355	9	0.2022

Since the critical value is $\chi_{critical}^2 = 30.14$ (with 19 degrees of freedom and $\alpha = 0.05$), then, in all simulations, the hypothesis that the speckle histogram corresponds to a Rayleigh distribution cannot be rejected.

5.9 Intravascular Ultrasound Phantoms

In this work, we also created computational phantoms to perform the simulation of intravascular ultrasound. Two methods were developed: 2D IVUS simulation and 3D IVUS simulation. In both methods, the IVUS phantoms are also built using finite elements method to deform the tissue (Figure 46 b and c), linear isomorphism to determine the scatterers displacement and Field II to simulate the ultrasound acquisition (Figure 46 d). However, the catheter transducer is built to simulate the emission of US beams, radially, from the inside of the lumen (Figure 46 d and e).

Analogously to UltraSSim, we have created a software that creates IVUS numeric phantoms generating simulated ultrasound images. The software is called IVUSSim (IntraVascular UltraSound Simulation) [45] and is available online [35] free from charges for educational and research purposes. More details about the 2D IVUS simulation can be obtained in [45].

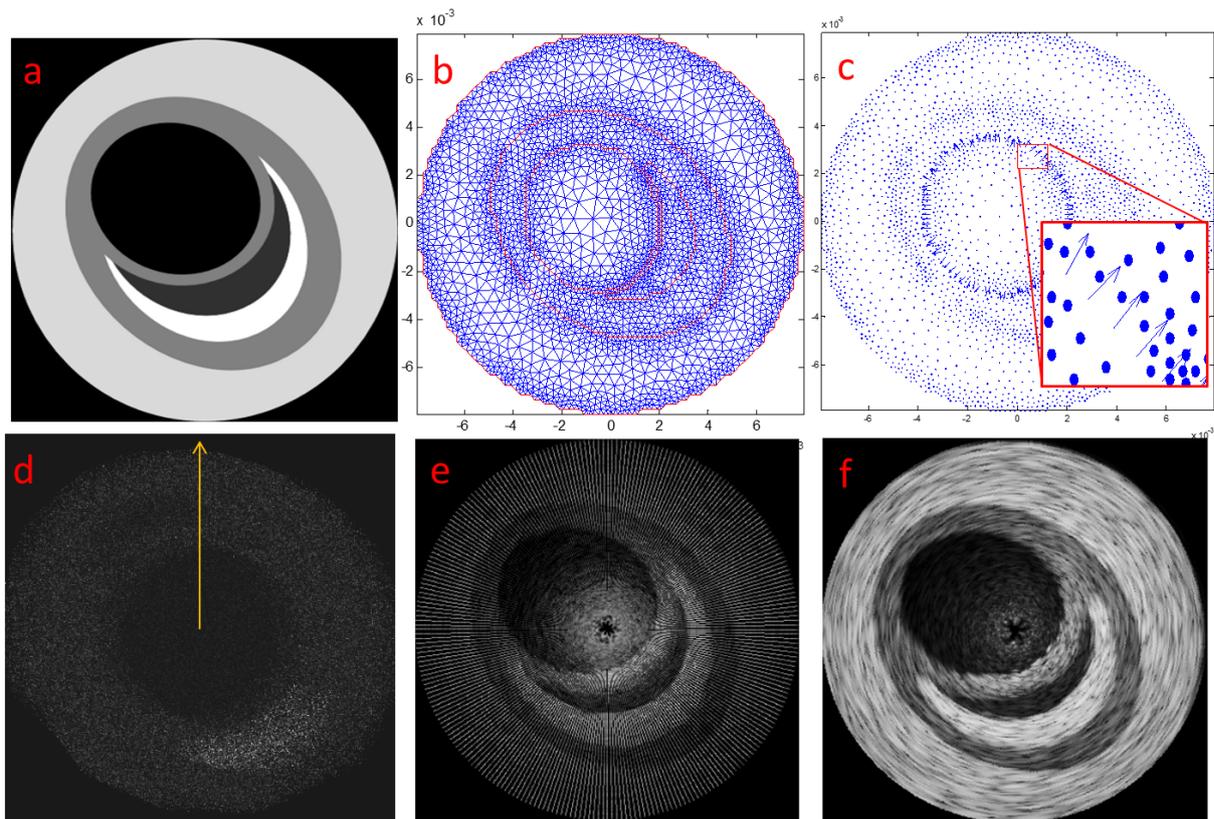


Figure 46: Simplified sequence of the construction of 2D intravascular phantom. a) planar model, b) triangular mesh, c) forces due to variation in internal pressure, d) scatterers and radial ultrasound beam (yellow arrow), e) distribution of the data acquired from 128 equiangularly distributed beams, f) simulated B-mode IVUS image.

In the 2D intravascular ultrasound simulation, the phantoms are constructed from a planar model of a blood vessel (Figure 46a). Next, a triangular mesh is obtained (Figure 46b) and finite elements method is performed to deform the mesh due to internal pressure variation, represented by perpendicular forces applied to the nodes at the lumen surface (Figure 46c). Then, Field II is used to synthesize a transducer and simulate the emission a set of US beams radially from inside the lumen (Figure 46 d and e). More comprehensive explanation about 2D intravascular ultrasound simulation can be found in Section 11.2.

The main difference of the 3D intravascular ultrasound models is that the model is a volumetric representation of a blood vessel (Figure 47). Therefore, since it is important to estimate the catheter position inside the blood vessel, we developed novel techniques to predict the catheter path inside a blood vessel using energy minimization.

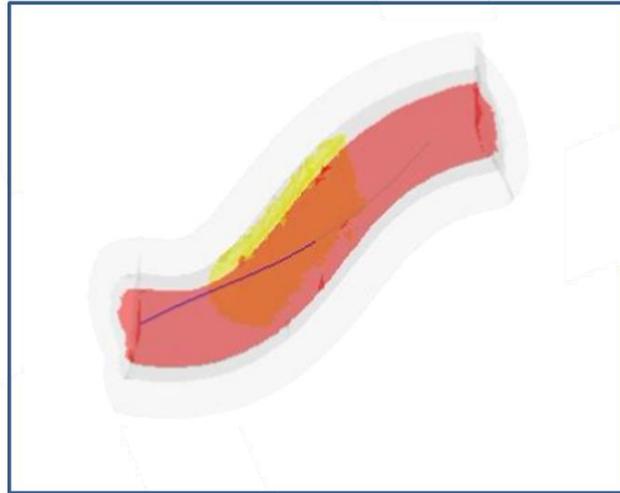


Figure 47: Volumetric model of a coronary with lesion (in yellow) and catheter (blue line inside the lumen)

In the following sections, we present the methods used to estimate the catheter path (Section 5.9.1), the pullback simulation and the arrangement of frames to simulate (Section 5.9.2) and dynamics of the acquisition in a living patient (Section 5.9.3). The 3D intravascular ultrasound simulation is further explained Section 11.3.

5.9.1 Catheter Path Simulation

It is important to estimate the path of the catheter inside a blood vessel because it determines the position of the cross section of the coronary that is being imaged. We have simulated the catheter path based on the fact that the catheter would rest at the minimum energy position. In this work, we have created two different methods to obtain the simulated catheter position inside the blood vessel. First, the method based on dynamic programming is introduced. Next, the algorithm based on active contours is presented. Finally, a comparison between the two methods with a physical phantom is performed.

5.9.1.1 Catheter path via Dynamic Programming (DP)

In order to calculate the minimum energy, we applied Dijkstra's shortest path algorithm [47]. The cost of the algorithm was based on the catheter curvature (Figure 48).

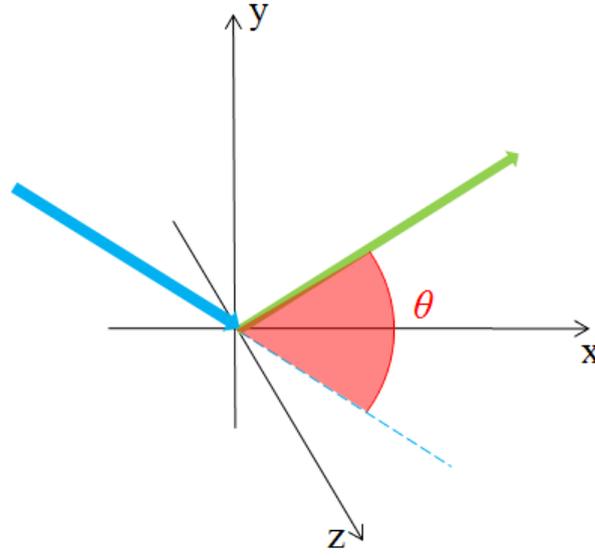


Figure 48: Angle θ between the direction of the predecessor (blue vector) and the next neighbor (green vector).

The cost function was calculated as follows:

$$\text{Eq. 80} \quad \text{cost} = \left(\frac{2 \cdot \text{asin}(\|\hat{t} - \hat{p}\|/2)}{\pi} \right)^2$$

where \hat{t} and \hat{p} are the normalized vectors of the directions of the next neighbor and predecessor, respectively. It is important to notice that

$$\text{Eq. 81} \quad \|\hat{t} - \hat{p}\|/2 = \sin(\theta/2)$$

where θ is the angle between the direction of the predecessor and the next neighbor. Hence, we can rewrite Eq. 80:

$$\text{Eq. 82} \quad \text{cost} = \left(\frac{\theta}{\pi} \right)^2$$

The cost ranges from 0 to 1 and it is minimum if \hat{t} and \hat{p} have the same direction and maximum if they have opposite direction. Figure 49 shows how the cost function varies according to θ .

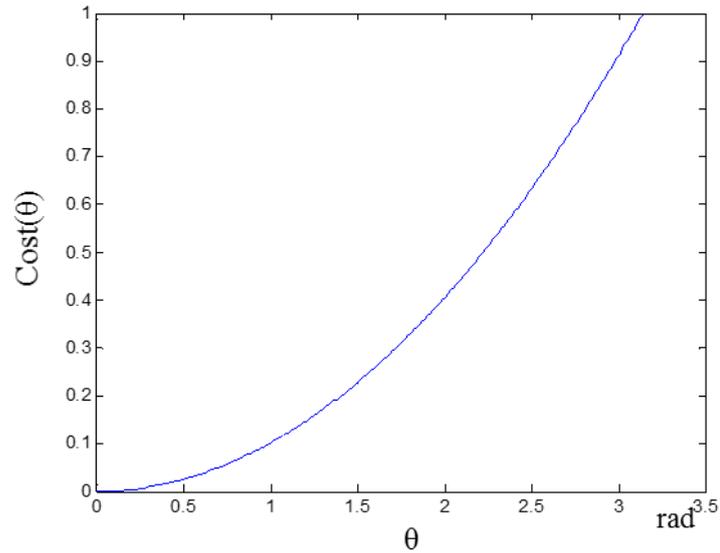


Figure 49: Plot of the function $cost(\theta)$.

It is important that both cost function and its derivative are crescent functions. This assures that the algorithm will distribute smaller angles as possible throughout the curve.

Figure 50 shows three examples of curvature distribution. In all cases, the angle between the first and last vectors is $\pi/2$. In the left, the curvature is made in one step of $\pi/2$; in the middle, there are two angles of $\pi/4$; and in the right, the curvature is distributed in three steps of $\pi/6$. The accumulated cost (acc.cost) of each situation is

$$\text{(left) } \text{acc.cost} = \left(\frac{\pi/2}{\pi}\right)^2 = 0.25$$

$$\text{(middle) } \text{acc.cost} = \left(\frac{\pi/4}{\pi}\right)^2 + \left(\frac{\pi/4}{\pi}\right)^2 = 0.125$$

$$\text{(right) } \text{acc.cost} = \left(\frac{\pi/6}{\pi}\right)^2 + \left(\frac{\pi/6}{\pi}\right)^2 + \left(\frac{\pi/6}{\pi}\right)^2 = 0.083$$

Therefore, the situation on the right presents the smallest cost (energy).

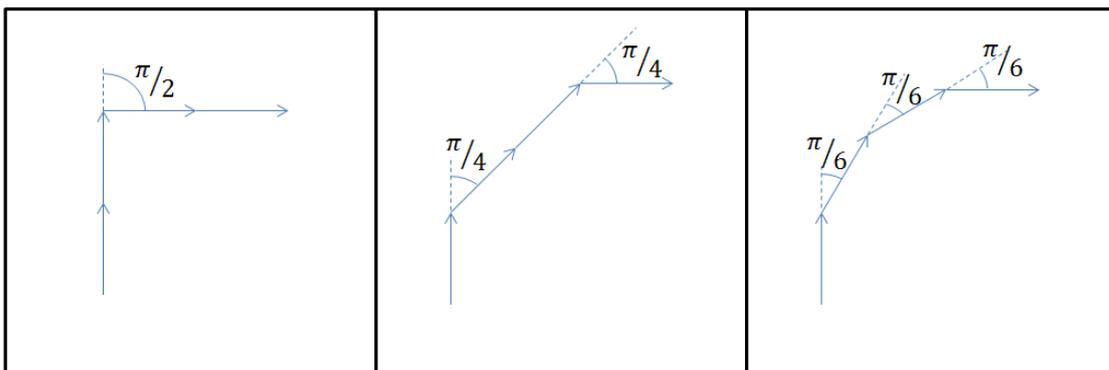


Figure 50: 3 different ways to distribute a curvature that sums 90° .

In this study, the user must enter the catheter initial position and direction, and the algorithm calculates and provides the cost, length and path related to each voxel of the phantom lumen.

We have tested two different implementations, DP1 and DP2:

DP1: each node is defined by a voxel. This implementation is faster and needs less memory to be performed, but it is less reliable.

DP2: each node is defined by a voxel and a predecessor voxel. Although this implementation is more reliable, it is impracticable in 3D phantoms due to the great amount of required memory and time.

5.9.1.2 Catheter path via Active Contours (AC)

The second method is based on Active Contours and utilizes equilibrium of forces to obtain the minimum energy position [48].

The numeric guidewire is a discretized model, which consists of consecutive segments united by vertices (Figure 51).

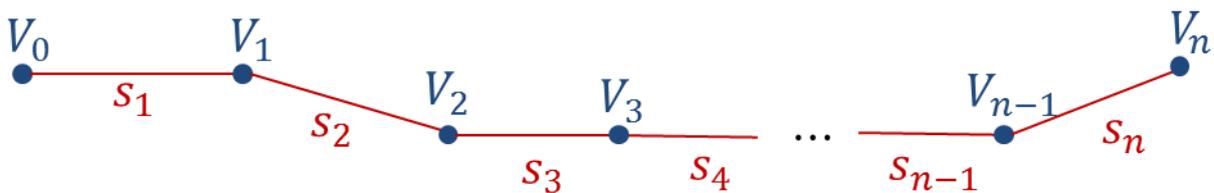


Figure 51: Representation of the numeric guidewire model.

The guidewire deformation is performed by forces that are applied iteratively at the vertices. The forces may be divided into three groups according to their nature: a) Longitudinal forces, which account for maintaining the guidewire length; b) Angular forces, that are responsible for unbending the guidewire, leading to the minimum energy position; and c) External forces, which are performed by the blood vessel that constrains the guidewire to its interior.

It is easy to notice that the longitudinal and angular forces are internal; therefore, reaction forces should also be applied to the guidewire. Detailed explanation about the forces will be presented in the following sections.

Although the algorithm was designed to work in 3D, the following illustrations were made planar for clarity.

A. LONGITUDINAL FORCE

The longitudinal force f^L works as an elastic force that tends to keep the original length of each segment.

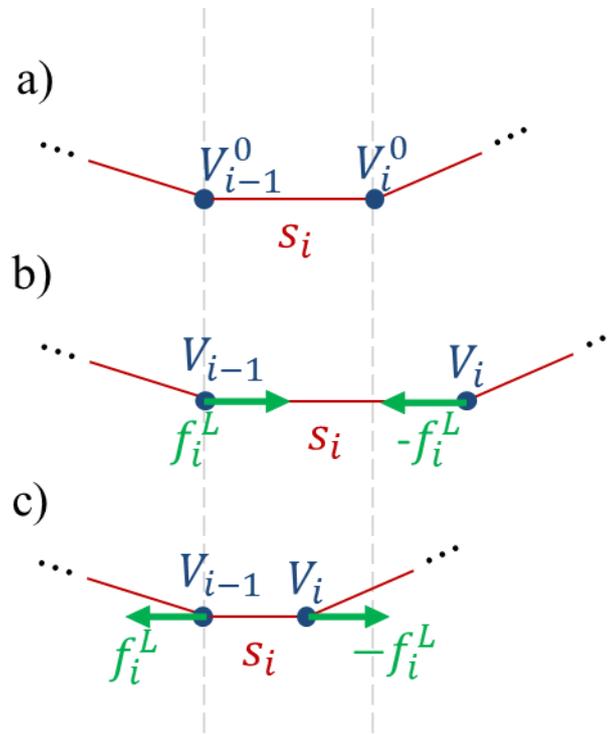


Figure 52: a) Illustration of s_i with its original length and its two consecutive vertices V_{i-1} and V_i , b) case in which the length of s_i is greater than the original length and f_i^L is applied in V_i towards V_{i-1} and vice versa, c) case in which the length of s_i is smaller than the original length and f_i^L is applied in V_i away from V_{i-1} and vice versa. The vertical dashed lines indicate V_i and V_{i-1} original position.

First, the initial length L^0 of each segment is registered. Then, for each iteration, the instantaneous length L is computed and the difference between L and L^0 governs the amplitude of \vec{f}_i^L , which has the same direction \hat{d}_i as the segment.

$$\text{Eq. 83} \quad L_i^0 = |V_i^0 - V_{i-1}^0|$$

$$\text{Eq. 84} \quad L_i = |V_i - V_{i-1}|$$

$$\text{Eq. 85} \quad \hat{d}_i = \frac{V_i - V_{i-1}}{|V_i - V_{i-1}|}$$

$$\text{Eq. 86} \quad \vec{f}_i^L = \hat{d}_i \cdot (L_i - L_i^0)$$

where V_i^0 and V_i are the vertex original position and instantaneous position, respectively.

The user introduces the vertices initial position and the initial length of each segment is computed using the pair of consecutive vertices adjacent to it. During the curve deformation, the resulting force may stretch or compress a segment. If the instantaneous length of s_i is greater than the original length, then f_i^L pulls V_i towards V_{i-1} and vice versa (Figure 52b). On the other hand, if the instantaneous length of s_i is smaller than the original length, then f_i^L pushes V_i away from V_{i-1} and vice versa (Figure 52c).

B. CURVATURE FORCE

The minimum energy position of the guidewire consists of smoothest possible curvatures. A straight structure, when bended, has internal forces that tend to restore the straight position. The curvature forces f^C simulate those internal forces (Figure 53).

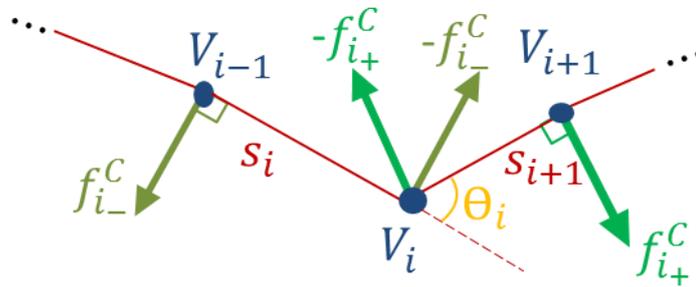


Figure 53: Illustration of f^C , which forces the alignment of V_{i-1} , V_i and V_{i+1} .

Let us define curvature intensity c_i by

$$\text{Eq. 87} \quad \hat{d}_i = \frac{V_i - V_{i-1}}{|V_i - V_{i-1}|}$$

$$\text{Eq. 88} \quad \sin\left(\frac{\theta_i}{2}\right) = \frac{|\hat{d}_{i+1} - \hat{d}_i|}{2}$$

$$\text{Eq. 89} \quad c_i = \frac{\theta_i}{\pi}$$

where \hat{d}_i and \hat{d}_{i+1} are the direction of the two consecutive segments s_i and s_{i+1} . The range of the curvature intensity is $0 \leq c_i \leq 1$, where 0 represents \hat{d}_i and \hat{d}_{i+1} in the same direction (straight position) and 1 represents \hat{d}_i and \hat{d}_{i+1} in the opposite direction.

The force direction \hat{f}_{i+} , applied to vertices V_i and V_{i+1} , is defined by

$$\text{Eq. 90} \quad \left| \begin{array}{l} \hat{f}_{i+} \cdot \hat{d}_i = 0 \\ \hat{f}_{i+} \cdot (\hat{d}_i \times \hat{d}_{i-1}) = 0 \\ |\hat{f}_{i+}| = 1 \end{array} \right.$$

where the first equation ensures that \hat{f}_{i+} is perpendicular to \hat{d}_i , the second equation bounds \vec{v} to belong to the plane defined by \hat{d}_i and \hat{d}_{i-1} and the last equation defines the norm of \hat{f}_{i+} .

The system of equations above provides two solutions of $\hat{f}_{i+} = \{\hat{f}_\alpha; \hat{f}_\beta\}$ in opposite directions, $\hat{f}_\alpha = -\hat{f}_\beta$. Then, the angles $\theta_\alpha = \hat{d}_{i-1} \angle \vec{v}_\alpha$ and $\theta_\beta = \hat{d}_{i-1} \angle \vec{v}_\beta$ are computed; and the solution force related to the angle that is smaller than 90° is applied to V_{i+1} and the other is applied to V_i .

$$\text{Eq. 91} \quad \vec{f}_{i+} = c_i \cdot \hat{f}_{i+} \Rightarrow \left| \begin{array}{l} f_{i+1}^{C+} = \vec{f}_{i+} \\ f_i^{C+} = -\vec{f}_{i+} \end{array} \right.$$

Analogously, the force direction \hat{f}_{i-} , applied to vertices V_{i-1} and V_i , is defined by

$$\text{Eq. 92} \quad \left| \begin{array}{l} \hat{f}_{i-} \cdot \hat{d}_i = 0 \\ \hat{f}_{i-} \cdot (\hat{d}_i \times \hat{d}_{i+1}) = 0 \\ |\hat{f}_{i-}| = 1 \end{array} \right.$$

Again, two vectors compose the solution and the one that provides an angle smaller than 90° with \hat{d}_{i+1} is applied to V_{i-1} and the other is applied to V_i .

$$\text{Eq. 93} \quad \vec{f}_{i-} = c_i \cdot \hat{f}_{i-} \Rightarrow \left| \begin{array}{l} f_{i-1}^{C-} = \vec{f}_{i-} \\ f_i^{C-} = -\vec{f}_{i-} \end{array} \right.$$

Since the curvature of each vertex provides four forces – two at itself, one at the previous vertex and one at the next -, the total curvature force at vertex i is:

$$\text{Eq. 94} \quad f_i^C = c_{i-1} \cdot \hat{f}_{i-1+} - c_i \cdot \hat{f}_{i+} - c_i \cdot \hat{f}_{i-} + c_{i+1} \cdot \hat{f}_{i+1-}$$

C. EXTERNAL FORCE

The external force is simply the force of the vessel wall bounding the guidewire path to its interior.

A lumen mask is created, where the lumen is the object and its exterior is the background. Then, the distance transform is performed [46].

If a vertex instantaneous position is outside the lumen, f_i^E is calculated by

$$\text{Eq. 95} \quad \left| \begin{aligned} \hat{d}_i^E &= \frac{p_{nearest} - V_i}{|p_{nearest} - V_i|} \\ \vec{f}_i^E &= D_{DT} \cdot \hat{d}_i^E \end{aligned} \right.$$

where V_i is the vertex position of the guidewire path that is outside the lumen, $p_{nearest}$ is the position of the nearest voxel inside the lumen, \hat{d}_i^E is the unit vector that defines the direction of f_i^E and D_{DT} is the distance between V_i and $p_{nearest}$.

Since the algorithm checks only the vertices position, it is possible that two neighboring vertices are inside the lumen but the connecting edge crosses the vessel wall. The user may work with a small distance between neighboring vertices, generating a large number of vertices, making the resolution of the curve adequate for checking the guidewire inside the blood vessel. Another option is to perform an interpolation to check if an edge crosses the vessel wall and to apply the forces on the connected vertices.

D. GLOBAL FORCE

The user enters the guidewire initial curve. Then, the global forces \vec{f}_i on each vertex and the new position of the vertex are calculated as follows

$$\text{Eq. 96} \quad \vec{f}_i = w_L \cdot \vec{f}_i^L + w_C \cdot \vec{f}_i^C + w_E \cdot \vec{f}_i^E$$

$$\text{Eq. 97} \quad v_i = v_i \cdot (1 - d_c) + \vec{f}_i \cdot \Delta t / m$$

$$\text{Eq. 98} \quad V_i = V_i + v_i \cdot \Delta t$$

where w_L , w_C and w_E are the weights of \vec{f}_i^L and \vec{f}_i^C , respectively. v_i , d_c , Δt and m are the vertex instantaneous speed, drag coefficient, time interval between each iteration and vertex mass, respectively.

The drag coefficient, $0 \leq d_c \leq 1$, controls the velocity conservation – inertia - from one iteration to the next. If $d_c = 0$, the curve moves faster, but may become unstable or enter a

resonance. If $d_c = 1$, the system becomes quasi-static, the curve deformation is more steady and slower.

Figure 54 illustrates a sequence of frames that shows the progress of a curve to the estimated position.

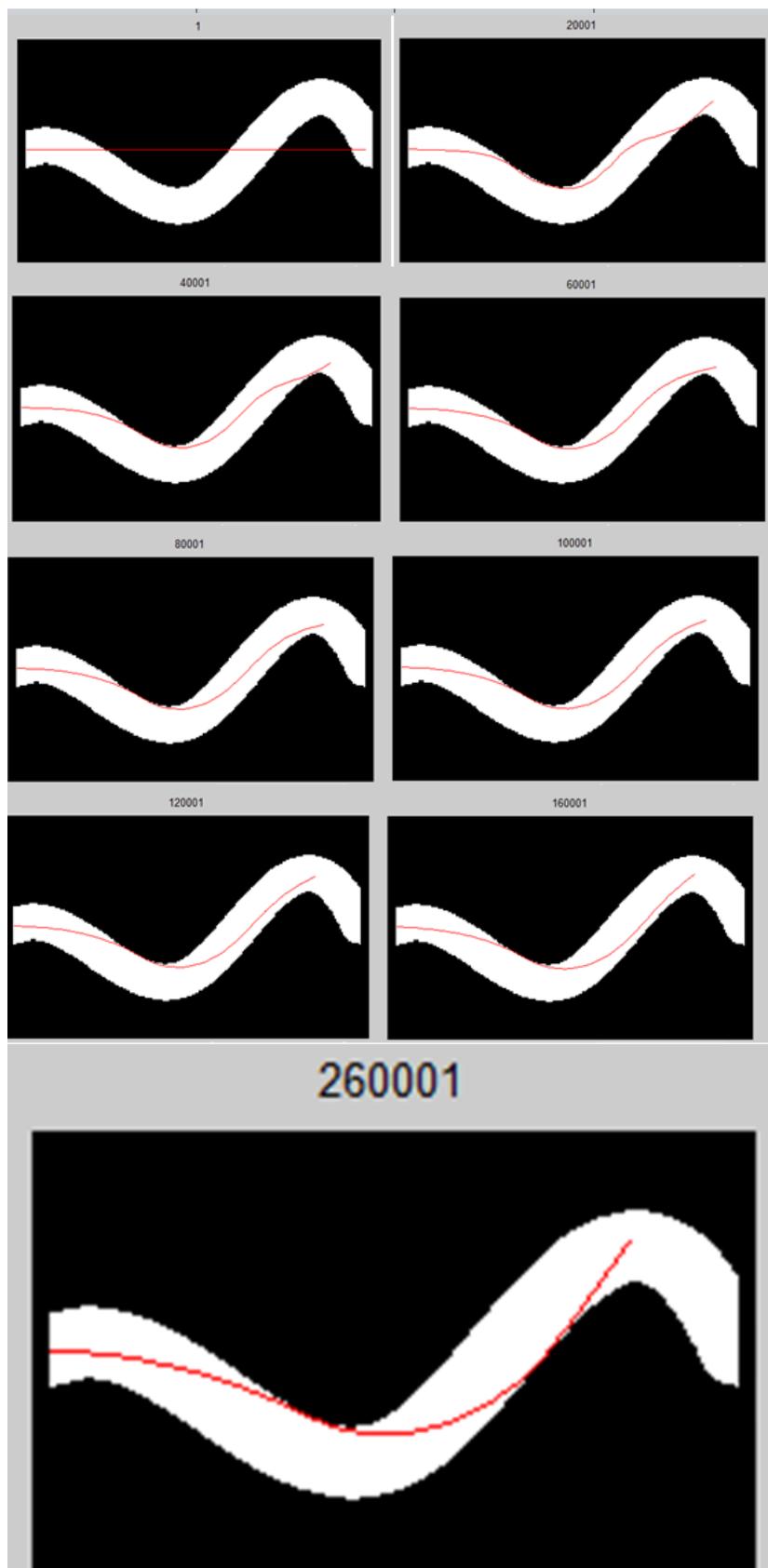


Figure 54: Sequence of frames illustrating the active contour deformation to equilibrium.

5.9.1.3 Comparison of DP and AC with physical phantom

In this section, we present some results involving both guidewire/catheter path simulators presented in this study.

Both techniques were applied in four different physical phantoms (Figure 55) and their performances were evaluated using root mean squared error (RMSE), Hausdorff distance (HD) and Hausdorff Mean (HM).

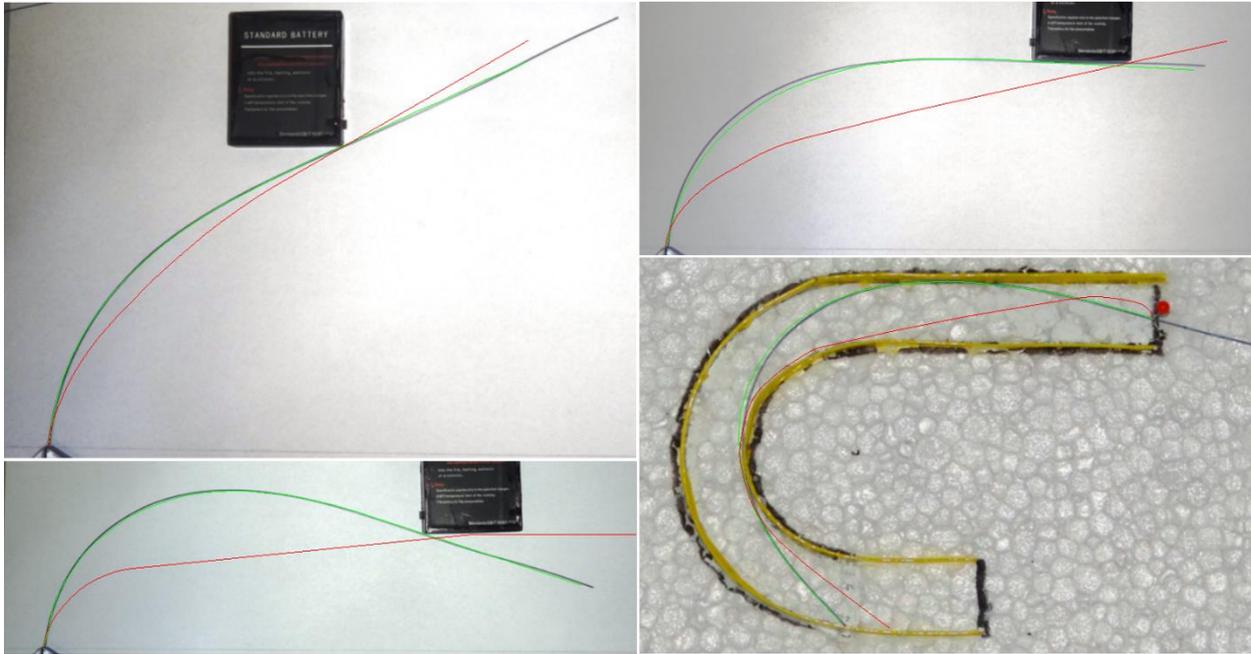


Figure 55: Four physical phantoms utilized during the comparison of performance of the Dynamic Programming and Active contours path simulators. The red line and the green line depict the path estimated by the dynamic programming and active contours, respectively.

RMSE, HD and HM measure how close the simulated curve is to the real one and they are calculated as follows:

$$\text{Eq. 99} \quad RMSE = \sqrt{\frac{\sum_{i=1}^N |p_i^{sim} - p_i^{ref}|^2}{N}}$$

$$\text{Eq. 100} \quad HD = \max\{|p_i^{sim} - p_i^{ref}|\}, i = 1, 2, \dots, N$$

$$\text{Eq. 101} \quad HM = \frac{\sum_{i=1}^N |p_i^{sim} - p_i^{ref}|}{N}$$

where p_i^{sim} is the pixel position of the simulated curve, p_i^{ref} is the position of closest pixel to p_i^{sim} that belongs to the real curve, and N is the number of pixels of the simulated curve.

The curve in the Dynamic Programming method was obtained by the retrieval of the minimum energy path for the corresponding inserted length of the guidewire. The parameters used during the Active Contours method were $w_L = 1$, $w_C = 40$, $w_E = 1$, $d_c = 0.5$, $\Delta t = 1$ and $m = 2$.

Table 12 provides the numeric results of the performance of both methods. Figure 55 also contain the plots of the Dynamic Programming (red) and Active Contours (green) curves.

Table 12: Accuracy results of the guidewire path simulators

Model #	Dynamic Programming			Active Contours		
	RMSE*	HM*	HD*	RMSE*	HM*	HD*
A	3.15	2.63	5.85	0.52	0.29	1.83
B	11.02	9.16	18.39	1.08	0.90	2.15
C	13.00	10.94	22.89	0.60	0.44	1.60
D	5.57	4.27	11.71	0.30	0.20	1.78

*All the measures are in mm.

As seen, in Figure 55 and Table 12, the DP method was unable to predict the guidewire/catheter path. Since the AC method presented good results, only AC was utilized to perform the simulations in this study and the usage of DP was discontinued.

5.9.2 Catheter Pullback Simulation

In this study, we simulated the catheter pullback in two different ways. Figure 56 is compounded by two sequences of four images that illustrate the catheter pullback for both situations. The first one is the free-tip catheter, in which the catheter is free to assume the minimum energy position (Figure 56 a-d). The second is the sheath-covered catheter, where the catheter is covered by a plastic sheath that works as a guide to the catheter (Figure 56 e-h). Therefore, the sheath assumes the minimum energy position and the catheter follows the sheath from inside.

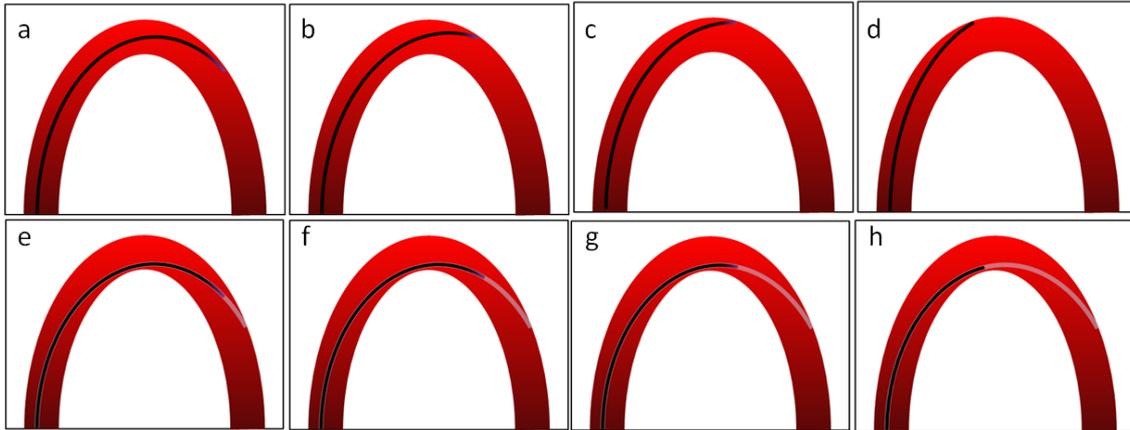


Figure 56: a-d) Pullback sequence of a free-tip catheter. e-f) Pullback sequence of a sheath-covered catheter.

5.9.2.1 Free-Tip Catheter

In order to simulate the catheter pullback, the user introduces the initial catheter length inside the lumen and the pullback step [m/frame]. Then, the algorithm finds the minimum energy path related to the length. Next, the length is decreased one step. Alternately, the algorithm finds the path and decreases one step from the length, until the smallest possible length, as shown in the following pseudo-algorithms.

<p>For DP</p> <p>USER $\rightarrow L_0$ USER $\rightarrow step$ L = L_0 $i_{Frame} = 0$</p> <p>while(L > 0) $i_{Frame} = i_{Frame} + 1$ $c = find(L \leq voxel_{length} < L + step)$ $m = find(voxel(c)_{cost} = \min(voxel(c)_{cost}))$ $catheterPath(iFrame) = C(L)$ L = L - step End</p>	<p>For AC:</p> <p>USER $\rightarrow L_0$ USER $\rightarrow step$ L = L_0 $i_{Frame} = 0$</p> <p>while(L > 0) $i_{Frame} = i_{Frame} + 1$ $c = performACalgorithm(L)$ $catheterPath(iFrame) = c$</p> <p>L = L - step End</p>
--	--

The main difference between both pseudo-algorithms, for DP and AC, is that DP is performed only one time, before the pullback simulation - as mentioned in Section 5.9.1.1, in the catheter path simulation through DP, each path has a related length. On the other hand, AC is performed multiple times, because the catheter path simulation through AC must be performed for each different catheter length.

5.9.2.2 Catheter enclosed by sheath

When using the catheter guided by the sheath, only one computed path is considered. In both methods, DP and AC, the user enters the sheath length and the associated path is retrieved only one time. The catheter tip will travel through the sheath path from distal to proximal portion as illustrated in the sequence of Figure 56 e-h

A cubic spline is computed in order to obtain greater resolution between two consecutive points of the path. Each point of the spline has an associated length value, which is the distance to the proximal point through the curve.

Let $C(x)$ be the curve from the proximal point until the point related to the length x . Then

$$\begin{aligned} C_1 &= C(L_0) \\ C_2 &= C(L_0 - \text{step}) \\ &\vdots \\ C_N &= C(L_0 - (n - 1) \cdot \text{step}) \end{aligned}$$

where C_i, L_0 and step are the i -th curve, the catheter initial length and the decrease of length between two consecutive frames, respectively. Hence, basically, C_i is a fraction of the curve $C_1 = C(L_0)$.

5.9.2.3 Calculation of the position and direction of the catheter tip

Both methods provide the catheter path, which is formed by a sequence of nodes. The catheter tip is located at the distalmost node of the path.

The catheter tip normalized direction is calculated as the difference between the positions of the last node and the position of the penultimate node.

$$\text{Eq. 102} \quad \hat{d}_{cat} = [u, v, w] = \frac{[x_n, y_n, z_n] - [x_{n-1}, y_{n-1}, z_{n-1}]}{\|[x_n, y_n, z_n] - [x_{n-1}, y_{n-1}, z_{n-1}]\|}$$

where $d_{cat} = [u, v, w]$ is the catheter tip normalized direction, $\{x_n, y_n, z_n\}$ are the coordinates of the last and distalmost node of the catheter path and $\{x_{n-1}, y_{n-1}, z_{n-1}\}$ are the coordinates of the penultimate node.

5.9.3 Arrangement of frames from different cardiac phases

In real clinical cases, the IVUS acquisition is performed in living patients. Hence, besides the catheter pullback movement, there is also the cardiac movement. The composition of those

movements leads to rotational and translational motion of the catheter in relation to the blood vessel (Figure 57). In the literature, there are many researches related to that problem in two different ways. One is retrieving a specific cardiac phase from an IVUS sequence [49]. The other is assessing the catheter motion in relation to the blood vessel in order to perform the blood vessel alignment [50].

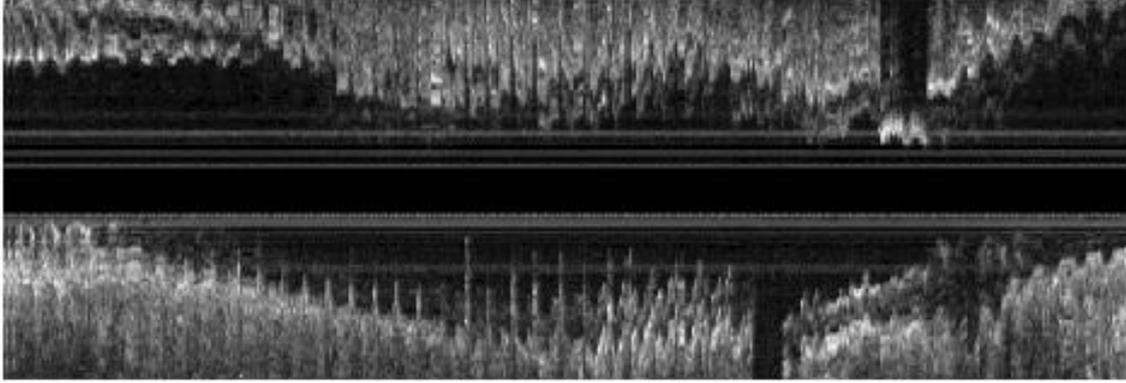


Figure 57: Typical longitudinal cross-section obtained during IVUS pullback. The saw-blade shape “artifacts” are caused by cardiac motion.

The retrieval of a specific cardiac phase from an IVUS sequence consists on separating a set of IVUS images that were acquired at the same phase of the cardiac cycle. In this work, we performed the opposite procedure to obtain a trustworthy numerically simulated IVUS sequence. As explained in Section 11.3.4, the simulated IVUS images were acquired with a static blood vessel. The simulated pullback and speckle noise incorporation were performed on each intraluminal pressure. As a result, we obtained a stack of simulated IVUS images for each cardiac phase. The images were organized in such a way as to produce an IVUS sequence that mimics the composite movements of the pullback and cardiac cycle. According to Arbab-Zadeh [51], the catheter mean displacement along its longitudinal axis is $1.5 \pm 0.8 \text{ mm}$ with maximum of 5 mm .

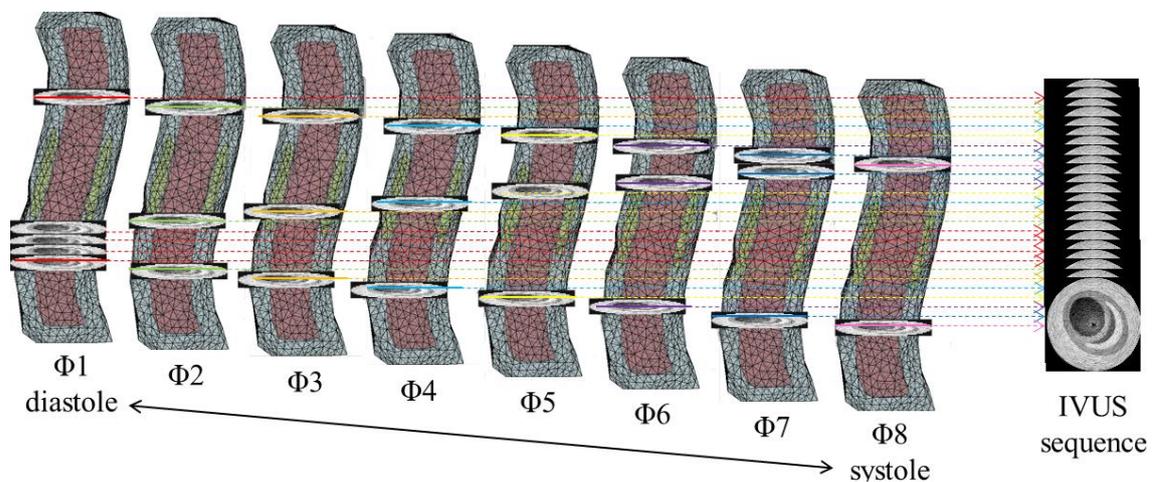
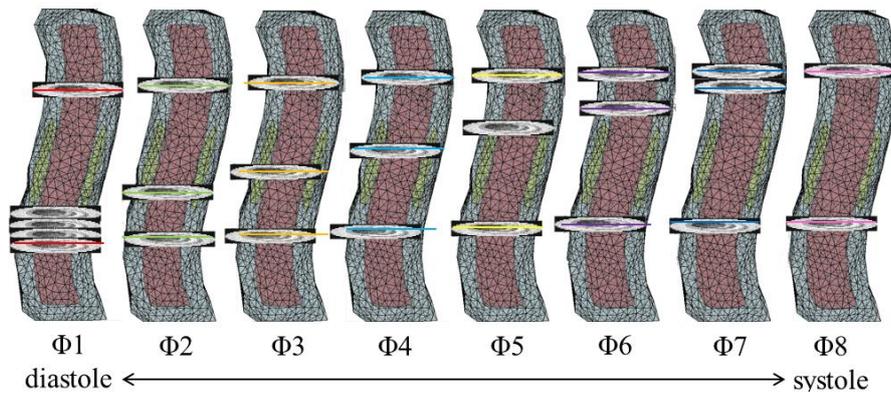


Figure 58: Combination of IVUS images from different cardiac cycles.

The assessment of the catheter motion in relation to the blood vessel consists on estimating the longitudinal, rotational and lateral catheter motion taking the coronary as reference. Observe in Figure 58 that, besides deformed due to blood pressure, the blood vessel phases are gradually misaligned from $\Phi 1$ to $\Phi 8$. This misalignment mimics the catheter longitudinal motion. The rotational motion can be achieved by rotating the IVUS images in the sequence according to the cardiac phase and the lateral motion is performed in the simulated pullback (Section 11.3.4). Figure 59 shows the acquisition distribution after cardiac phases alignment.



6. Structures Movement Estimation in Transcutaneous US

6.1 Performance on Numeric Simulations

In order to perform the computational simulation, four different models were created. All of them consisted on a phantom with a cylindrical lesion (Figure 60). The cylinder had a 30 pixel-radius and its center was located 35 pixels below the surface.

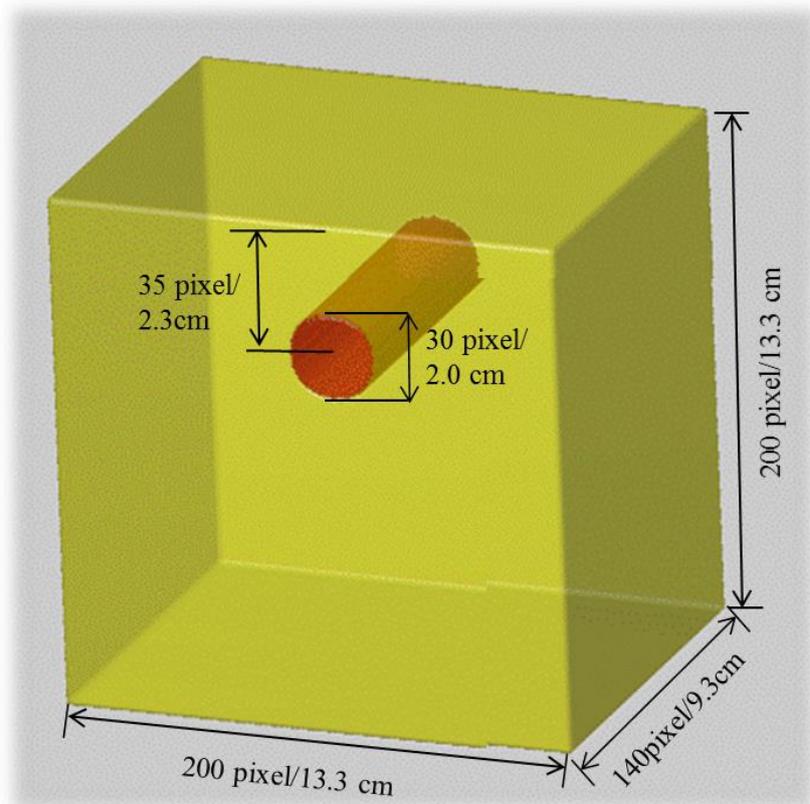


Figure 60: Numerical phantom with cylindrical lesion

The difference between the four models are related to the elastic and acoustic properties of the lesions in relation to the background (Table 13).

Table 13: Differences between the four numeric models

Model	Lesion properties (in relation to background)	
	Acoustics	Elastic
A	Hyperechoic	harder
B	Isoechoic	harder
C	Hypoechoic	softer
D	Isoechoic	softer

Then UltraSSim was performed to simulate the B-mode ultrasound images. A sequence of 11 images with progressive pressure was created to simulate the clinical acquisition in which the physician pressures the tissue with the transducer. The parameters used in the simulation are summarized in Table 14.

Table 14: Parameters used in the numeric simulation

Phantom Properties		
Name	Unit	value
Phantom Size	Pixels	200x200x140
Phantom Size	mm	133x133x93
Phantom Resolution	$\mu\text{m}/\text{pixel}$	667
Background Scatterers Density	units/ mm^3	3
Lesion Scatterers Density	units/ mm^3	3
Background Scattering Amplitude	-	1
Background Elastic Modulus	kPa	30
Lesion Scattering Amplitude*	-	1, 2, 1 and 0.6*
Lesion Elastic Modulus*	kPa	1000, 1000, 1 and 1*
Attenuation	dB/cm/MHz	0
Poisson Ratio	-	0.495
Ultrasound Speed	m/s	1540
Linear Transducer Properties		
Name	Unit	Value
Transducer Contact Surface Size	pixels	60x40
Transducer Contact Surface Size	Cm	4.0x2.7
Number of elements	units	192
Active elements during emission	units	1
Active elements during reception	units	32
Element Height	μm	5000
Element Width	μm	440
Distance between elements	μm	50
Elements-tissue gap	Mm	5
Number of Lines per Image	units	50
Center Frequency	MHz	4
Sampling Frequency	MHz	100
Applied Forces	Kgf	0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0
Applied Pressures	kPa	0.0, 3.7, 7.3, 11.0, 14.7, 18.4, 22.0, 25.7, 29.4, 33.1, 36.7

*models A, B, C and D, respectively

Figure 61 shows the resulting b-mode image from the first frame of the model A, B, C and D.

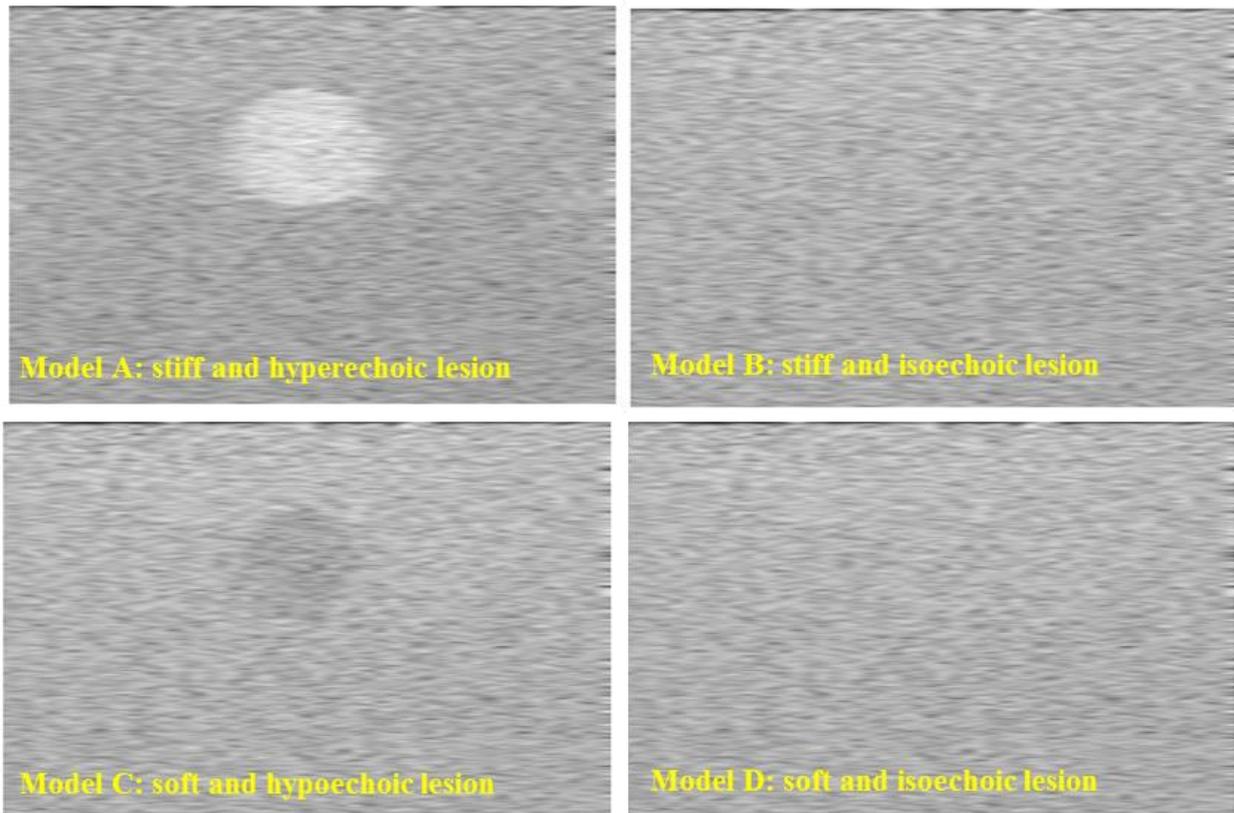


Figure 61: First frame of one of the simulations of each model.

The simulation was performed 5 times with different scatterer distribution to each model, hence, generating 20 movies. Each movie consisted in 11 frames with 693x1050 pixels.

In order to mitigate the effects of the boundaries, the results next to the borders were neglected (Figure 62): 15 top lines, 15 bottom lines, 100 left columns and 100 right columns.

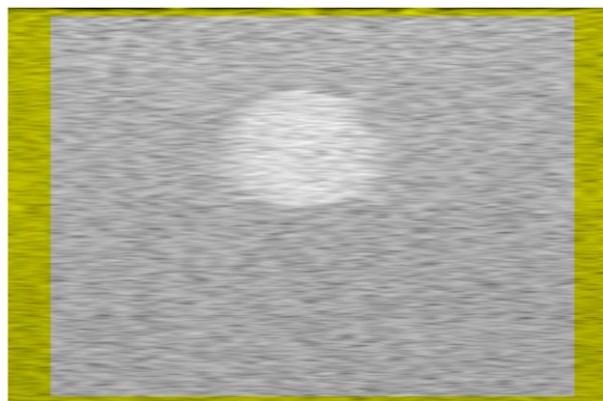


Figure 62: Border region, in yellow, representing the position of the neglected results to minimize the effect of the boundaries.

6.1.1 Optical Flow

Optical flow was performed in the B-mode image sequence with different configurations: 10x10 and 20x20 window size, and 5 and 10 iterations. The procedure was performed on every pixel of the image, using Lucas-Kanade algorithm.

Models A and B (hard lesion) presented similar results; therefore, their results were averaged together and are shown in Table 15. Analogously, the results from models C and D (soft lesion).

Table 15: Movement Estimation Errors of Optical Flow on models A and B

Number of iterations	Window Size	
	10x10	20x20
10	5.65±0.36	3.84±0.12
20	5.68±0.28	3.81±0.10

*All the errors are in pixels.

Analogously, the results from models C and D (soft lesion) were averaged together, because they were also similar, and are shown in Table 16.

Table 16: Movement Estimation Errors of Optical Flow on models C and D

Number of iterations	Window Size	
	10x10	20x20
10	16.79±1.2	15.77±0.91
20	16.84±1.25	15.79±0.90

*All the errors are in pixels.

Optical Flow presented better results with bigger window size. However, no significant difference was observed with the increase of number of iterations from 10 to 20.

The errors from the phantoms with hard lesions were up to 4 times smaller in relation to the errors from the phantoms with soft lesions.

6.1.2 2D Block Matching

We studied the performance of the 2D Block Matching with different matching functions, namely sum of absolute differences (SAD, Eq. 42), sum of squared differences (SSD, Eq. 43) and

normalized correlation coefficient (NCC, Eq. 44). For each function, we also tested different square window sizes: 11x11, 21x21 and 31x31 pixels.

Block matching was performed on every pixel of the image, where each pixel represents the center of the matching window.

For the same function and window size, there was no significant difference between the results of the four models (A, B, C, and D). Therefore, their results were averaged together and are shown in Table 17.

Table 17: Movement Estimation Errors of 2D Block Matching with different configurations

Function	Window Size		
	10x10	20x20	30x30
SAD	1.26 ± 0.06	1.08 ± 0.09	1.36 ± 0.07
SSD	1.38 ± 0.04	1.09 ± 0.08	1.38 ± 0.06
NCC	1.36 ± 0.04	1.09 ± 0.08	1.33 ± 0.06

*All the errors are in pixels.

As seen in Table 17, for all functions, the performance was slightly better with window size 20x20 pixels. Moreover, for the same window size, there was no significant difference between the different matching functions. Therefore, from now on, only SAD will be used in this work, because that is the simplest and fastest matching function among the three of them.

6.2 Performance on Physical Phantoms

We created physical phantoms in order to analyze the performance of both optical flow and 2D block matching using real ultrasound equipment. The heterogeneous phantoms consisted in a background with a 1-cm-diameter cylindrical inclusion were built using unflavored gelatin and fiber. The phantom was built in a 3-step process (Figure 63). First, the 2-cm-height lower background was built. Second, the inclusion was built and put on the top of the lower background. Third, the rest of the background was filled until its surface was 2 cm above the inclusion.

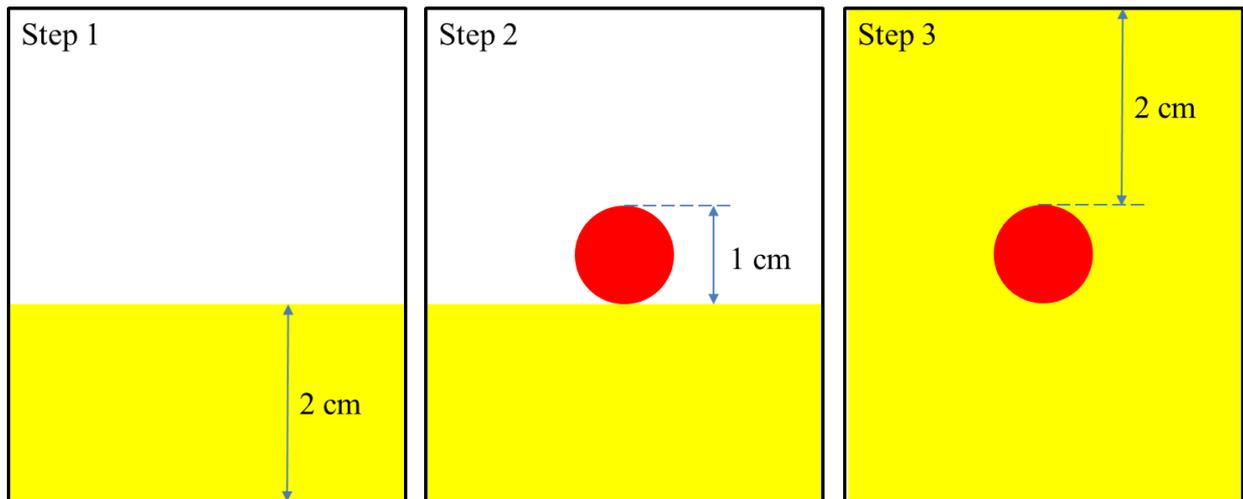


Figure 63: Three-step process used to build a heterogeneous phantom with cylindrical inclusion

Four different phantom models were built using unflavored gelatin and sugar-free psyllium hydrophilic mucilloid fiber [52]. The material composition of the background and lesion of the models are summarized in Table 18, which also contains the inclusion mechanical and acoustic characteristic in relation to the background.

Table 18: Composition and properties of the physical phantom structures

Structure	Material Composition (grams per liter of water)		Inclusion properties (in relation to background)	
	Gelatin	Psyllium	Elastic	Acoustics
Background	30	45	-	-
Inclusion A	56	45	harder	Isoechoic
Inclusion B	56	20	harder	Hypoechoic
Inclusion C	15	45	softer	Isoechoic
Inclusion D	30	70	isoelastic	Hyperechoic

The B-mode images were acquired using the Ultrasound System SonixTouch Q+ (Analogic Ultrasound™) [53]. The acquisition process was performed using a phased array transducer with the following setup (Table 19).

Table 19: Configuration used in the B-mode imaging

Name	Value
Center Frequency (MHz)	10 MHz
Imaging Depth (cm)	4 cm
Focal Points	0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 3.5 , 4.0
Dynamic Range (dB)	65
FPS	13

In addition to the B-mode image sequence, real-time strain elastography was also acquired to be used as reference for visual comparison to the performance of both 2D Block Matching and Optical Flow (Figure 64).

After the B-mode image acquisition, 2D Block Matching (window size 21x21 pixel) was performed in the image sequence in order to estimate the displacement of each pixel. Then, a strain map was built by the computation of the local strain in each pixel as follows

$$\text{Eq. 103} \quad \varepsilon(x, y) = \frac{u_x(x+1, y) - u_x(x-1, y)}{2}$$

where $\varepsilon(x, y)$ is the local strains at the position (x, y) and u_x is the axial displacement.

Analogously, optical flow (window size 20x20 and 5 iterations) was also applied to the B-mode image sequence in order to estimate the tissue movements and build a strain map.

Figure 64 illustrates a B-mode image, the elastography provided by the ultrasound equipment, the strain map computed using 2D Block Matching and the strain map computed using optical flow for each physical phantom.

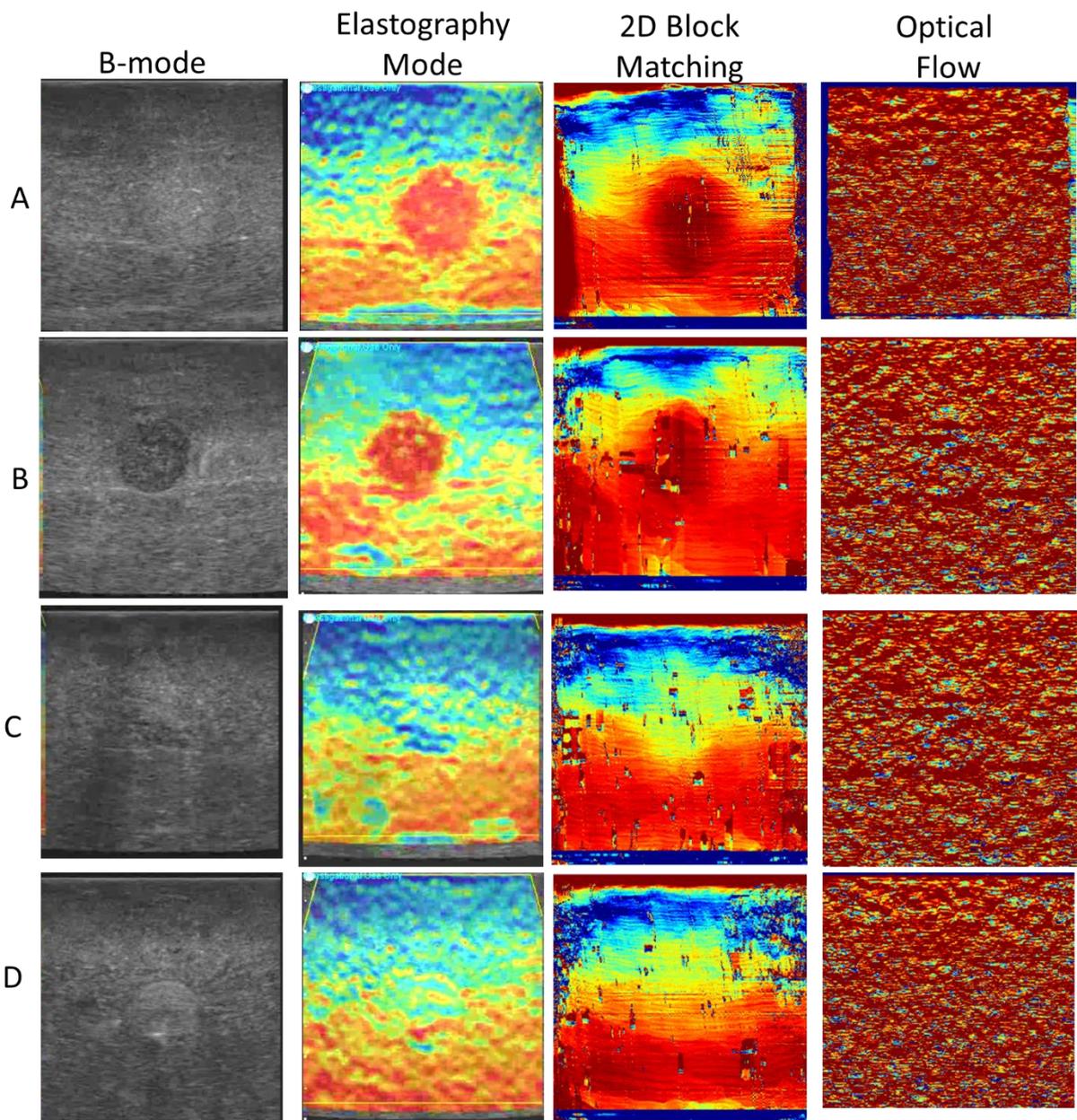


Figure 64: Visual assessment of the elastography performed in the four physical phantoms. From left to right, B-mode image, strain elastography from the equipment, elastography computed using 2D Block Matching and Optical Flow, respectively.

It is easy to observe that the strain maps computed with 2D Block Matching are compatible to the elastography from the ultrasound equipment. In model A and C, the inclusion, which is not visible in the B-mode image, can be located in the strain maps. In model D, the inclusion is visible in the B-mode, but invisible in the strain map, which is expected, since the background and the inclusion have the same elasticity. On the other hand, optical flow provided poor movement estimation in all cases.

7. Structures Movement Estimation in Intravascular Ultrasound

As explained in Section 1.2.2 and Section 5.9.2, IVUS imaging is performed during the catheter pullback. In other words, each frame represents a different transversal cut the blood vessel. Besides, since the IVUS images are obtained from living patients, the heart contractions displaces the catheter inside the blood vessel. Those facts hinder the movement estimation of a specific position of the blood vessel.

In this study, we will quantitatively analyze the impact of the cardiac contractions in movement estimation utilizing numeric simulation.

The simulation of the sequence of IVUS images was performed as explained in Section 11.3 with the parameters shown in Table 20 and the model illustrated in Figure 65.

Table 20: Parameter used in this simulation

Parameter	Value
Ultrasound Frequency	40 MHz
Number of transducers	1
Number of US beams per image	256
Pullback Speed	0.5 mm/s
Heart Rate	90 bpm
Frame Rate	15 fps
Frames per Cardiac Cycle	10
Imaged length	5 mm

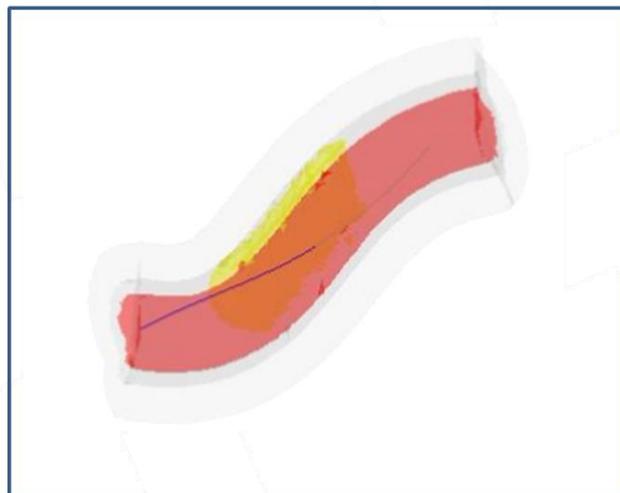


Figure 65: Model of coronary with lesion (in yellow).

In order to simulate the heart dynamics, the cardiac cycle was divided into 10 phases. Each phase has a different incremental pressure that is related to the variation of blood pressure and may deform the blood vessel from inside. The longitudinal displacement due to cardiac motion is also being considered. According to [51], in a cardiac cycle the catheter moves 1.5 ± 0.8 mm longitudinally inside the blood vessel. This phenomenon was simulated by changing the catheter (and sheath) penetration length. The pressure increase and longitudinal displacement for each cardiac phase are shown in Table 21. A total of 10 seconds of continuous IVUS was simulated, comprising 5 mm of pullback, 151 frames, and 16 frames of phase 1.

Table 21: Catheter longitudinal displacement (LD) in mm and pressure increase (PI) in mmHg for each phase

Phase	PI	LD	Phase	PI	LD
1	0	0	6	30	-1.2
2	20	-0.7	7	25	-0.9
3	30	-1.3	8	20	-0.6
4	40	-1.5	9	12	-0.3
5	35	-1.4	10	6	-0.1

In order to compare the impact of cardiac motion on structure tracking, 2D Block Matching and Optical Flow were applied to two different sets of images: 1) The whole sequence of IVUS images, considering the 10 phases, and 2) The set of images that belong to phase 1. Then, both results were compared to the gold-standard, which was obtained manually. Since images that belong to a same cardiac phase (set 2) will not be affected by heart movement, it is expected that the error will be less than for the set 1. However, pullback movement will affect both sets, more severely on set 2.

Figure 66 illustrates the first image from the sequence of IVUS images with the 3 points used in the structure tracking.

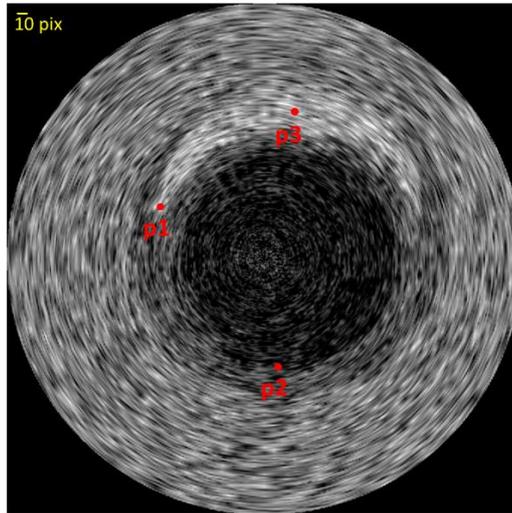


Figure 66: Simulated IVUS image and the points that were tracked: p1 at the left border of the lesion, p2 at the inferior limit of the lumen and p3 at the center of the lesion. The yellow horizontal line at the upper left shows length of 10 pixels.

Figure 67 and Figure 68 show the movement estimation error for the 2D Block Matching and Optical Flow, respectively, of the 16 images from phase 1 and the first 16 images from the whole IVUS sequence (a-c). It also presents the error along the complete IVUS sequence with the error of images from phase 1 in their related position (d-f). All error results are shown in pixels. The yellow horizontal line at the upper left of Figure 66 shows magnitude of a 10-pixel error.

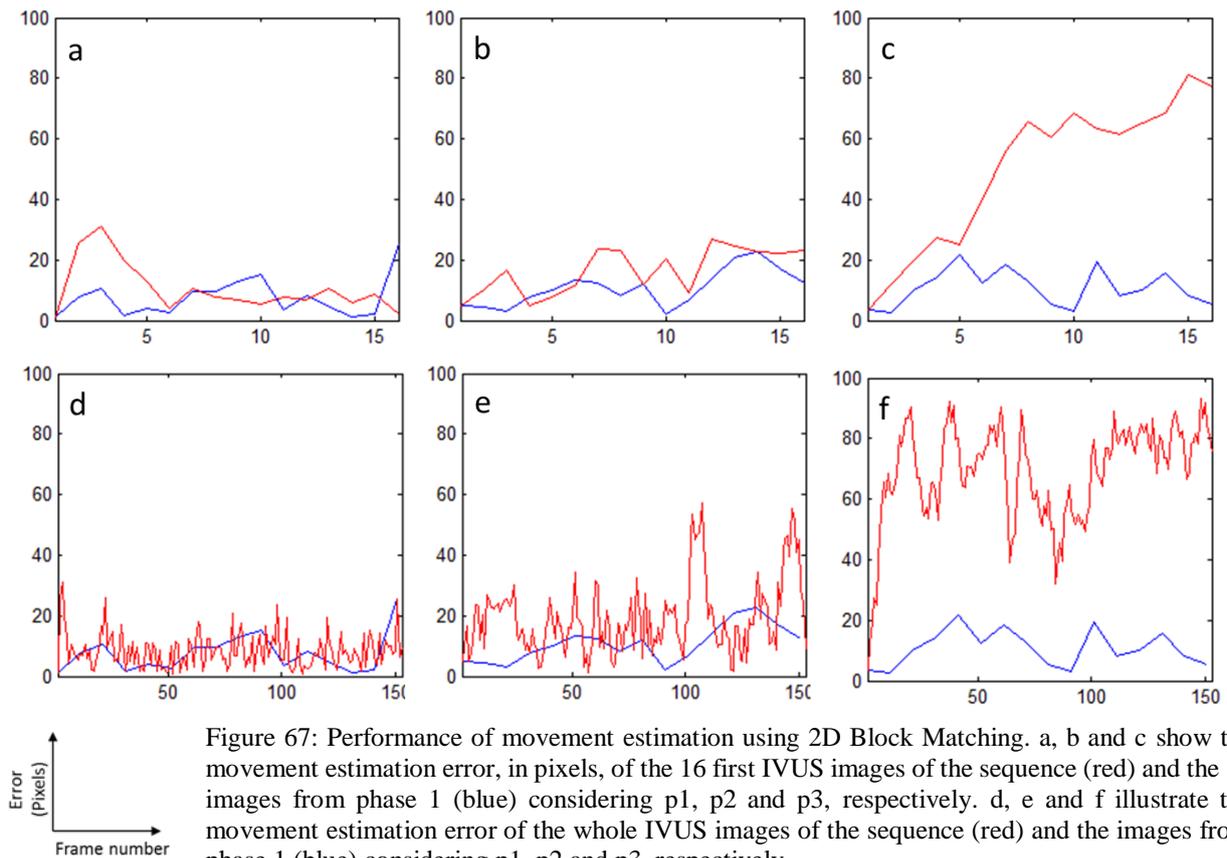


Figure 67: Performance of movement estimation using 2D Block Matching. a, b and c show the movement estimation error, in pixels, of the 16 first IVUS images of the sequence (red) and the 16 images from phase 1 (blue) considering p1, p2 and p3, respectively. d, e and f illustrate the movement estimation error of the whole IVUS images of the sequence (red) and the images from phase 1 (blue) considering p1, p2 and p3, respectively.

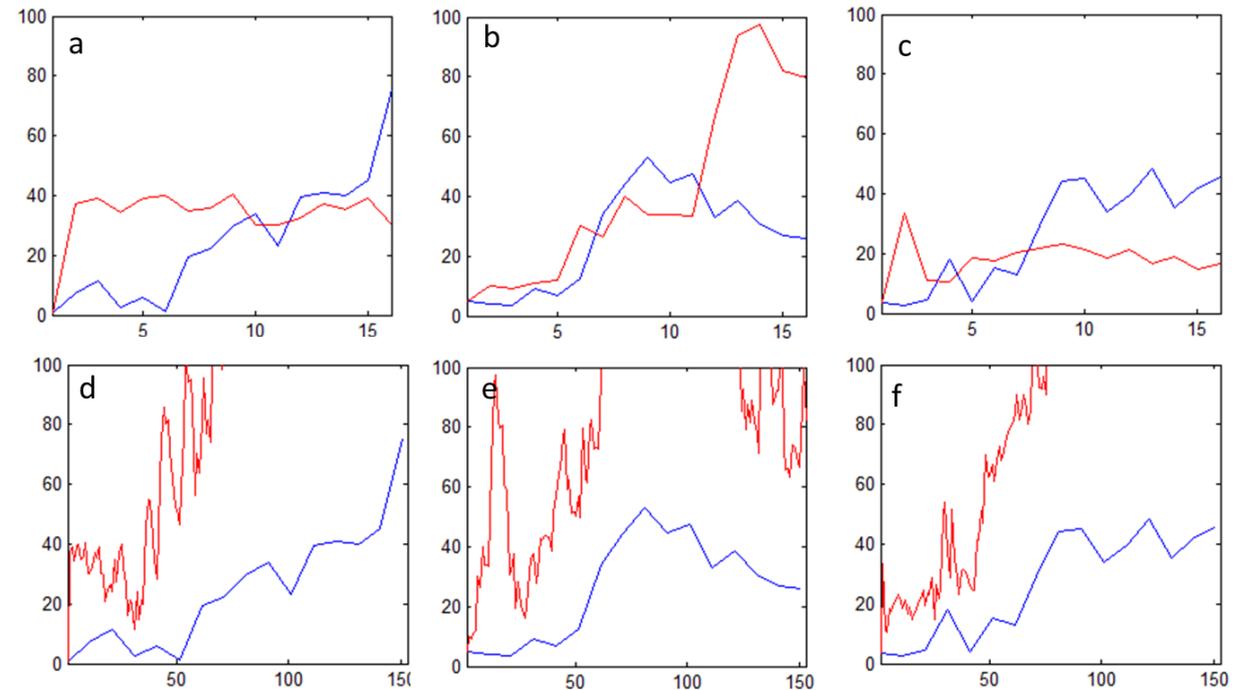


Figure 68: Performance of movement estimation using Optical Flow. a, b and c shows the movement estimation error, in pixels, of the 16 first IVUS images of the sequence (red) and the 16 images from phase 1 (blue) considering p1, p2 and p3, respectively. d, e and f illustrate the movement estimation error of the whole IVUS images of the sequence (red) and the images from phase 1 (blue) considering p1, p2 and p3, respectively.

From Figure 68, it is possible to see that, in the case of the points located at the lumen and lesion borders, the movement estimation errors with the heart motion were close to the errors considering only one phase of the cardiac cycle. However, the point located at the middle of the lesion structure has poor movement estimation in the presence of cardiac motion, while the movement estimation considering the same phase presented good results. Therefore, heart motion is more likely to hinder the tracking of structures in homogeneous regions. This analysis shows the importance of IVUS synchronization with EKG for quantitative purposes and suggests that tracking techniques provides results that are more reliable with frames separated according to cardiac phases.

2D block matching provided poor movement estimation, because each frame represents an acquisition in a different transversal cut. In other words, speckle tracking is a limited technique when there is no speckle pattern deformation/translation to follow.

This study considered the catheter longitudinal displacement and blood vessel deformation due to variation on internal pressure, which reproduces the dynamic effects of clinical IVUS acquisition. Future work may include different examples as well other IVUS artifacts, such as catheter twist, which causes image rotation.

8. Conclusions and Discussions

In Section 4.2, it was shown that ISF and ISFAD are simple and effective techniques to remove the speckle texture without blurring the edges of the imaged structures. Those filters are able to improve visualization and the performance of some computational procedures, such as segmentation (Table 6). However, they were not utilized in the structures movement estimation (Section 6 and Section 7) of this work for two reasons. First, the adopted methodology consisted in applying 2D block matching and Optical Flow directly to the B-mode images. In other words, the tracking methods should be performed and evaluated without the interference of any other image processing techniques, such as filters. Second, the speckle texture removal would not improve the tracking methods because the local variation is important to estimate displacement, especially in homogeneous region.

UltraSSim (Section 5) utilizes finite elements method and linear isomorphism to estimate the scatterers displacement. Therefore, the tissue deformation needs to be sufficiently small in order to be assumed linear. In other words, the estimation may not be valid for large compression.

IVUSSim (Section 5.9) is an intravascular ultrasound simulator that is able to deform the blood vessel according to the variation of its internal pressure. It is also able to predict the position of the catheter inside the blood vessel, which is important to determine the plane that is being imaged. Those abilities are suitable for the imaging of coronary with no (or very low) movement or deformation, such as carotid. On the other hand, the application of IVUSSim on coronary simulation is limited because it does not take in consideration the blood vessel movement and deformation due to the cardiac contraction. Future work may contemplate an anatomic and dynamic study of the coronaries during cardiac contractions in order to make IVUSSim more realistic for numeric simulation involving coronary.

2D Block Matching performs a pixel-by-pixel comparison (Section 3.5.2, Eq. 42-Eq. 44); therefore, its performance depends on a ‘well-behaved’ speckle texture deformation. Hence, during the image acquisition with transcutaneous ultrasound, the compression must be operated strictly in the axial direction. In other words, there must have no transducer movement in the elevational direction, otherwise the speckle pattern would change drastically and the speckle tracking would be unfeasible in homogeneous regions. This is coherent to the poor results of 2D Block Matching in intravascular ultrasound, where the consecutive frames are acquired in different planes.

Optical Flow presented poor results for the transcutaneous ultrasound with numerical and physical phantoms. This may be caused by the low frame rate. The flow between consecutive

frames was too large for the differential equation of the optical flow to hold – usually, it works when there is less than the pixel spacing. Especially on homogeneous regions, once the point ‘loses its track’, it tends to follow the wrong path due to the apparent repetition of the speckle pattern throughout the image. The application of the pyramid procedure [33] to obtain a coarse estimation was ineffective because it alters the speckle texture, which is used to estimate the flow. In the intravascular ultrasound numerical ultrasound, better results were obtained with the tracking points located at the border of structures.

8.1 Main contributions

8.1.1 Novel Computational Techniques

During the development of this work, five novel computational techniques involving ultrasound imaging were created:

- a filter that is able to remove the speckle texture without blurring the edges of the structures (ISFAD, Section 3.4.2),
- a parameter based on the speckle noise texture to estimate the window size of speckle filters (SNOL, Section 3.4.5),
- a technique to predict the scatterer displacement due to tissue deformation based on finite elements method and linear isomorphism (Section 5.3.2), which led to the development of a computer program to simulate ultrasound acquisition with mechanical compressions,
- a method based on equilibrium of forces to determine the position of a catheter inside a blood vessel (Section 5.9.1.2)
- an innovative implementation of 2D block matching with sub-pixel linear interpolation and displacement propagation (Section 3.5.2).

8.1.2 Comparative Results

Using numerical and physical phantoms, the performances of different methods to solve a specific problem were compared qualitatively and quantitatively.

- Comparison between two speckle noise simulators (Section 4.1)
- Comparison of the performance of 11 different speckle noise filters (Section 4.2)

- Comparison of different methods to predict the catheter position inside a blood vessel (Section 5.9.1)
- Comparison between optical flow and 2D block matching performances on estimating the tissue movement due to axial compression on transcutaneous ultrasound (Section 6)
- Comparison of speckle tracking performance applied on IVUS B-mode images from only one phase of the cardiac cycle and from continuous acquisition (Section 7)

8.2 Future works

Inclusion of transducer displacement and tilting in the elevational direction for tridimensional approaches using UltraSSim (Section 5).

Include, in IVUSSim (Section 11.3), catheter twist, which causes image rotation. Future work may contemplate an anatomic and dynamic study of the coronaries during cardiac contractions in order to make IVUSSim more realistic for numeric simulation involving coronary.

Develop a technique to select automatically points that contains good tracking features for optical flow in a sequence of IVUS images and neglect the points that tend to lead to errors.

9. Publications

(2010) M. M. S. Matsumoto, F. M. Cardoso, P. A. Lemos, S. S. Furuie, “Coronary 3D reconstruction using IVUS images only: a numeric phantom investigation”. In: SPIE Medical Imaging 2010: Ultrasonic Imaging, Tomography, and Therapy, San Diego (USA), vol. 7629, Feb. 2010.

(2010) F. M. Cardoso, M. M. S. Matsumoto, S. S. Furuie, “Filter performances on ultrasound images”. In: XXII CBEB, Tiradentes (MG), Nov. 2010.

(2011) F. M. Cardoso, M. M. S. Matsumoto, S. S. Furuie, “Interference based speckle filter”. In: SPIE Medical Imaging 2011: Ultrasonic Imaging, Tomography, and Therapy, Orlando (USA), vol. 7968, Feb. 2011.

(2011) F. M. Cardoso, M. C. Moraes, S. S. Furuie, “A Framework to Create Realistic IVUS Phantoms for Different Intraluminal pressures”. In: Computing in Cardiology, Hangzhou (China), Sep. 2011.

(2012) S. S. Furuie, F. M. Cardoso, “Estimation of deformations in ultrasound images using dynamic programming”. In: SPIE IS&T Electronic Imaging 2012, Burlingame (USA), vol. 8295, Feb. 2012.

(2012) M. C. Moraes, F. M. Cardoso, S. S. Furuie, “Atherosclerotic plaque characterization using plaque area variation in IVUS images during compression: a computational investigation”. *Revista Brasileira de Engenharia Biomédica* 30 (2), 159-172.

(2012) M.C. Moraes, F. M. Cardoso, S. S. Furuie, “Atherosclerotic tissue classification by plaque area ratio in IVUS images”. In: XXIII CBEB, Porto de Galinhas (PE), out. 2012.

(2012) F. M. Cardoso, M. M. S. Matsumoto, S. S. Furuie, “Edge-Preserving Speckle Texture Removal by Interference-Based Speckle Filtering Followed by Anisotropic Diffusion”. *Ultrasound in Medicine and Biology*, vol. 38, No. 8, pp. 1414-1428, Mar. 2012.

(2012) F. M. Cardoso, M.C. Moraes, S. S. Furuie, "Realistic IVUS image generation in different intraluminal pressures". *Ultrasound in Medicine and Biology*, vol. 38, No. 12, pp. 2104-2119, Aug. 2012.

(2014) F. M. Cardoso, S. S. Furuie, "Guidewire path simulation using equilibrium of forces", Proc. SPIE 9033, *Medical Imaging 2014: Physics of Medical Imaging*, 90334U, San Diego (USA), Mar 2014

(2014) F. M. Cardoso, M.C. Moraes, S. S. Furuie, "The impact of cardiac motion on movement estimation of structures in intravascular ultrasound." In: XXIV CBEB, Uberaba (MG), out. 2014.

(2014) F. M. Cardoso, S. S. Furuie, 'Automatic Window Size Estimation for Speckle Noise Filters', 6th European Conference of the International Federation for Medical and Biological Engineering IFMBE Proceedings Volume 45, 2015, pp 208-211

UltraSSim: Matlab-based software that simulates tridimensional transcutaneous ultrasound acquisition with manual compression.

IVUSSim: Matlab-based software that simulates tridimensional intravascular ultrasound imaging during catheter pullback.

Both software, UltraSSim and IVUSSim, are available online and free from charges for purposes of research and education.

10. References

- [1] J. C. Bamber, “Diagnostic ultrasound.” S. Webb, The physics of medical imaging. New York: Taylor e Francis Group; 1988. pp 319-388.
- [2]<http://www.bidmc.org/YourHealth/HealthNotes/MedicalImaging/ImagingSafely/WhichTestsExposeYouToRadiationAndWhichOnesDont.aspx>. Last Accessed:18/12/2014
- [3] F. M. Cardoso, M. M. S. Matsumoto, S. S. Furuie, “Edge-Preserving Speckle Texture Removal by Interference-Based Speckle Filtering Followed by Anisotropic Diffusion.” Ultrasound in Medicine and Biology, vol. 38, No. 8, pp. 1414-1428, Mar. 2012.
- [4] F. Ragazzoni, M. Deandrea, A. Mormile, M. J. Ramunni, F. Garino, G. Magliona, M. Motta, B. Torchio, R. Garberoglio, P. Limone, “High Diagnostic Accuracy and Interobserver Reliability of Real-Time Elastography in the Evaluation of Thyroid Nodules”, Ultrasound in Medicine and Biology, vol. 38 (7) , pp. 1154-1162, July 2012
- [5] T. Varghese, “Quasi-Static Ultrasound Elastography”, Ultrasound Clinics, vol. 4 (3), pp. 323–338, July 2009
- [6] Global status report on noncommunicable diseases 2010. Geneva, World Health Organization, 2011.
- [7] http://www.full-health.com/atherosclerosis_complications.htm; Last Access: 18/12/2014
- [8] C.G. Costa, Endemias Cardiovasculares no Brasil. . A.C. Lopes, Tratado da Clínica Médica. 1st Ed. São Paulo: Editora Rocca; 2006. Pp. 402-406.
- [9] Available on: <http://www.unifesp.br/denf/NIEn/CARDIOSITE/dados.htm> . Last access: 14/1/2013
- [10] Available on:
<http://www.americanheart.org/downloadable/heart/1140534985281Statsupdate06book.pdf> . Last access: 14/1/2013.
- [11] MINISTÉRIO DA SAÚDE DE PORTUGAL. Portal da Saúde – Portugal – 2006. Available on:
<http://www.portaldasaude.pt/portal/conteudos/enciclopedia+da+saude/doencas/doencas+do+aparelho+circulatorio/doencascardiovasculares.htm> . Last Access: 14/1/2013.
- [12]<http://www.world-heart-federation.org/publications/heart-beat-e-newsletter/heart-beat-march-2013/advocacy-news/reduce-cvd-deaths-by-25-by-2025>. Last access: 31/12/14
- [13] Available on: <http://www.lab.anhb.uwa.edu.au/mb140/corepages/vascular/vascular.htm> . Last Access: 14/1/2013.
- [14] Available on: <http://www.normatem.com/vp.html> . Last Access: 14/1/2013.
- [15] User manual. GalaxyTM Sistema de Ultrassom Intravascular. Volume 1. DOC # 50571-001. Boston Scientific.

- [16] Available on: <<http://www.ptca.org/ivus/ivus.html> . Last Access: 14/1/2013.
- [17] Available on: <http://www.emodinamica.gise.it/34/3404.htm>
- [18] Available on:
http://www.clevelandclinic.org/heartcenter/pub/professionals/presentations/ivus_print.htm . Last Access: 14/1/2013.
- [19] J. C. Bamber, R. J. Dickinson, "Ultrasonic B-scanning: A computer simulation," *Phys. Med. Biol.*25, pp. 463-479, 1980.
- [20] Y. Yu, S. T. Acton, "Speckle Reducing Anisotropic Diffusion", *IEEE transactions on imaging processing*, vol.11, no.11, pp.1260-1270, Nov.2002
- [21] P. Abbott, M. Braun, "Simulation of ultrasound image data by a quadrature method," in *Proc. Eng.Phys. Sci. Med. Health Conf.*, 209, 1996.
- [22] J. A. Jensen, "Field: A program for simulating ultrasound systems," *Medical & Biological Engineering & Computing*, Supplement 1, Part 1 34, pp. 351-353, 1996.
- [23] Available on: <http://field-ii.dk/> . Last Access: 14/1/2013.
- [24] Available on: <<http://www.hitachi-medical-systems.eu/products-and-services/ultrasound/hitachi-real-time-tissue-elastography-hi-rte/clinical-applications.html>>. Last Access: 14/1/2013.
- [25] R. L. Maurice, M. Daronat, J. Ohayon, E. Stoyanova, F. S. Foster, G. Cloutier, "Non-invasive high-frequency vascular ultrasound elastography.", *Physics in medicine and biology*, vol.50, pp. 1611-1628, Mar 2005.
- [26] P. Coupé, P. Hellier, C. Kervrann, C. Barillot, "Nonlocal means-based speckle filtering for ultrasound images.", *IEEE Trans Image Process* 2009;18:10:2221-2229.
- [27] P.C. Tay, G.D. Garson, S.T. Acton, J.A. Hossack, "Ultrasound despeckling for contrast enhancement.", *IEEE Trans Image Process* 2010;19:7:1847-1860.
- [28] Z. Wang, A.C. Bovik, R Mongrain, H.R. Sheikh, E.P. Simoncelli, "Image Quality Assesment: From Error Visibility to Structural Similarity," *IEEE Trans Image Process*, v. 13, n. 4, p. 600-612, Apr. 2004.
- [29] D.P. Huttenlocher , A. Klanderman , W.J. Rucklidge, "Comparing images using the Hausdorff distance", *IEEE Trans Pattern Anal Mach Intell* 1993;15:9:850-863.
- [30] J. Udupa, S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models Image Process.*, vol. 58, no. 3, pp. 246-261, 1996.
- [31] C. Xu, J.L. Prince, "Snakes, shapes, and gradient vector flow." *IEEE Trans Image Process* 1998;7:3:359-369.
- [32] C. Li, C. Xu, C. Gui, M.D. Fox, "Level Set Evolution Without Re-initialization: A New Variational Formulation". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2005.

- [33] B. D. Lucas, T. Kanade. “An iterative image registration technique with an application to stereo vision.”, International Joint Conferences on Artificial Intelligence 1981
- [34] A. Hein, W. D. O’Brien, “Current time – domain methods for assessing tissue motion by analysis from reflected ultrasound echoes – a review”, IEEE Trans. Ultras. Ferro. Freq. Control, vol.40, no.2, pp. 84-102, Mar 2003
- [35] www.leb.usp.br/UltraSSim
- [36] P.I. Kattan, “MATLAB Guide to Finite Elements: An Interactive Approach”, 2nd Edition. New York: Springer Berlin Heidelberg, 2008.
- [37] J.N. Reddy, “An Introduction to the Finite Element Method”, 3rd Edition. New York: McGraw-Hill, 2006.
- [38] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, “The Finite Element Method: Its Basis & Fundamentals.”, 6th edition. Great Britain: Elsevier; 2010.
- [39] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, “The Finite Element Method: For Solid and Structural Mechanics.”, 6th edition. Great Britain: Elsevier; 2010
- [40] Q. Fang, D.A. Boas, “Tetrahedral mesh generation from volumetric binary and gray-scale images,” Proceedings of IEEE International Symposium on Biomedical Imaging 2009, pp. 1142-1145, 2009
- [41] Available on: <http://iso2mesh.sourceforge.net/cgi-bin/index.cgi> . Last access: 14/01/2013.
- [42] M.G. Danilouchkine, F. Mastik, A.F.W. van der Steen. “A study of coronary artery rotational motion with dense scale-space optical flow in intravascular ultrasound.”, Phys. Med. Biol. 54 (2009) 1397–1418
- [43] H. Anton, C. Rorres, “Elementary Linear Algebra: Applications Version”, 9th Edition. John Wiley & Sons, Inc 2005.
- [44] F. Kallel, C.D. Prihoda, J. Ophir, “Contrast-Transfer Efficiency For Continuously Varying Tissue Moduli: Simulation And Phantom Validation.”, Ultrasound In Med. & Biol., Vol. 27, No. 8, Pp. 1115–1125, 2001
- [45] F.M. Cardoso, M.C. Moraes, S.S. Furuie, “Realistic IVUS image generation in different intraluminal pressures.”, Ultrasound in Medicine and Biology, vol. 38, No. 12, pp. 2104-2119, Aug. 2012.
- [46] R. Fabbri, L. F. Costa, J. C Torlli, and O. M. Bruno, “2D Euclidean distance transforms: a comparative survey,” ACM Computing Surveys, vol. 40, Issue 1, pp. 12–24, Feb. 2008.
- [47] M.H. Xu, Y. Q. Liu, Q. L. Huang, Y. X. Chang, G. F. Luan, “An improved Dijkstra’s shortest path algorithm for sparse network.”, Applied Mathematics and Computation, vol.185, pp. 247-254, 2007.
- [48] S. Lobregt, M.A. Viergever, “A Discrete Dynamic Contour Model.”, IEEE Transactions on Medical Imaging, Vol. 14, No. 1. (1995), pp. 12-24

- [49] H. Zhu, K. Oakeson, M. Friedman, "Retrieval of Cardiac Phase from IVUS Sequences", *Ultrasonic Imaging and Signal Processing*, SPIE Medical Imaging, pp. 135-146, 2003
- [50] M. Rosales, P. Radeva, O. Rodriguez-Leor, D. Gil, "Modelling of image-catheter motion for 3-D IVUS.", *Medical Image Analysis* vol.13, pp. 91–104, 2009.
- [51] A. Arbab-Zadeh, A. deMaria, W. Penny, R. Russo, B. Kimura, V. Bhargava, "Axial movement of the intravascular ultrasound probe during the cardiac cycle: Implications for three-dimensional reconstruction and measurements of coronary dimensions.", *American Heart Journal* Nov 1999;138(5):865-72
- [52] R.O. Bude, R.S. Adler, "An easily made, low-cost, tissue-like ultrasound phantom material", *J Clin Ultrasound* 23:271-273, May 1995
- [53] <http://www.analogicultrasound.com/ultrasonix/systems/sonixtouch>

11. Appendix

11.1 Finite Elements Method Formulation

We present here the equations utilized to perform the FEM with linear tetrahedral elements.

Let $\{x_i, y_i, z_i\}; i = 1,2,3,4$ be the spatial coordinates of the four nodes of a tetrahedral element. The element stiffness matrix is given by

$$\text{Eq. 104} \quad \kappa = VB^TDB$$

where V is the element volume

$$\text{Eq. 105} \quad V = \frac{1}{6} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}$$

and the matrix D is given by

$$\text{Eq. 106} \quad D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

where E is the modulus of elasticity and ν is the Poisson's ratio.

Let N_1, N_2, N_3 and N_4 be the shape functions

$$\text{Eq. 107} \quad \begin{cases} N_1 = \frac{1}{6V} (\alpha_1 + \beta_1 x + \gamma_1 y + \delta_1 z) \\ N_2 = \frac{1}{6V} (\alpha_2 + \beta_2 x + \gamma_2 y + \delta_2 z) \\ N_3 = \frac{1}{6V} (\alpha_3 + \beta_3 x + \gamma_3 y + \delta_3 z) \\ N_4 = \frac{1}{6V} (\alpha_4 + \beta_4 x + \gamma_4 y + \delta_4 z) \end{cases}$$

where

$$\alpha_1 = \begin{bmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix} \quad \alpha_2 = - \begin{bmatrix} x_1 & y_1 & z_1 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix} \quad \alpha_3 = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_4 & y_4 & z_4 \end{bmatrix} \quad \alpha_4 = - \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

$$\beta_1 = - \begin{bmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{bmatrix} \quad \beta_2 = \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{bmatrix} \quad \beta_3 = - \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_4 & z_4 \end{bmatrix} \quad \beta_4 = \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{bmatrix}$$

$$\gamma_1 = \begin{bmatrix} 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{bmatrix} \quad \gamma_2 = - \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{bmatrix} \quad \gamma_3 = \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_4 & z_4 \end{bmatrix} \quad \gamma_4 = - \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{bmatrix}$$

$$\delta_1 = - \begin{bmatrix} 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{bmatrix} \quad \delta_2 = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{bmatrix} \quad \delta_3 = - \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_4 & y_4 \end{bmatrix} \quad \delta_4 = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$$

Then, the matrix B is given by

$$\text{Eq. 108} \quad B = \begin{bmatrix} \frac{\partial N_1}{\delta x} & 0 & 0 & \frac{\partial N_2}{\delta x} & 0 & 0 & \frac{\partial N_3}{\delta x} & 0 & 0 & \frac{\partial N_4}{\delta x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\delta y} & 0 & 0 & \frac{\partial N_2}{\delta y} & 0 & 0 & \frac{\partial N_3}{\delta y} & 0 & 0 & \frac{\partial N_4}{\delta y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\delta z} & 0 & 0 & \frac{\partial N_2}{\delta z} & 0 & 0 & \frac{\partial N_3}{\delta z} & 0 & 0 & \frac{\partial N_4}{\delta z} \\ \frac{\partial N_1}{\delta y} & \frac{\partial N_1}{\delta x} & 0 & \frac{\partial N_2}{\delta y} & \frac{\partial N_2}{\delta x} & 0 & \frac{\partial N_3}{\delta y} & \frac{\partial N_3}{\delta x} & 0 & \frac{\partial N_4}{\delta y} & \frac{\partial N_4}{\delta x} & 0 \\ 0 & \frac{\partial N_1}{\delta z} & \frac{\partial N_1}{\delta y} & 0 & \frac{\partial N_2}{\delta z} & \frac{\partial N_2}{\delta y} & 0 & \frac{\partial N_3}{\delta z} & \frac{\partial N_3}{\delta y} & 0 & \frac{\partial N_4}{\delta z} & \frac{\partial N_4}{\delta y} \\ \frac{\partial N_1}{\delta z} & 0 & \frac{\partial N_1}{\delta x} & \frac{\partial N_2}{\delta z} & 0 & \frac{\partial N_2}{\delta x} & \frac{\partial N_3}{\delta z} & 0 & \frac{\partial N_3}{\delta x} & \frac{\partial N_4}{\delta z} & 0 & \frac{\partial N_4}{\delta x} \end{bmatrix}$$

$$\text{Eq. 109} \quad B = \begin{bmatrix} \beta_1 & 0 & 0 & \beta_2 & 0 & 0 & \beta_3 & 0 & 0 & \beta_4 & 0 & 0 \\ 0 & \gamma_1 & 0 & 0 & \gamma_2 & 0 & 0 & \gamma_3 & 0 & 0 & \gamma_4 & 0 \\ 0 & 0 & \delta_1 & 0 & 0 & \delta_2 & 0 & 0 & \delta_3 & 0 & 0 & \delta_4 \\ \gamma_1 & \beta_1 & 0 & \gamma_2 & \beta_2 & 0 & \gamma_3 & \beta_3 & 0 & \gamma_4 & \beta_4 & 0 \\ 0 & \delta_1 & \gamma_1 & 0 & \delta_2 & \gamma_2 & 0 & \delta_3 & \gamma_3 & 0 & \delta_4 & \gamma_4 \\ \delta_1 & 0 & \beta_1 & \delta_2 & 0 & \beta_2 & \delta_3 & 0 & \beta_3 & \delta_4 & 0 & \beta_4 \end{bmatrix} \frac{1}{6V}$$

It is easy to realize that the size of κ is 12×12 , which corresponds to 3 degrees of freedom of each of the four nodes. Analogously, the global stiffness matrix K size is $3n \times 3n$, where n is the number of nodes in the entire mesh.

K is built by including the values of the element stiffness matrix in the lines and columns that are related to each node number and direction. Figure 69 illustrates a mesh with two elements and Figure 70 shows the element stiffness matrix of each element.

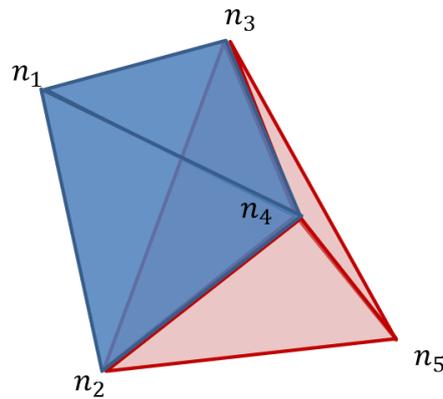


Figure 69: Tetrahedral mesh with two elements.

Let κ_A and κ_B be the element stiffness matrix of the blue and red tetrahedra, respectively. Figure 70 shows the construction of the K from κ_A and κ_B . It is important to notice that each element of K , κ_A and κ_B represents one node in one direction. For example, in Figure 70, $K_{7,7}$ represents the node 3 in x direction, hence it is the sum of $a_{7,7}$ and $b_{4,4}$, which are the positions related to the same node and direction.

A	1.x1	2.y1	3.z1	4.x2	5.y2	6.z2	7.x3	8.y3	9.z3	10.x4	11.y4	12.z4
1.x1	a _{1,1}	a _{1,2}	a _{1,3}	a _{1,4}	a _{1,5}	a _{1,6}	a _{1,7}	a _{1,8}	a _{1,9}	a _{1,10}	a _{1,11}	a _{1,12}
2.y1	a _{2,1}	a _{2,2}	a _{2,3}	a _{2,4}	a _{2,5}	a _{2,6}	a _{2,7}	a _{2,8}	a _{2,9}	a _{2,10}	a _{2,11}	a _{2,12}
3.z1	a _{3,1}	a _{3,2}	a _{3,3}	a _{3,4}	a _{3,5}	a _{3,6}	a _{3,7}	a _{3,8}	a _{3,9}	a _{3,10}	a _{3,11}	a _{3,12}
4.x2	a _{4,1}	a _{4,2}	a _{4,3}	a _{4,4}	a _{4,5}	a _{4,6}	a _{4,7}	a _{4,8}	a _{4,9}	a _{4,10}	a _{4,11}	a _{4,12}
5.y2	a _{5,1}	a _{5,2}	a _{5,3}	a _{5,4}	a _{5,5}	a _{5,6}	a _{5,7}	a _{5,8}	a _{5,9}	a _{5,10}	a _{5,11}	a _{5,12}
6.z2	a _{6,1}	a _{6,2}	a _{6,3}	a _{6,4}	a _{6,5}	a _{6,6}	a _{6,7}	a _{6,8}	a _{6,9}	a _{6,10}	a _{6,11}	a _{6,12}
7.x3	a _{7,1}	a _{7,2}	a _{7,3}	a _{7,4}	a _{7,5}	a _{7,6}	a _{7,7}	a _{7,8}	a _{7,9}	a _{7,10}	a _{7,11}	a _{7,12}
8.y3	a _{8,1}	a _{8,2}	a _{8,3}	a _{8,4}	a _{8,5}	a _{8,6}	a _{8,7}	a _{8,8}	a _{8,9}	a _{8,10}	a _{8,11}	a _{8,12}
9.z3	a _{9,1}	a _{9,2}	a _{9,3}	a _{9,4}	a _{9,5}	a _{9,6}	a _{9,7}	a _{9,8}	a _{9,9}	a _{9,10}	a _{9,11}	a _{9,12}
10.x4	a _{10,1}	a _{10,2}	a _{10,3}	a _{10,4}	a _{10,5}	a _{10,6}	a _{10,7}	a _{10,8}	a _{10,9}	a _{10,10}	a _{10,11}	a _{10,12}
11.y4	a _{11,1}	a _{11,2}	a _{11,3}	a _{11,4}	a _{11,5}	a _{11,6}	a _{11,7}	a _{11,8}	a _{11,9}	a _{11,10}	a _{11,11}	a _{11,12}
12.z4	a _{12,1}	a _{12,2}	a _{12,3}	a _{12,4}	a _{12,5}	a _{12,6}	a _{12,7}	a _{12,8}	a _{12,9}	a _{12,10}	a _{12,11}	a _{12,12}

B	1.x2	2.y2	3.z2	4.x3	5.y3	6.z3	7.x4	8.y4	9.z4	10.x5	11.y5	12.z5
1.x2	b _{1,1}	b _{1,2}	b _{1,3}	b _{1,4}	b _{1,5}	b _{1,6}	b _{1,7}	b _{1,8}	b _{1,9}	b _{1,10}	b _{1,11}	b _{1,12}
2.y2	b _{2,1}	b _{2,2}	b _{2,3}	b _{2,4}	b _{2,5}	b _{2,6}	b _{2,7}	b _{2,8}	b _{2,9}	b _{2,10}	b _{2,11}	b _{2,12}
3.z2	b _{3,1}	b _{3,2}	b _{3,3}	b _{3,4}	b _{3,5}	b _{3,6}	b _{3,7}	b _{3,8}	b _{3,9}	b _{3,10}	b _{3,11}	b _{3,12}
4.x3	b _{4,1}	b _{4,2}	b _{4,3}	b _{4,4}	b _{4,5}	b _{4,6}	b _{4,7}	b _{4,8}	b _{4,9}	b _{4,10}	b _{4,11}	b _{4,12}
5.y3	b _{5,1}	b _{5,2}	b _{5,3}	b _{5,4}	b _{5,5}	b _{5,6}	b _{5,7}	b _{5,8}	b _{5,9}	b _{5,10}	b _{5,11}	b _{5,12}
6.z3	b _{6,1}	b _{6,2}	b _{6,3}	b _{6,4}	b _{6,5}	b _{6,6}	b _{6,7}	b _{6,8}	b _{6,9}	b _{6,10}	b _{6,11}	b _{6,12}
7.x4	b _{7,1}	b _{7,2}	b _{7,3}	b _{7,4}	b _{7,5}	b _{7,6}	b _{7,7}	b _{7,8}	b _{7,9}	b _{7,10}	b _{7,11}	b _{7,12}
8.y4	b _{8,1}	b _{8,2}	b _{8,3}	b _{8,4}	b _{8,5}	b _{8,6}	b _{8,7}	b _{8,8}	b _{8,9}	b _{8,10}	b _{8,11}	b _{8,12}
9.z4	b _{9,1}	b _{9,2}	b _{9,3}	b _{9,4}	b _{9,5}	b _{9,6}	b _{9,7}	b _{9,8}	b _{9,9}	b _{9,10}	b _{9,11}	b _{9,12}
10.x5	b _{10,1}	b _{10,2}	b _{10,3}	b _{10,4}	b _{10,5}	b _{10,6}	b _{10,7}	b _{10,8}	b _{10,9}	b _{10,10}	b _{10,11}	b _{10,12}
11.y5	b _{11,1}	b _{11,2}	b _{11,3}	b _{11,4}	b _{11,5}	b _{11,6}	b _{11,7}	b _{11,8}	b _{11,9}	b _{11,10}	b _{11,11}	b _{11,12}
12.z5	b _{12,1}	b _{12,2}	b _{12,3}	b _{12,4}	b _{12,5}	b _{12,6}	b _{12,7}	b _{12,8}	b _{12,9}	b _{12,10}	b _{12,11}	b _{12,12}

Figure 70: Illustration of the construction of the global stiffness matrix from two element stiffness matrix

A+B	1.x1	2.y1	3.z1	4.x2	5.y2	6.z2	7.x3	8.y3	9.z3	10.x4	11.y4	12.z4	13.x5	14.y5	15.z5
1.x1	a _{1,1}	a _{1,2}	a _{1,3}	a _{1,4}	a _{1,5}	a _{1,6}	a _{1,7}	a _{1,8}	a _{1,9}	a _{1,10}	a _{1,11}	a _{1,12}			
2.y1	a _{2,1}	a _{2,2}	a _{2,3}	a _{2,4}	a _{2,5}	a _{2,6}	a _{2,7}	a _{2,8}	a _{2,9}	a _{2,10}	a _{2,11}	a _{2,12}			
3.z1	a _{3,1}	a _{3,2}	a _{3,3}	a _{3,4}	a _{3,5}	a _{3,6}	a _{3,7}	a _{3,8}	a _{3,9}	a _{3,10}	a _{3,11}	a _{3,12}			
4.x2	a _{4,1}	a _{4,2}	a _{4,3}	a _{4,4} +b _{1,1}	a _{4,5} +b _{1,2}	a _{4,6} +b _{1,3}	a _{4,7} +b _{1,4}	a _{4,8} +b _{1,5}	a _{4,9} +b _{1,6}	a _{4,10} +b _{1,7}	a _{4,11} +b _{1,8}	a _{4,12} +b _{1,9}	b _{1,10}	b _{1,11}	b _{1,12}
5.y2	a _{5,1}	a _{5,2}	a _{5,3}	a _{5,4} +b _{2,1}	a _{5,5} +b _{2,2}	a _{5,6} +b _{2,3}	a _{5,7} +b _{2,4}	a _{5,8} +b _{2,5}	a _{5,9} +b _{2,6}	a _{5,10} +b _{2,7}	a _{5,11} +b _{2,8}	a _{5,12} +b _{2,9}	b _{2,10}	b _{2,11}	b _{2,12}
6.z2	a _{6,1}	a _{6,2}	a _{6,3}	a _{6,4} +b _{3,1}	a _{6,5} +b _{3,2}	a _{6,6} +b _{3,3}	a _{6,7} +b _{3,4}	a _{6,8} +b _{3,5}	a _{6,9} +b _{3,6}	a _{6,10} +b _{3,7}	a _{6,11} +b _{3,8}	a _{6,12} +b _{3,9}	b _{3,10}	b _{3,11}	b _{3,12}
7.x3	a _{7,1}	a _{7,2}	a _{7,3}	a _{7,4} +b _{4,1}	a _{7,5} +b _{4,2}	a _{7,6} +b _{4,3}	a _{7,7} +b _{4,4}	a _{7,8} +b _{4,5}	a _{7,9} +b _{4,6}	a _{7,10} +b _{4,7}	a _{7,11} +b _{4,8}	a _{7,12} +b _{4,9}	b _{4,10}	b _{4,11}	b _{4,12}
8.y3	a _{8,1}	a _{8,2}	a _{8,3}	a _{8,4} +b _{5,1}	a _{8,5} +b _{5,2}	a _{8,6} +b _{5,3}	a _{8,7} +b _{5,4}	a _{8,8} +b _{5,5}	a _{8,9} +b _{5,6}	a _{8,10} +b _{5,7}	a _{8,11} +b _{5,8}	a _{8,12} +b _{5,9}	b _{5,10}	b _{5,11}	b _{5,12}
9.z3	a _{9,1}	a _{9,2}	a _{9,3}	a _{9,4} +b _{6,1}	a _{9,5} +b _{6,2}	a _{9,6} +b _{6,3}	a _{9,7} +b _{6,4}	a _{9,8} +b _{6,5}	a _{9,9} +b _{6,6}	a _{9,10} +b _{6,7}	a _{9,11} +b _{6,8}	a _{9,12} +b _{6,9}	b _{6,10}	b _{6,11}	b _{6,12}
10.x4	a _{10,1}	a _{10,2}	a _{10,3}	a _{10,4} +b _{7,1}	a _{10,5} +b _{7,2}	a _{10,6} +b _{7,3}	a _{10,7} +b _{7,4}	a _{10,8} +b _{7,5}	a _{10,9} +b _{7,6}	a _{10,10} +b _{7,7}	a _{10,11} +b _{7,8}	a _{10,12} +b _{7,9}	b _{7,10}	b _{7,11}	b _{7,12}
11.y4	a _{11,1}	a _{11,2}	a _{11,3}	a _{11,4} +b _{8,1}	a _{11,5} +b _{8,2}	a _{11,6} +b _{8,3}	a _{11,7} +b _{8,4}	a _{11,8} +b _{8,5}	a _{11,9} +b _{8,6}	a _{11,10} +b _{8,7}	a _{11,11} +b _{8,8}	a _{11,12} +b _{8,9}	b _{8,10}	b _{8,11}	b _{8,12}
12.z4	a _{12,1}	a _{12,2}	a _{12,3}	a _{12,4} +b _{9,1}	a _{12,5} +b _{9,2}	a _{12,6} +b _{9,3}	a _{12,7} +b _{9,4}	a _{12,8} +b _{9,5}	a _{12,9} +b _{9,6}	a _{12,10} +b _{9,7}	a _{12,11} +b _{9,8}	a _{12,12} +b _{9,9}	b _{9,10}	b _{9,11}	b _{9,12}
13.x5				b _{10,1}	b _{10,2}	b _{10,3}	b _{10,4}	b _{10,5}	b _{10,6}	b _{10,7}	b _{10,8}	b _{10,9}	b _{10,10}	b _{10,11}	b _{10,12}
14.y5				b _{11,1}	b _{11,2}	b _{11,3}	b _{11,4}	b _{11,5}	b _{11,6}	b _{11,7}	b _{11,8}	b _{11,9}	b _{11,10}	b _{11,11}	b _{11,12}
15.z5				b _{12,1}	b _{12,2}	b _{12,3}	b _{12,4}	b _{12,5}	b _{12,6}	b _{12,7}	b _{12,8}	b _{12,9}	b _{12,10}	b _{12,11}	b _{12,12}

After the construction of K , the software automatically detects the faces of the tetrahedra at the interface of the tissue with the probe, in the case of transcutaneous US, or the interface of the tissue with the lumen, in the case of IVUS. Then, the forces are distributed throughout the nodes that belong to those faces. The direction of the forces is perpendicular to the faces (Figure 32) and the intensity is defined by the user (Figure 25). The result is a vector F with $3n$ elements, in which each element represents a node in one of the three directions.

Next, the algorithm detects the nodes that belong to the boundary condition. Those nodes are not able to move in any direction. In the case of transcutaneous US, the boundary conditions are the nodes that belong to the Volume face that is distal to the face that is in contact with the probe. In the case of IVUS, the boundary conditions are the nodes at the outermost surface of the coronary phantom.

Then, matrix K reduction is performed, where the rows and columns related to the nodes and directions of the boundary conditions are eliminated Figure 71. The same procedure is applied in the vector F .

After obtaining K_{red} and F_{red} from K and F , respectively, the displacement of the nodes of the mesh is calculated as follows

Eq. 110
$$U = K_{red} \setminus F_{red}$$

Finally, the displacement of each node is added to its initial position in order to obtain the deformed mesh.

K	$1.x_1$	$2.y_1$	$3.z_1$	$4.x_2$	$5.y_2$	$6.z_2$	$7.x_3$	$8.y_3$	$9.z_3$	$10.x_4$	$11.y_4$	$12.z_4$	$13.x_5$	$14.y_5$	$15.z_5$
$1.x_1$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$	$a_{1,8}$	$a_{1,9}$	$a_{1,10}$	$a_{1,11}$	$a_{1,12}$			
$2.y_1$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$	$a_{2,8}$	$a_{2,9}$	$a_{2,10}$	$a_{2,11}$	$a_{2,12}$			
$3.z_1$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$	$a_{3,8}$	$a_{3,9}$	$a_{3,10}$	$a_{3,11}$	$a_{3,12}$			
$4.x_2$	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}+b_{1,1}$	$a_{4,5}+b_{1,2}$	$a_{4,6}+b_{1,3}$	$a_{4,7}+b_{1,4}$	$a_{4,8}+b_{1,5}$	$a_{4,9}+b_{1,6}$	$a_{4,10}+b_{1,7}$	$a_{4,11}+b_{1,8}$	$a_{4,12}+b_{1,9}$	$b_{1,10}$	$b_{1,11}$	$b_{1,12}$
$5.y_2$	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}+b_{2,1}$	$a_{5,5}+b_{2,2}$	$a_{5,6}+b_{2,3}$	$a_{5,7}+b_{2,4}$	$a_{5,8}+b_{2,5}$	$a_{5,9}+b_{2,6}$	$a_{5,10}+b_{2,7}$	$a_{5,11}+b_{2,8}$	$a_{5,12}+b_{2,9}$	$b_{2,10}$	$b_{2,11}$	$b_{2,12}$
$6.z_2$	$a_{6,1}$	$a_{6,2}$	$a_{6,3}$	$a_{6,4}+b_{3,1}$	$a_{6,5}+b_{3,2}$	$a_{6,6}+b_{3,3}$	$a_{6,7}+b_{3,4}$	$a_{6,8}+b_{3,5}$	$a_{6,9}+b_{3,6}$	$a_{6,10}+b_{3,7}$	$a_{6,11}+b_{3,8}$	$a_{6,12}+b_{3,9}$	$b_{3,10}$	$b_{3,11}$	$b_{3,12}$
$7.x_3$	$a_{7,1}$	$a_{7,2}$	$a_{7,3}$	$a_{7,4}+b_{4,1}$	$a_{7,5}+b_{4,2}$	$a_{7,6}+b_{4,3}$	$a_{7,7}+b_{4,4}$	$a_{7,8}+b_{4,5}$	$a_{7,9}+b_{4,6}$	$a_{7,10}+b_{4,7}$	$a_{7,11}+b_{4,8}$	$a_{7,12}+b_{4,9}$	$b_{4,10}$	$b_{4,11}$	$b_{4,12}$
$8.y_3$	$a_{8,1}$	$a_{8,2}$	$a_{8,3}$	$a_{8,4}+b_{5,1}$	$a_{8,5}+b_{5,2}$	$a_{8,6}+b_{5,3}$	$a_{8,7}+b_{5,4}$	$a_{8,8}+b_{5,5}$	$a_{8,9}+b_{5,6}$	$a_{8,10}+b_{5,7}$	$a_{8,11}+b_{5,8}$	$a_{8,12}+b_{5,9}$	$b_{5,10}$	$b_{5,11}$	$b_{5,12}$
$9.z_3$	$a_{9,1}$	$a_{9,2}$	$a_{9,3}$	$a_{9,4}+b_{6,1}$	$a_{9,5}+b_{6,2}$	$a_{9,6}+b_{6,3}$	$a_{9,7}+b_{6,4}$	$a_{9,8}+b_{6,5}$	$a_{9,9}+b_{6,6}$	$a_{9,10}+b_{6,7}$	$a_{9,11}+b_{6,8}$	$a_{9,12}+b_{6,9}$	$b_{6,10}$	$b_{6,11}$	$b_{6,12}$
$10.x_4$	$a_{10,1}$	$a_{10,2}$	$a_{10,3}$	$a_{10,4}+b_{7,1}$	$a_{10,5}+b_{7,2}$	$a_{10,6}+b_{7,3}$	$a_{10,7}+b_{7,4}$	$a_{10,8}+b_{7,5}$	$a_{10,9}+b_{7,6}$	$a_{10,10}+b_{7,7}$	$a_{10,11}+b_{7,8}$	$a_{10,12}+b_{7,9}$	$b_{7,10}$	$b_{7,11}$	$b_{7,12}$
$11.y_4$	$a_{11,1}$	$a_{11,2}$	$a_{11,3}$	$a_{11,4}+b_{8,1}$	$a_{11,5}+b_{8,2}$	$a_{11,6}+b_{8,3}$	$a_{11,7}+b_{8,4}$	$a_{11,8}+b_{8,5}$	$a_{11,9}+b_{8,6}$	$a_{11,10}+b_{8,7}$	$a_{11,11}+b_{8,8}$	$a_{11,12}+b_{8,9}$	$b_{8,10}$	$b_{8,11}$	$b_{8,12}$
$12.z_4$	$a_{12,1}$	$a_{12,2}$	$a_{12,3}$	$a_{12,4}+b_{9,1}$	$a_{12,5}+b_{9,2}$	$a_{12,6}+b_{9,3}$	$a_{12,7}+b_{9,4}$	$a_{12,8}+b_{9,5}$	$a_{12,9}+b_{9,6}$	$a_{12,10}+b_{9,7}$	$a_{12,11}+b_{9,8}$	$a_{12,12}+b_{9,9}$	$b_{9,10}$	$b_{9,11}$	$b_{9,12}$
$13.x_5$				$b_{10,1}$	$b_{10,2}$	$b_{10,3}$	$b_{10,4}$	$b_{10,5}$	$b_{10,6}$	$b_{10,7}$	$b_{10,8}$	$b_{10,9}$	$b_{10,10}$	$b_{10,11}$	$b_{10,12}$
$14.y_5$				$b_{11,1}$	$b_{11,2}$	$b_{11,3}$	$b_{11,4}$	$b_{11,5}$	$b_{11,6}$	$b_{11,7}$	$b_{11,8}$	$b_{11,9}$	$b_{11,10}$	$b_{11,11}$	$b_{11,12}$
$15.z_5$				$b_{12,1}$	$b_{12,2}$	$b_{12,3}$	$b_{12,4}$	$b_{12,5}$	$b_{12,6}$	$b_{12,7}$	$b_{12,8}$	$b_{12,9}$	$b_{12,10}$	$b_{12,11}$	$b_{12,12}$

K_{red}	$1.x_1$	$2.y_1$	$3.z_1$	$4.x_2$	$5.y_2$	$6.z_2$	$7.x_3$	$8.y_3$	$9.z_3$	$10.x_5$	$11.y_5$	$12.z_5$
$1.x_1$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$	$a_{1,8}$	$a_{1,9}$			
$2.y_1$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$	$a_{2,8}$	$a_{2,9}$			
$3.z_1$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$	$a_{3,8}$	$a_{3,9}$			
$4.x_2$	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}+b_{1,1}$	$a_{4,5}+b_{1,2}$	$a_{4,6}+b_{1,3}$	$a_{4,7}+b_{1,4}$	$a_{4,8}+b_{1,5}$	$a_{4,9}+b_{1,6}$	$b_{1,10}$	$b_{1,11}$	$b_{1,12}$
$5.y_2$	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}+b_{2,1}$	$a_{5,5}+b_{2,2}$	$a_{5,6}+b_{2,3}$	$a_{5,7}+b_{2,4}$	$a_{5,8}+b_{2,5}$	$a_{5,9}+b_{2,6}$	$b_{2,10}$	$b_{2,11}$	$b_{2,12}$
$6.z_2$	$a_{6,1}$	$a_{6,2}$	$a_{6,3}$	$a_{6,4}+b_{3,1}$	$a_{6,5}+b_{3,2}$	$a_{6,6}+b_{3,3}$	$a_{6,7}+b_{3,4}$	$a_{6,8}+b_{3,5}$	$a_{6,9}+b_{3,6}$	$b_{3,10}$	$b_{3,11}$	$b_{3,12}$
$7.x_3$	$a_{7,1}$	$a_{7,2}$	$a_{7,3}$	$a_{7,4}+b_{4,1}$	$a_{7,5}+b_{4,2}$	$a_{7,6}+b_{4,3}$	$a_{7,7}+b_{4,4}$	$a_{7,8}+b_{4,5}$	$a_{7,9}+b_{4,6}$	$b_{4,10}$	$b_{4,11}$	$b_{4,12}$
$8.y_3$	$a_{8,1}$	$a_{8,2}$	$a_{8,3}$	$a_{8,4}+b_{5,1}$	$a_{8,5}+b_{5,2}$	$a_{8,6}+b_{5,3}$	$a_{8,7}+b_{5,4}$	$a_{8,8}+b_{5,5}$	$a_{8,9}+b_{5,6}$	$b_{5,10}$	$b_{5,11}$	$b_{5,12}$
$9.z_3$	$a_{9,1}$	$a_{9,2}$	$a_{9,3}$	$a_{9,4}+b_{6,1}$	$a_{9,5}+b_{6,2}$	$a_{9,6}+b_{6,3}$	$a_{9,7}+b_{6,4}$	$a_{9,8}+b_{6,5}$	$a_{9,9}+b_{6,6}$	$b_{6,10}$	$b_{6,11}$	$b_{6,12}$
$10.x_5$				$b_{10,1}$	$b_{10,2}$	$b_{10,3}$	$b_{10,4}$	$b_{10,5}$	$b_{10,6}$	$b_{10,10}$	$b_{10,11}$	$b_{10,12}$
$11.y_5$				$b_{11,1}$	$b_{11,2}$	$b_{11,3}$	$b_{11,4}$	$b_{11,5}$	$b_{11,6}$	$b_{11,10}$	$b_{11,11}$	$b_{11,12}$
$12.z_5$				$b_{12,1}$	$b_{12,2}$	$b_{12,3}$	$b_{12,4}$	$b_{12,5}$	$b_{12,6}$	$b_{12,10}$	$b_{12,11}$	$b_{12,12}$

Figure 71: Illustration of the matrix reduction.

11.2 IVUS Simulation: 2D phantoms

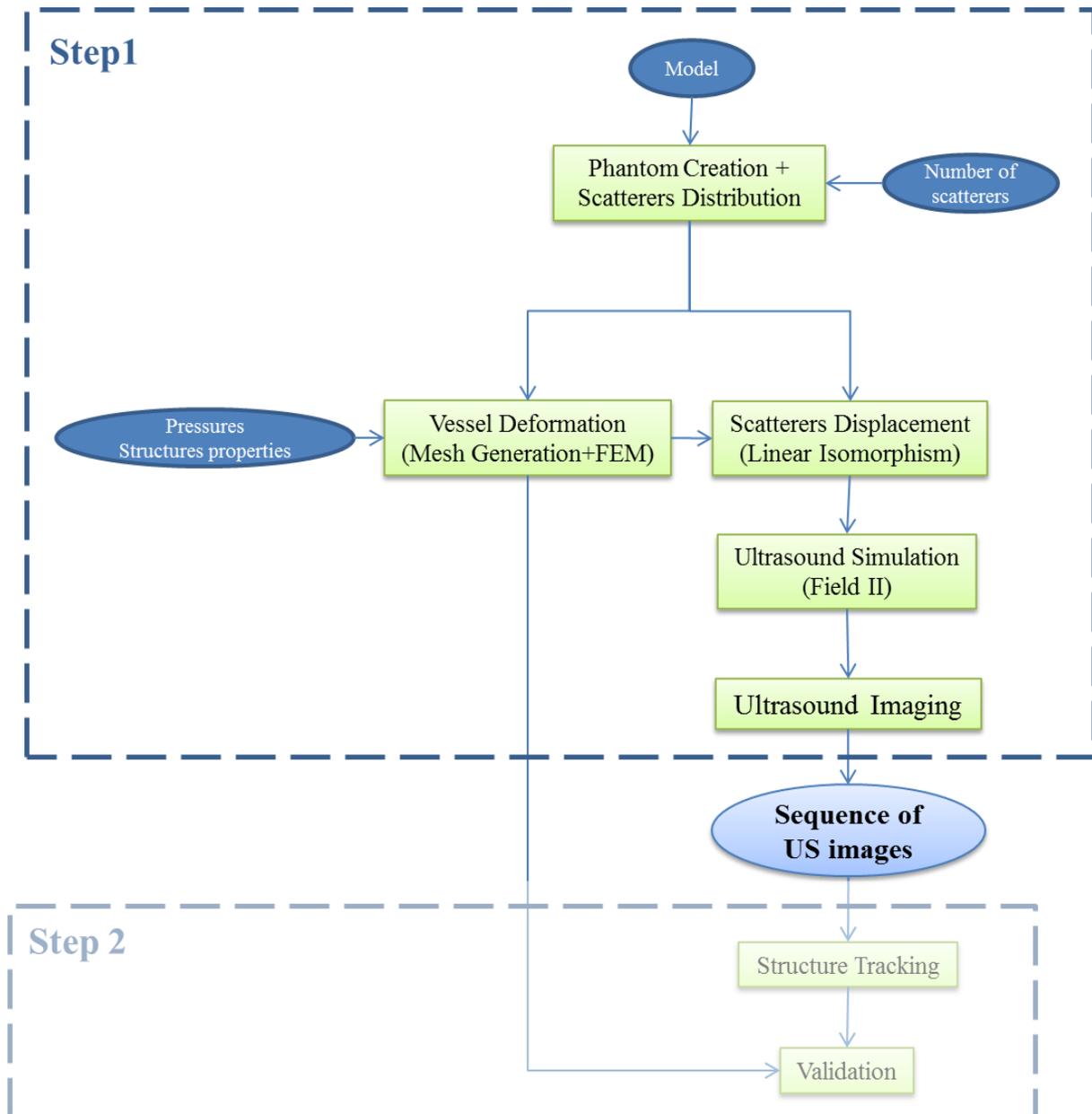


Figure 72: Block Diagram. Step 1 depicts the procedure to obtain the simulated IVUS images sequence. Step 2 illustrates the sequence for the elastography estimation

Firstly, we created numeric phantom in order to simulate a coronary cross sectional view. Then we applied FEM in order to simulate its deformation during a cardiac cycle. Next, we included speckle noise to simulate IVUS images.

Analogously to UltraSSim (Section 5), we have created a software that creates IVUS numeric phantoms generating simulated ultrasound images (Step 1 of Figure 72). The software is called IVUSSim (IntraVascular UltraSound Simulation) [45] and is available online [35] free from

charges for educational and research purposes. More details about the 2D IVUS simulation can be obtained in [45]. The sequence in Step 2 of Figure 72 is explained in section 6.

11.2.1 Coronary simulation

We created a numeric phantom that simulates the cross section slice of the blood vessel (Figure 73). The model consists on an 8-bit image with 400x400 pixels and contains lumen, intima/media, adventitia and lesion (Table 22). The lesion properties may vary according to its type: lipidic, fibrous or calcified.

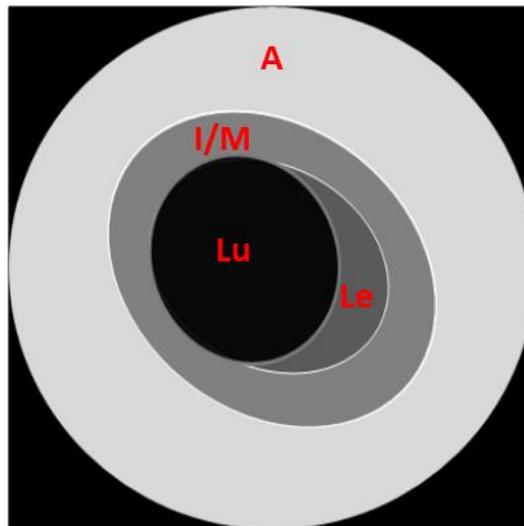


Figure 73: Numeric phantom that simulates the vessel cross-section. A: adventitia, I/M: intima/media, Lu: lumen and Le: lesion

Table 22: Parameters of finite elements method and brightness

STRUCTURE	ELASTICITY (kPa)	BRIGHTNESS (in relation to the calcified lesion)
Lumen	0	5%
Intima/media	600	50%
Lipidic lesion	25	25%
Calcified lesion	1200	100%

The user is able to control the tissue mean brightness through the scatterers density and scattering amplitude of the tissue, as explained in Section 5.8.2.

11.2.2 Cardiac Movements Simulation

We have simulated the cardiac compression through finite elements method (FEM), in order to get the static compressed image for several pressure values.

First, we create a gold-standard image (Figure 74a) and obtained its border pixels (Figure 74b), which were considered the initial set of nodes. Then, using the set of nodes, we created a triangular mesh using Delaunay algorithm (Figure 74c).

The nodes at the outer border of the vessel constituted the boundary conditions and, therefore, fixed nodes.

The pressure forces were applied at the nodes located at lumen border and their directions were perpendicular to the lumen border line (Figure 74d)

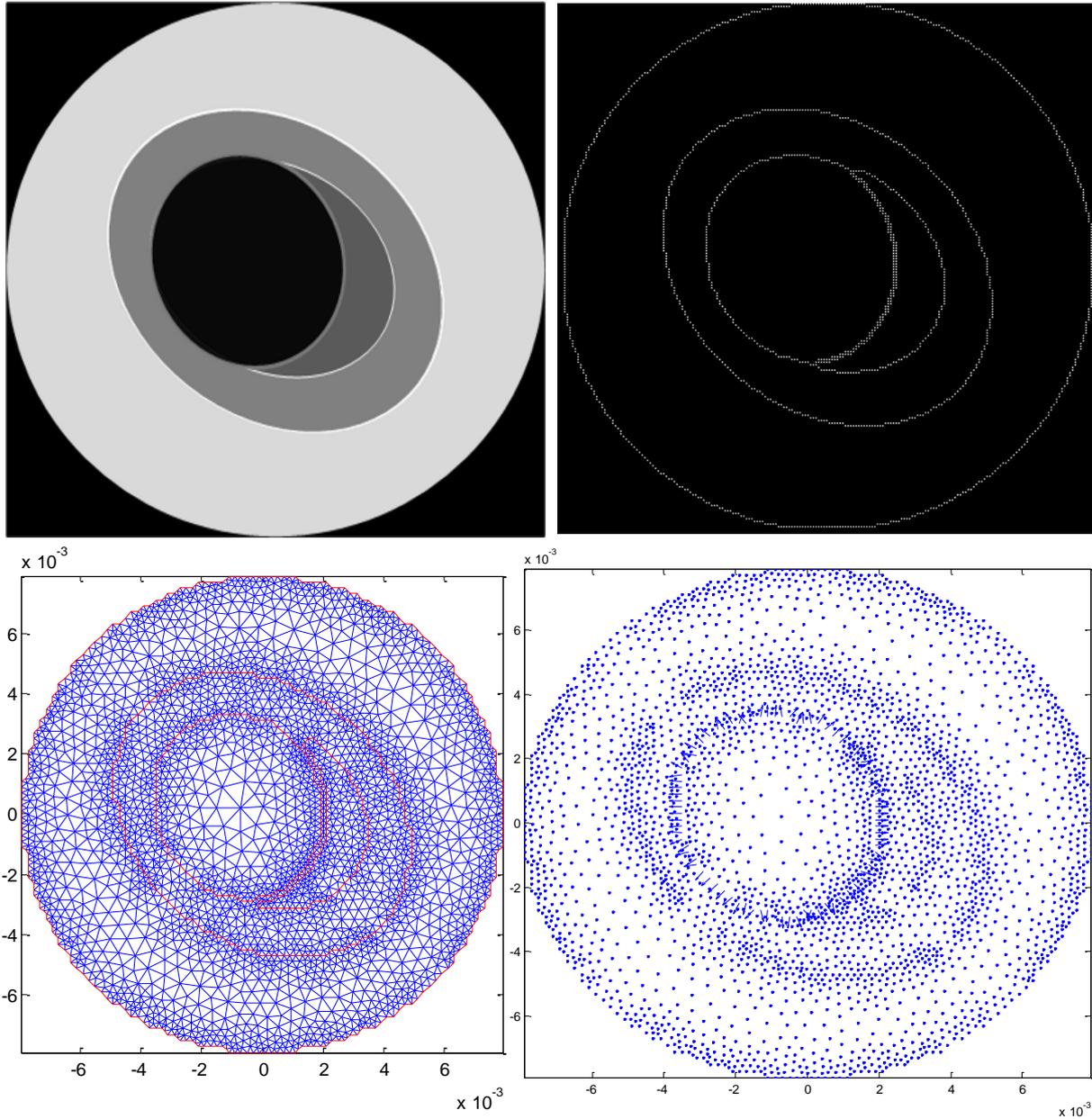


Figure 74: a) gold-standard image, b) image borders (initial set of nodes), c) mesh creation, d) nodes and pressure forces.

After defining the boundary conditions and the applied forces, FEM was applied to perform the blood vessel transformation. The deformed mesh and gold-standard images can be observed in Figure 75.

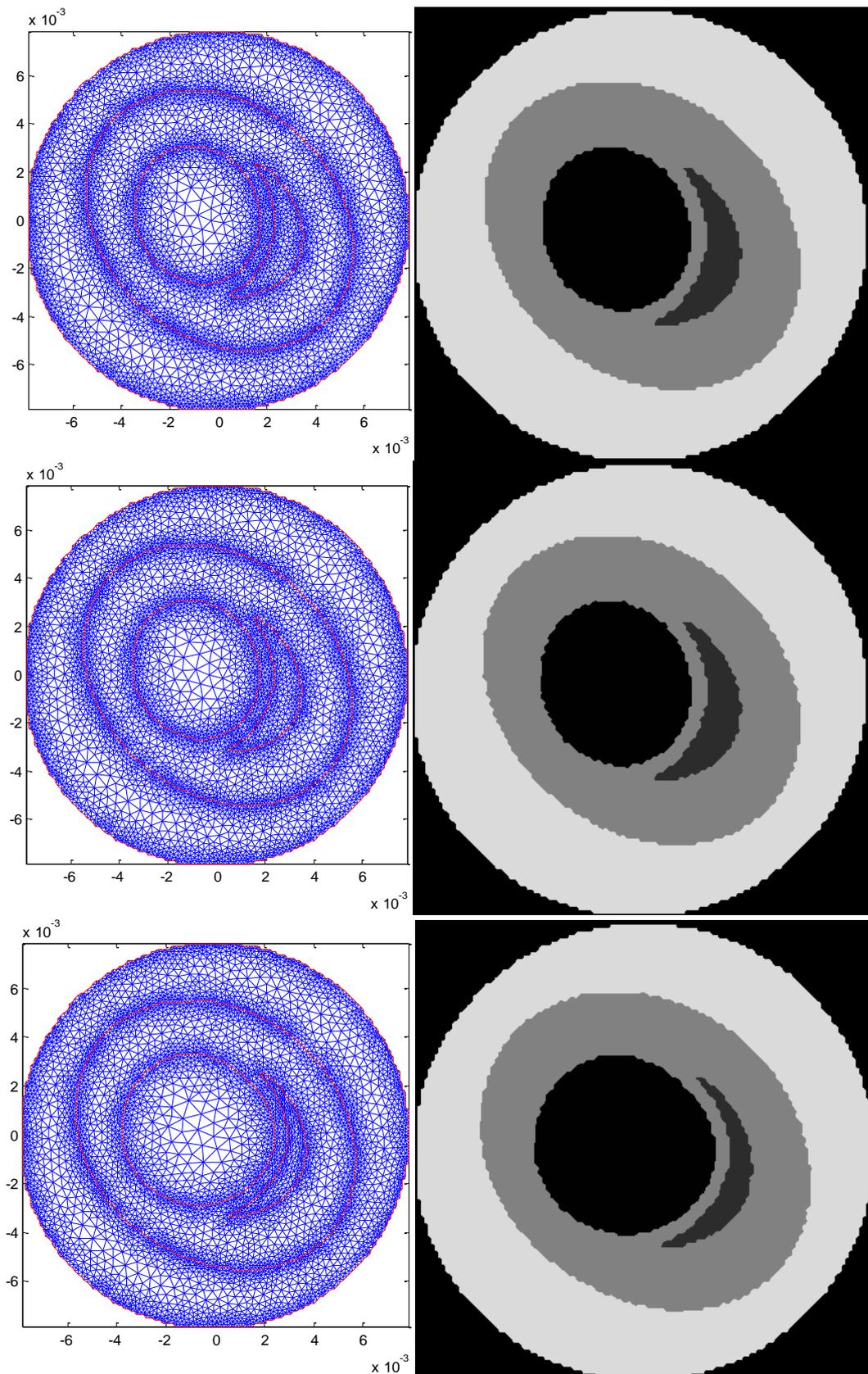


Figure 75: Mesh (left) and gold standard images (right) with initial pressure (top), after incremental pressure of 5.33kPa (middle) and 100kPa (bottom).

11.2.3 Scatterers Movement

After the deformation, the scatterers were distributed and displaced as explained in Section 5.3. Since, in Field II, the scatterers must be distributed in a 3D space, we projected the 2D gold-standard in the third dimension to a length greater than the US transducer in the longitudinal direction, e.g. 6 mm.

11.2.4 Speckle Noise Simulation

11.2.4.1 Scatterers positioning

For each frame, a set of 256 ultrasound waves are emitted perpendicularly to the catheter tip. The ultrasound waves are distributed radially and uniformly, as explained in Section 1.2.2.1.

Instead of emitting ultrasound waves in different directions, we kept the catheter transducer stationary and rotated the scatterers around it (Figure 76). Let $\{x, y, z\} = \{0, 0, 0\}$ be the catheter tip position and x be the catheter longitudinal axis. The scatterers rotation is performed as follows.

$$\text{Eq. 111} \quad \begin{bmatrix} x_i^\circ \\ y_i^\circ \\ z_i^\circ \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \cdot \cos(\theta) - z_i \cdot \sin(\theta) \\ y_i \cdot \sin(\theta) + z_i \cdot \cos(\theta) \end{bmatrix}, i = 1, 2, 3, \dots, N$$

where N is the number of frames (different deformations) and θ is the angle of rotation that varies in steps of $2\pi/256$.

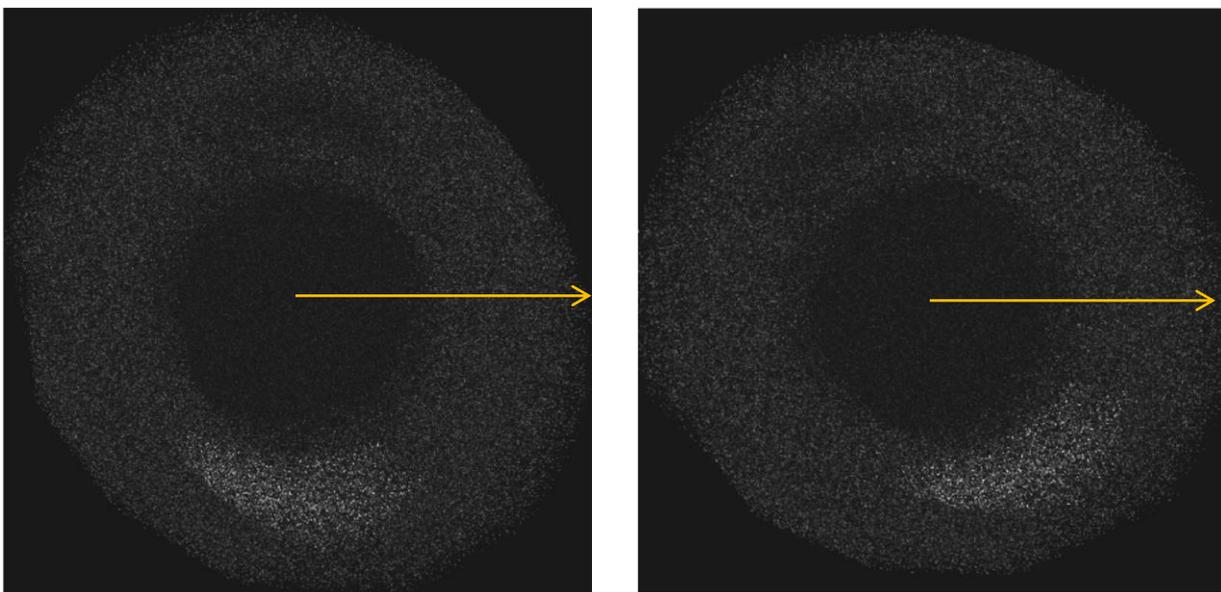


Figure 76: Scatterers imaging. Initial position (left) and after rotation of 30° CCW (right). The yellow arrow represents the direction of the emitting beam.

11.2.4.2 Ultrasound simulation – RF data acquisition

Firstly, a transducer must be synthesized. In this work, we utilized two different catheter systems: mechanical and solid state (Table 23).

The solid-state system consists of 64 transducers evenly distributed throughout the catheter tip perimeter. Each ultrasound beam was synthesized using a group of 15 consecutive elements.

Table 23: Parameters of ultrasound probe simulation

Parameter	Symbol	Mechanical	Solid State
Number of elements	-	1	64
Active elements during emission	-	1	1
Active elements during reception	-	1	15
Catheter diameter	-	1mm	1mm
Transducer size (WxH)	-	750x2250 μ m	46x700 μ m
Distance between transducers	-	-	3 μ m
IVUS resolution	Res_{IVUS}	80 μ m/pixel	40 μ m/pixel
Ultrasound center frequency	f_0	40 MHz	20 MHz
Sampling frequency	f_s	400 MHz	200MHz
Ultrasound speed	C	1540 m/s	1540 m/s

Both transducer emit 256 equiangularly distributed beams. The ultrasound beam consisted on a two-cycle sinusoidal wave weighted with a Hanning window (as in Section 5.4 - Figure 34c).

11.2.4.3 B-mode image formation

After defining the catheter properties and performing Field II, the set of signals was acquired. Then, in order to obtain the B-mode vectors, the signal envelope was obtained and log-compression was applied, as explained in Section 5.6. Figure 77 shows the polar image, which is formed by the stack of the vectors.

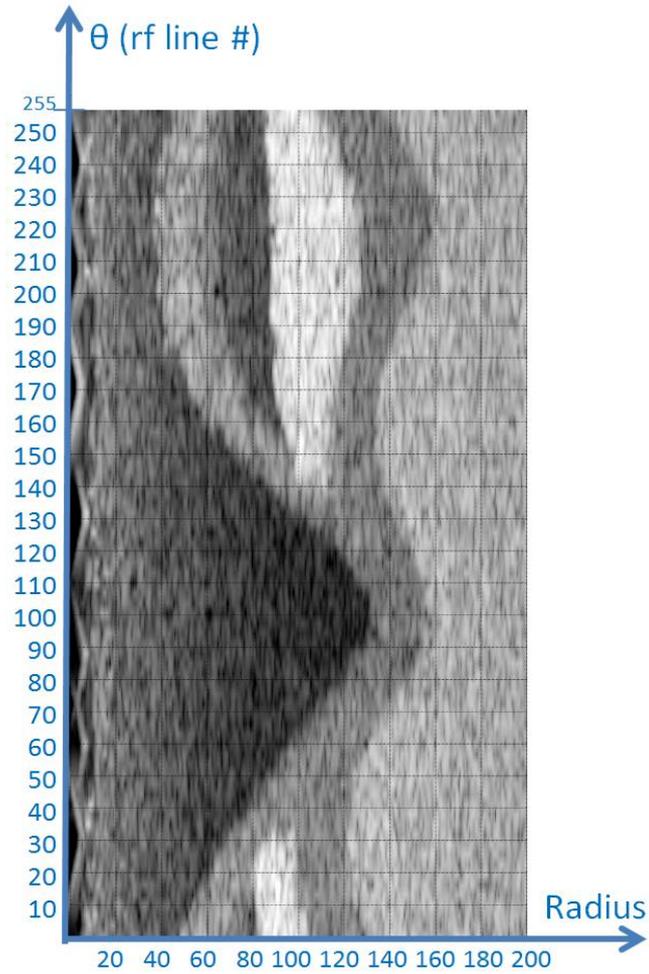


Figure 77: Stack of the 256 B-mode vectors after downsampling forming the polar image.

The vectors were distributed radially and uniformly from the center of a 400x400 pixel image (Figure 78).

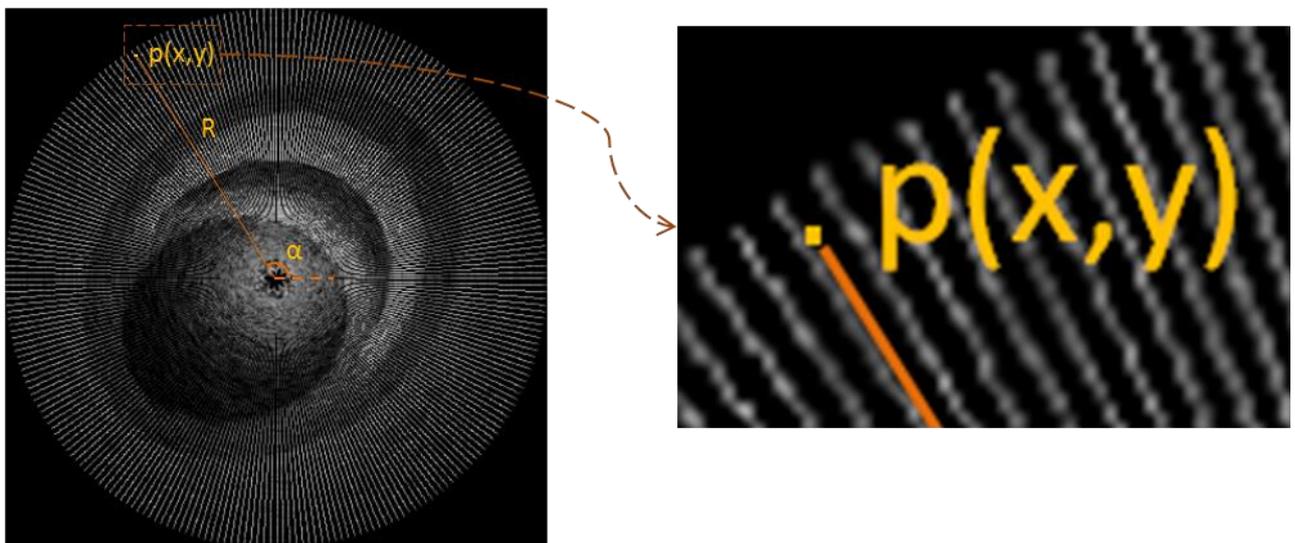


Figure 78: Radial distribution of the vectors throughout the image. The yellow pixel $p(x,y)$ represents a pixel that was not covered by the lines, R and α represents the distance from the center, and angulation, respectively.

In order to fill the empty spaces in the image that were not covered by the radial vectors, bilinear interpolation was performed. Let $p(x, y)$ be a pixel that was not covered by the radially distributed vectors. First, we calculate its distance from the image center R (radius) and its angulation α (Figure 78).

$$\text{Eq. 112} \quad R = \|(x, y) - (x_c, y_c)\|$$

$$\text{Eq. 113} \quad \alpha_{RF} = \alpha \cdot \frac{256}{360^\circ}$$

Then, bilinear interpolation was performed as explained in Section 5.6.2.

Figure 79 illustrates an example of the resulting image.

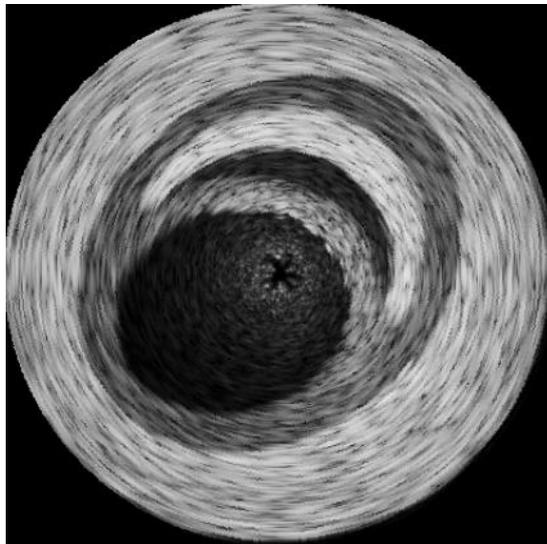


Figure 79: Interpolated image.

Figure 80 illustrates the sequence of simulated US frames during deformation.

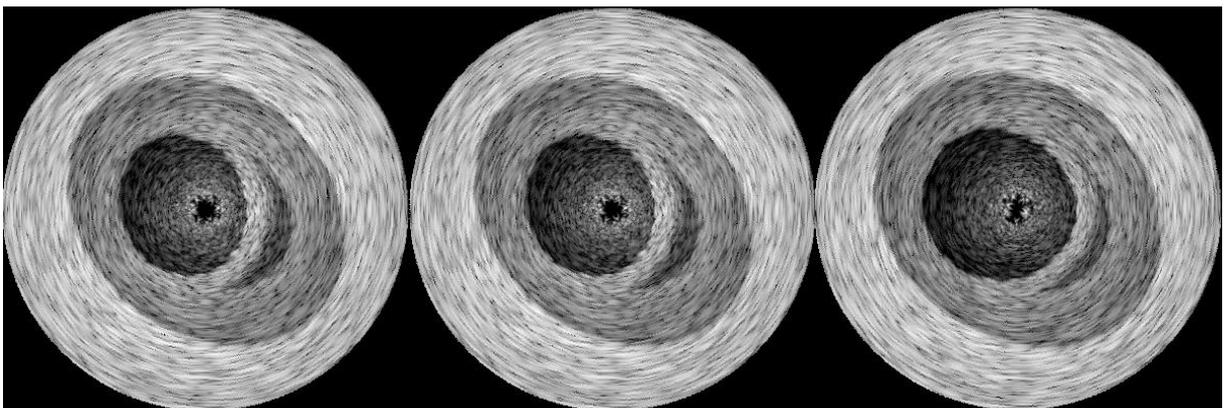


Figure 80: Sequence of B-mode images. From right to left, initial position, after compression of 5.33Pa and after compression of 100kPa.

11.3 IVUS Simulation: 3D phantoms

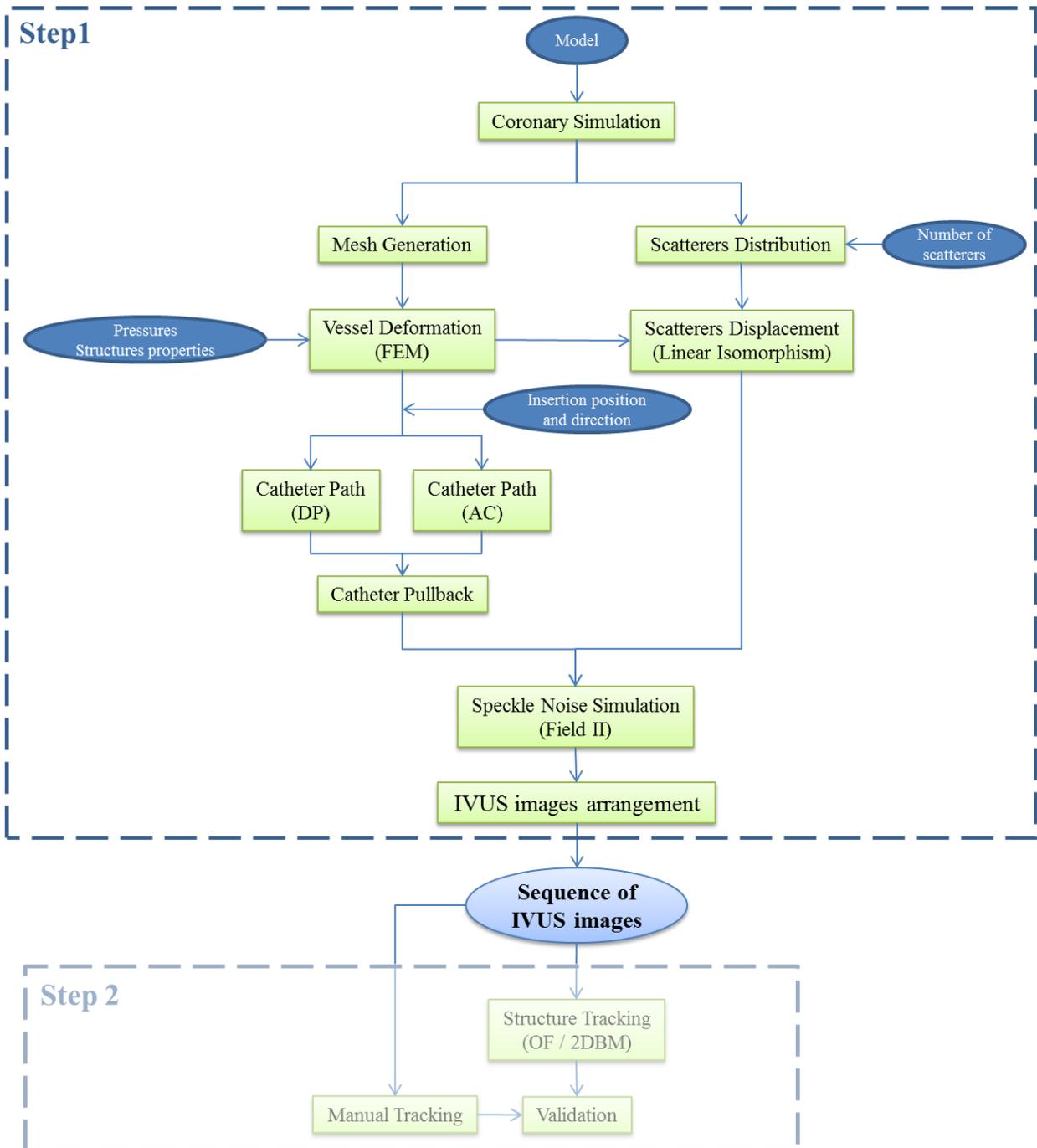


Figure 81: Block Diagram. Step 1 depicts the procedure to obtain the simulated IVUS images sequence. Step 2 illustrates the sequence for the movement estimation.

Firstly, we created a numeric phantom in order to mimic a 3D blood vessel. Then we applied FEM to simulate its deformation during a cardiac cycle. Next, we computed the catheter position inside the blood vessel and simulated the pullback. Then we included speckle noise to simulate IVUS images. In this work, we developed a software to simulate the intravascular ultrasound acquisition. The software is based on Matlab and is called IVUSSim. After the

publication of IVUSSim, it will be available online free from charges for educational and research purposes.

The sequence in Step 2 of Figure 81 is explained in section 7.

11.3.1 Coronary simulation

We created numeric phantoms that simulate the blood vessel and its structures. The image (Figure 82) represents a simulated coronary with two lesions. The properties (elasticity, scatterers density and scatterers amplitude) of each structure may be chosen as explained in Section 11.2.1.

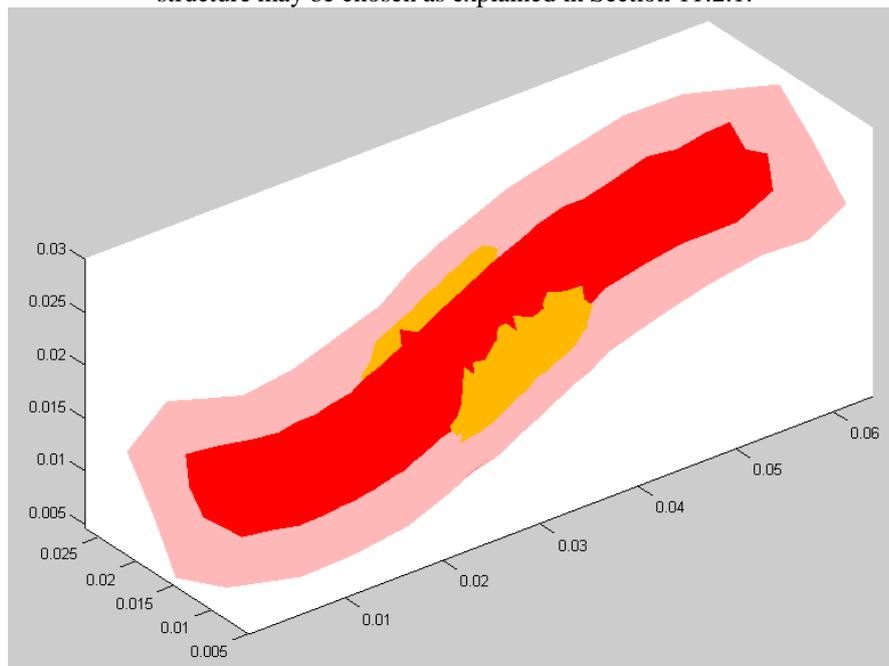


Figure 82: Simulated coronary, vessel wall in light red, lumen in red and two lesions in yellow

11.3.2 Cardiac Movements Simulation (FEM)

We have simulated the cardiac compression through FEM, in order to get the static compressed image for several pressure values.

Firstly, we generated a mesh using Delaunay algorithm, dividing the vessel volume in tetrahedrons (Figure 83). The nodes at the outer border of the vessel constituted the boundary conditions and, therefore, fixed nodes.

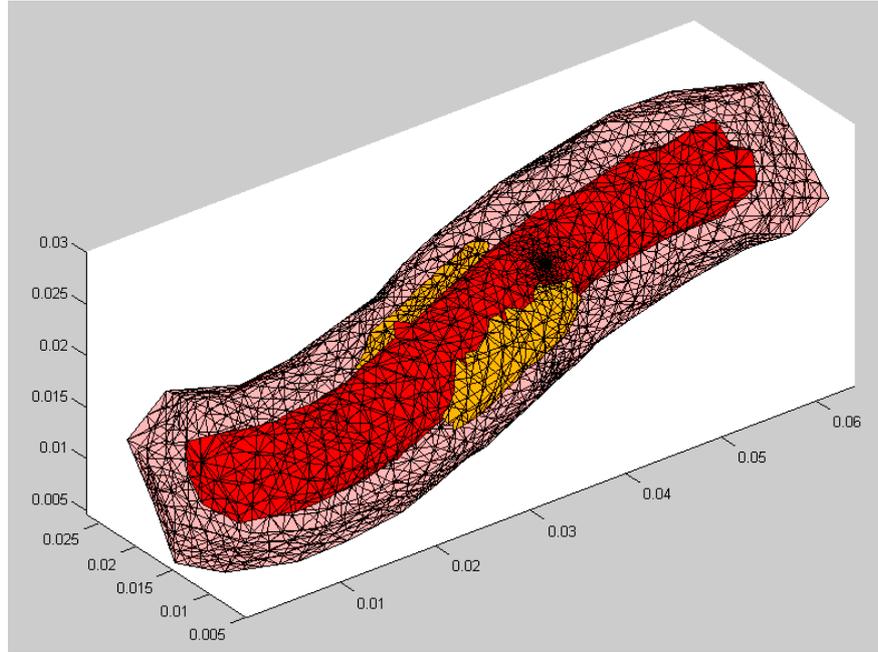


Figure 83: Generated mesh dividing the simulated coronary in tetrahedrons

The pressure forces were applied at the nodes located at lumen border and their directions were perpendicular to the triangular face that contains the nodes. First, we located the faces that belong to the lumen border and calculated their area. Then we multiplied the pressure by the area in order to obtain the force applied on each face. Next, we divided the product by three and the result represents the intensity of the force vector distributed the force to each of the three nodes that constitute the triangular face.

$$\text{Eq. 114} \quad F_{\alpha} = \text{pressure} \cdot A_{\alpha}$$

$$\text{Eq. 115} \quad \vec{F}_{\alpha} = F_{\alpha} \cdot \vec{n}_{\alpha}$$

$$\text{Eq. 116} \quad \begin{cases} \vec{F}_1 = \vec{F}_1 + \vec{F}_{\alpha}/3 \\ \vec{F}_2 = \vec{F}_2 + \vec{F}_{\alpha}/3 \\ \vec{F}_3 = \vec{F}_3 + \vec{F}_{\alpha}/3 \end{cases}$$

where α is a triangle that belongs to the lumen border, A_{α} is its area, F_{α} is the force applied on α . \vec{n}_{α} is the vector normal to the plane that contains α and $\{\vec{F}_1, \vec{F}_2, \vec{F}_3\}$ are the forces applied at the three nodes on the vertices of α .

All forces, $\{\vec{F}_1, \vec{F}_2, \vec{F}_3, \vec{F}_4, \dots, \vec{F}_N\}$, are initialized with zero value and are incremented iteratively. For example, let node1 be the vertice of the triangles α, β, γ and δ . \vec{F}_1 will have the following result.

$$\text{Eq. 117} \quad \begin{cases} \vec{F}_1 = \vec{F}_1 + \vec{F}_\alpha/3 \\ \vec{F}_1 = \vec{F}_1 + \vec{F}_\beta/3 \\ \vec{F}_1 = \vec{F}_1 + \vec{F}_\gamma/3 \\ \vec{F}_1 = \vec{F}_1 + \vec{F}_\delta/3 \end{cases} \rightarrow \vec{F}_1 = \vec{F}_\alpha/3 + \vec{F}_\beta/3 + \vec{F}_\gamma/3 + \vec{F}_\delta/3$$

Figure 84 illustrates the force vectors at the nodes on the lumen border.

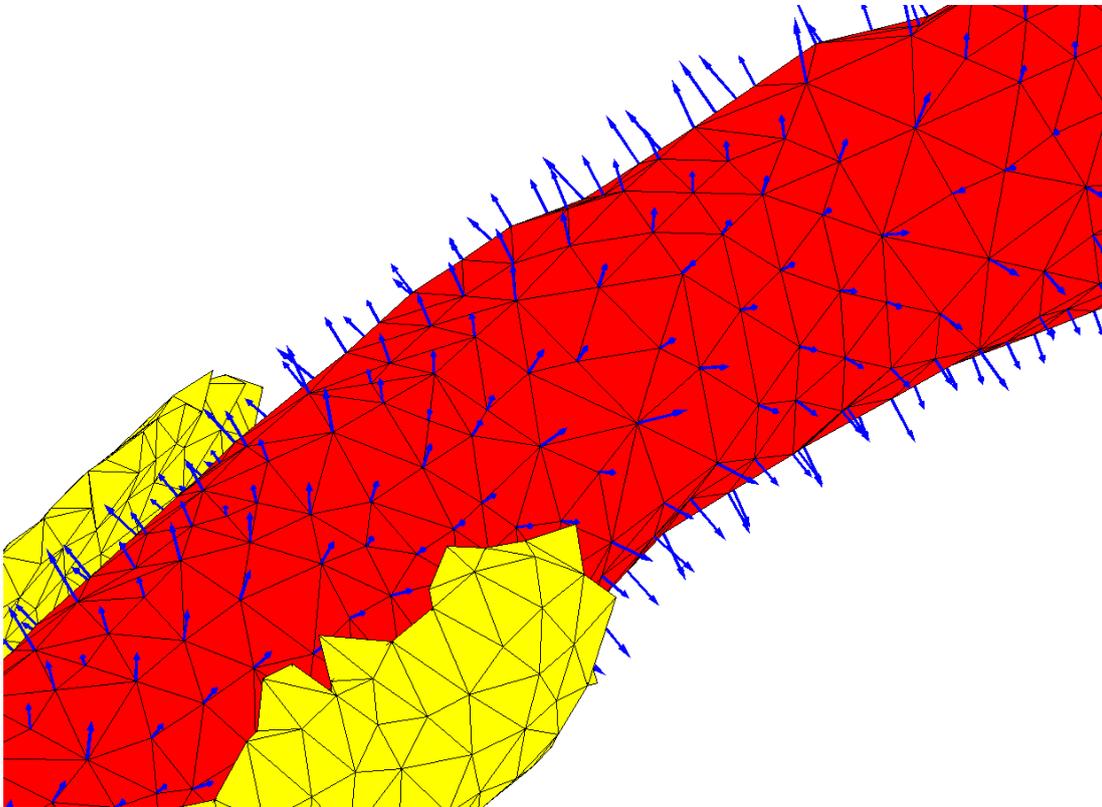


Figure 84: Force (blue vectors) applied at the nodes located at the lumen border.

Finally, we applied the Finite Elements Method with the following parameters (Table 24).

Table 24: Parameters of finite elements method

Parameter	Symbol	Value
Internal Variation of Pressure	ΔP	Cardiac = 40 mmHg / 5.33 kPa Balloon = 1520 mmHg / 200 kPa
Poisson Ratio	N	0.49
Young Moduli	E	Intima/media = 600 kPa Adventitia = 80 kPa Calcified lesion = 1200 kPa Lipidic Lesion = 25 kPa Fibrotic lesion = 296 kPa

Figure 85 illustrate the deformation of the blood vessel. Figure 85a and Figure 85b represent the blood vessel initial position and after compression of 1 atm, respectively. Although internal pressure 1 atm is not quite realistic, Figure 85b was included to facilitate the visualization of the vessel deformation.

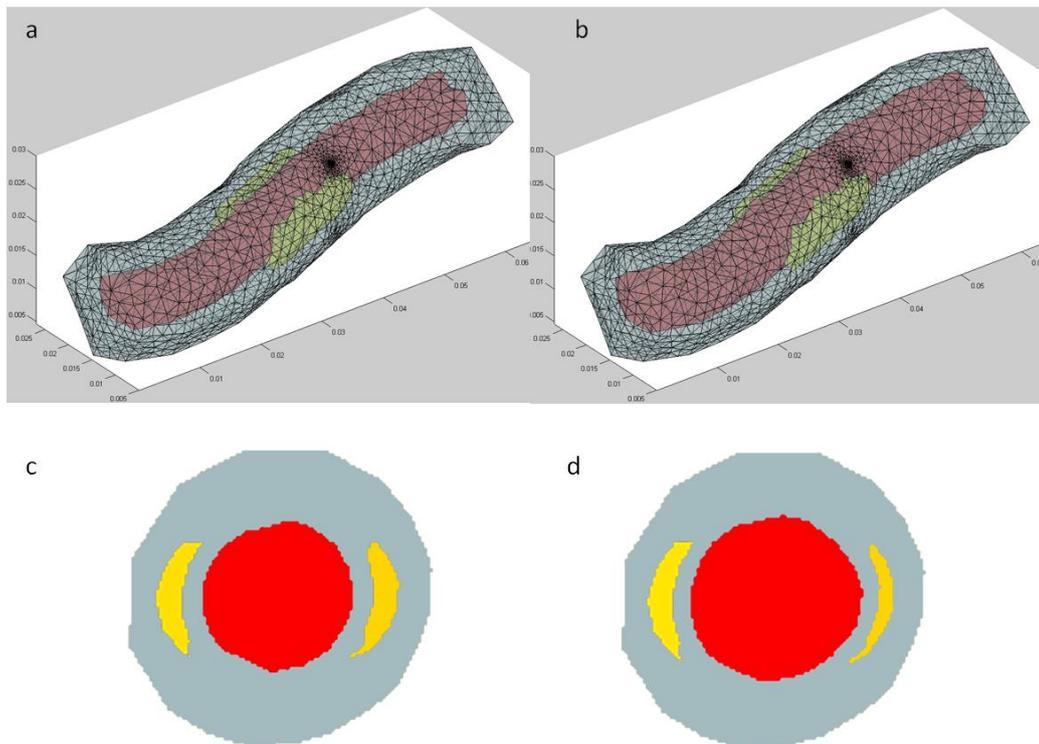


Figure 85: a) vessel mesh before deformation, b) vessel mesh after deformation, c) vessel cross-section before deformation, d) vessel cross-section after deformation.

11.3.3 Catheter path simulation

The catheter path simulation was performed as explained in Section 5.9.1.

11.3.4 Catheter pullback

The catheter path simulation was performed as explained in Section 5.9.2.

11.3.5 Scatterers distribution

The scatterers were distributed as explained in Section 5.3.1

11.3.5.1 Scatterers movement according to the coronary deformation

The procedure to simulate the scatterers movement is the same technique presented in section 5.3.2

11.3.6 Speckle Noise Simulation

11.3.6.1 Scatterers positioning

Section 11.3.4 explained how the catheter pullback is simulated. As a result, the position $P_{cat} = \{x_{cat}, y_{cat}, z_{cat}\}$ and normalized direction $\hat{d}_{cat} = \{u_{cat}, v_{cat}, w_{cat}\}$ of the catheter in each frame is provided. This information is useful to determine the origin and direction of the ultrasound beams inside the lumen.

To facilitate the transducer set up, the catheter was considered the center of the axis with direction $\{1,0,0\}$ (Figure 86). Hence, all scatterers were translated and rotated as follows

$$\text{Eq. 118} \quad [x'_i \ y'_i \ z'_i] = [x_i \ y_i \ z_i] - [x_{cat} \ y_{cat} \ z_{cat}] , \ i = 1, 2, 3, \dots, N$$

$$\text{Eq. 119} \quad [\hat{u}, \hat{v}] = \frac{[u, v]}{|[u, v]|}$$

$$\text{Eq. 120} \quad \sin\left(\frac{\theta}{2}\right) = \frac{|[\hat{u}, \hat{v}] - [1, 0]|}{2}$$

$$\text{Eq. 121} \quad \theta = -\text{sign}(v) * 2 * \arcsin\left(\frac{\theta}{2}\right)$$

$$\text{Eq. 122} \quad [\hat{u}, \hat{w}] = \frac{[u, w]}{|[u, w]|}$$

$$\text{Eq. 123} \quad \sin\left(\frac{\varphi}{2}\right) = \frac{|[\hat{u}, \hat{w}] - [1, 0]|}{2}$$

$$\text{Eq. 124} \quad \varphi = -\text{sign}(w) * 2 * \arcsin\left(\frac{\varphi}{2}\right)$$

$$\text{Eq. 125} \quad \begin{bmatrix} x''_i \\ y''_i \\ z''_i \end{bmatrix} = \begin{bmatrix} x'_i \cdot \cos(\theta) - y'_i \cdot \sin(\theta) \\ x'_i \cdot \sin(\theta) + y'_i \cdot \cos(\theta) \\ z'_i \end{bmatrix}, i = 1, 2, 3, \dots, N$$

$$\text{Eq. 126} \quad \begin{bmatrix} x'''_i \\ y'''_i \\ z'''_i \end{bmatrix} = \begin{bmatrix} x''_i \cdot \cos(\varphi) - z''_i \cdot \sin(\varphi) \\ y''_i \\ x''_i \cdot \sin(\varphi) - z''_i \cdot \cos(\varphi) \end{bmatrix}, i = 1, 2, 3, \dots, N$$

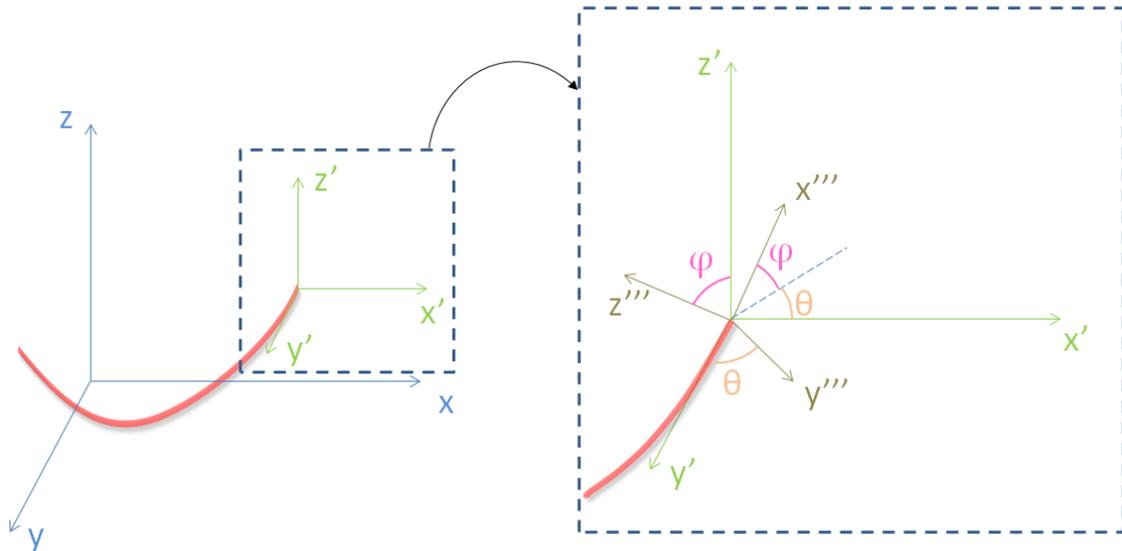


Figure 86: Translation (left) and rotation (right) top put the reference at the catheter tip

Once scatterers coordinates are changed to take the transducer as reference, the simulation of the ultrasound and construction of the B-mode images are performed as explained in Sections 11.2.4.2 and 11.2.4.3, respectively.

11.3.7 Arrangement of frames from different cardiac phases

The arrangement of the simulated IVUS frames were performed as explained in Section 5.9.3.

11.3.8 Catheter twist simulation

Another phenomenon experienced during the catheter pullback in clinical cases is the catheter physical rotation around its longitudinal axis. This phenomenon may be simulated by three different techniques:

- 1- After the translation of the scatterers to the position where the catheter is the reference (Eq. 118-Eq. 126), the scatterers are linearly rotate using Eq. 111.
- 2- Rotate the distribution of the RF data.
- 3- Rotate the B-mode image (Figure 79) after it is formed.

The catheter twist simulation was not performed because it is beyond the scope of this study